

David de Oliveira Lemes

Fábula PXP

A técnica de Programação Exploratória (PXP): Projetos de Criação e Desenvolvimento de Jogos Digitais

São Paulo

2015

David de Oliveira Lemes

Fábula PXP

**A técnica de Programação Exploratória (PXP):
Projetos de Criação e Desenvolvimento de Jogos Digitais**

Tese apresentada à Banca Examinadora da Pontifícia Universidade Católica de São Paulo, como exigência parcial para obtenção do título de DOUTOR em Tecnologias da Inteligência e Design Digital, na área de concentração de Processos Cognitivos e Ambientes Digitais, na linha de pesquisa de modelagem de Sistemas de Software.

Pontifícia Universidade Católica de São Paulo – PUC-SP

Departamento de Computação

Tecnologias da Inteligência e Design Digital

Orientador: Dr. Ítalo S. Vega

São Paulo

2015

David de Oliveira Lemes
Fábula PXP
A técnica de Programação Exploratória (PXP):
Projetos de Criação e Desenvolvimento de Jogos Digitais/ David de Oliveira Lemes.
– São Paulo, 2015-
145 p. : il. (algumas color.) ; 30 cm.

Orientador: Dr. Ítalo S. Vega

Tese (Doutorado) – Pontifícia Universidade Católica de São Paulo – PUC-SP
Departamento de Computação
Tecnologias da Inteligência e Design Digital, 2015.

1. Fábula PXP. 2. Programação Exploratória. I. Vega, Ítalo S.. II. Pontifícia
Universidade Católica de São Paulo. III. Faculdade de Ciências Exatas e Tecnolo-
gia. IV. Título

CDU 02:141:005.7

David de Oliveira Lemes

Fábula PXP
A técnica de Programação Exploratória (PXP):
Projetos de Criação e Desenvolvimento de Jogos Digitais

Tese apresentada à Banca Examinadora da Pontifícia Universidade Católica de São Paulo, como exigência parcial para obtenção do título de DOUTOR em Tecnologias da Inteligência e Design Digital, na área de concentração de Processos Cognitivos e Ambientes Digitais, na linha de pesquisa de modelagem de Sistemas de Software.

Dr. Ítalo S. Vega
Orientador

Professor
Convidado 1

Professor
Convidado 2

Professor
Convidado 3

Professor
Convidado 4

Para Andrea, Pedro e Livia.

Agradecimentos

À minha família, que soube entender todos os meus momentos de ausência. Esta parece ser uma frase padrão em muitos trabalhos desta natureza.

A todos os colegas do GEMS - Grupo de Estudos em Modelagem de Software da PUC-SP, que ouviram as minhas ideias, ainda em construção, pacientemente ao longo dos últimos quatro anos e tiveram a delicadeza e generosidade em colaborar a cada nova apresentação. Foram excelentes orientações coletivas.

Aos amigos do Departamento de Computação da PUC-SP, que sempre se interessaram pela pesquisa e sempre que puderam, colaboraram com ideias, comentários e indicações de leitura.

Ao Prof. Dr. Ítalo Santiago Vega, pela presença e preocupação constante durante toda a trajetória desta pesquisa (e além dela também), amizade e empenho acadêmico admirável.

Aos colegas do TIDD - Programa de Pós-graduação em Tecnologias da Inteligência e Design Digital da PUC-SP pelas ajudas recebidas, especialmente para Eduardo Camargo, Francisco Supino Marcondes e Mário Madureira Fontes, seja pela colaboração direta com links interessantes sobre assuntos relacionados à pesquisa ou por compartilhar o conhecimento em Latex¹.

Um agradecimento especial ao Vitor Ferreira, que participou do processo do experimento prático presente neste trabalho.

¹ Mais detalhes em: <<http://www.latex-project.org>>. Acessado em 17/01/2015

Resumo

O desenvolvimento de jogos digitais é uma atividade exponencialmente em ascensão à medida que os mais diversos dispositivos os quais suportam os games se popularizam. Sob o olhar do desenvolvimento de games, nos dias atuais, diversas ferramentas possibilitam que o mais inexperiente programador possa criar e desenvolver um jogo digital.

Esta pesquisa apresenta um modelo de trabalho que busca apresentar a importância da narrativa nos jogos, como as ideias são organizadas para a produção de um jogo digital e, sobretudo, uma técnica de programação que ajudará o game designer, ou o projetista de jogos digitais a entender como uma máquina de cálculos, o computador, pode ampliar ou limitar os usos das ideias e narrativas originais do projeto.

Palavras-chave: Fábula de Jogo, Unidade Mínima de Fábula, Programação Exploratória, Documento de Projeto de Jogo, Documento Executável de Jogo.

Abstract

The development of digital games is an activity that grows exponentially as the many game supporting devices become popular. Under the game development perspective, nowadays, several tools "promise" to make the most inexperienced programmer able to create and develop a digital game.

This research presents a work model that seeks to introduce the importance of narrative into games, the way the ideas are organized to produce a digital game and, overall, a programming technique that will help the game designer, or digital games designer to understand how a calculations, the computer, may enhance or hinder the use of ideas and original narrative of a project.

Keywords: Game Fable, Minimal Fable Unity, Exploratory Programming, Document of Game Project, Executable Game Document.

Lista de ilustrações

Figura 1 – Super Mario Bros e Princesa Peach	27
Figura 2 – Regras simples para o projeto de um jogo	29
Figura 3 – Estudos para a Natividade — Leonardo da Vinci	44
Figura 4 – O protótipo de Braid e sua versão final.	54
Figura 5 – Fluxo do jogo RunnerChief	124
Figura 6 – Fluxo de telas do jogo RunnerChief	125
Figura 7 – Pré PXP: Vila Mágica	125
Figura 8 – Pré PXP: Floresta Mágica	126
Figura 9 – Pré PXP: Nuvens Mágicas, Dia	126
Figura 10 – Pré PXP: Nuvens Mágicas, Noite	127
Figura 11 – Pós PXP: Vila Mágica	127
Figura 12 – Pós PXP: Floresta Mágica	128
Figura 13 – Pós PXP: Nuvens Mágicas, Dia	128
Figura 14 – Pós PXP: Nuvens Mágicas, Noite	129

Lista de tabelas

Tabela 1 – Atividades de uma prototipação.	52
--	----

Sumário

Introdução	19
I Fundamentos Conceituais	23
1 Fábula de Jogo, Narrativa e Projeto de Jogos	25
1.1 Fábula de Jogo	25
1.2 Unidade Mínima de Fábula	28
1.3 Processo de Projeto de Jogo	28
1.4 Sentido da Documentação e Problemas Invisíveis	31
II Programação Exploratória e Fábula PXP	35
2 Programação Exploratória de Computadores	37
2.1 Algoritmos e Programação	37
2.2 Programação Exploratória	40
2.3 PXP e o Design	42
2.4 Prática da Programação Exploratória	45
3 Fábula PXP	49
3.1 Documento Executável de Jogo	49
3.2 Prototipação	51
3.2.1 Braid e a prototipação de ideias	53
3.3 Linguagens de programação e Engines	55
3.4 Fábula PXP	56
4 Estudo de Caso	59
Conclusão	65
Referências	69
Apêndices	73
APÊNDICE A A Criação do Projeto RunnerChief	75
A.1 Documento de Brainstorm	75
APÊNDICE B Diário de Desenvolvimento	81

B.1 Projeto RunnerChief	81
APÊNDICE C Documentação do Projeto RunnerChief	101
C.1 Documento de Projeto de Jogo	101

Introdução

Começar um texto introdutório comentando os números de um mercado em franco crescimento como o mercado de games é repetir o que diversas pesquisas acadêmicas e textos jornalísticos fazem continuamente, mas é difícil fugir desta armadilha, pois estes números crescem a cada dia. Segundo o Mapeamento da Indústria Brasileira e Global de Jogos Digitais², realizado pelo Núcleo de Política e Gestão Tecnológica (PGT) da Universidade de São Paulo (USP), em 2011 o mercado mundial de games movimentou US\$ 74 bilhões, e as previsões indicam que deverá ultrapassar US\$ 82 bilhões em 2015. No Brasil, estima-se que o mercado já esteja perto de US\$ 3 bilhões. Como podemos ver, o setor de desenvolvimento de jogos digitais é uma atividade que cresce exponencialmente à medida que os mais diversos dispositivos que suportam os games se popularizam. São consoles dos mais diversos, smartphones, tablets e também jogos que rodam diretamente na internet em qualquer navegador. É um mercado em franca expansão.

Sob o olhar do desenvolvimento de games, nos dias atuais, diversas ferramentas possibilitam que o mais inexperiente desenvolvedor possa criar e programar um jogo digital utilizando os recursos já existentes nestes softwares de desenvolvimento. Esta simplicidade se deve ao fato de as engines de games³, (ou motores de jogos), softwares que incorporam diversas funcionalidades presentes nos games também se popularizam em função das facilidades oferecidas para os desenvolvedores de games e do baixo custo atual.

Mas qual é a implicação de um processo como este? Ao passo que tais ferramentas automatizam as mais diversas etapas de produção de um jogo, podem acabar com o que é o grande diferencial em um projeto de jogo digital: a criatividade. Basta analisar o seguinte: se todos os produtores de jogos usam as mesmas ferramentas de desenvolvimento, os mesmos softwares de modelagem 3D e quase sempre, as mesmas linguagens de programação, o que diferencia um jogo de outro? Sua concepção criativa, seu roteiro, sua narrativa e seus elementos artísticos. O processo de criação presente na produção de um game é fundamental para que boas ideias venham a ganhar forma e, conseqüentemente, sejam codificadas para assim serem então transformadas em um jogo digital, sem deixar de considerar um ponto crucial presente

² Para ler a pesquisa completa, visite o endereço: <<http://bit.ly/1hEzYMg>>. Acessado em 17/01/2015.

³ Também chamado de motor de jogo, conhecido pelo termo em inglês, game engine, ou simplesmente engine, é um programa de computador (ou conjunto de bibliotecas) que visa simplificar e abstrair o desenvolvimento de jogos digitais ou outras aplicações com gráficos em tempo real, para videogames ou computadores. As funcionalidades tipicamente fornecidas por um motor de jogo incluem: motor gráfico para renderizar gráficos 2D e 3D, motor de física para simular a física ou simplesmente para fazer detecção de colisão, suporte a animação, sons, inteligência artificial, jogos em rede, gerência de memória, gerência de arquivos entre outros. Mais detalhes em: <<http://bit.ly/1CjSrFN>>. Acessado em 17/01/2015.

em todos os games: a narrativa. O Capítulo 1 apresenta a fundamentação conceitual para se pensar a produção de jogos digitais e propõe uma maneira particular de pensar a narrativa nos games, além da forma de como documentar um projeto de criação e desenvolvimento de um game.

A presente pesquisa apresenta a técnica de programação como recurso para auxiliar na construção de narrativas de jogos para que estes sejam o mais fiel possível à ideia conceitual e assim preservar (ou mesmo ampliar) o projeto original. O Capítulo 2 apresenta a técnica de Programação Exploratória visando lidar com o problema da criatividade anteriormente mencionado.

Além desta técnica de programação, esta tese apresenta formas de documentar a criação de um jogo digital. Dentro deste processo de criação, o conceito inicial, a ideia que dará origem à narrativa presente no jogo digital, é um dos fatores (mas não o único) que pode determinar o bom andamento durante a produção de game. E podemos dizer que tudo começa pela narrativa, pela história a ser contada, pela história a ser jogada. É a boa história que pode determinar assim a boa produção de um jogo digital.

Mas como criar uma boa história de forma que ela seja a base da produção de um jogo digital? E como determinar que elementos presentes nesta narrativa sejam passíveis, ou não, de serem codificados a ponto de serem transformados em um jogo digital? Poderia algum processo ou técnica de programação ajudar no desenvolvimento narrativo de jogo digital? Será possível construir um método de trabalho onde uma técnica de programação estivesse à disposição dos criadores de um jogo digital para contribuir em novas proposições narrativas?

A grande proliferação de cursos de graduação em Jogos Digitais e, conseqüentemente, a ampliação do mercado como um todo, faz com que a metodologia de produção de jogos seja cada vez mais pensada e repensada. Do ponto de vista do desenvolvimento de software, podemos aplicar as mais diversas técnicas já estabelecidas pela Engenharia de Software⁴ em busca de um produto de qualidade. Contudo, isto não basta quando o assunto é um jogo digital. O game, apesar de ser um software de entretenimento, conta com elementos que têm como principal fator a criação e a narrativa. Repensar a forma de trabalhar o projeto de um jogo digital é fundamental para que este setor se desenvolva ainda mais. E este repensar implica em não estar preso às tecnologias atuais e disponíveis para a produção e jogo, e sim manter o foco na liberdade de criação narrativa amparada pelos recursos técnicos e tecnológicos existentes atualmente.

⁴ Engenharia de Software é uma área da computação voltada à especificação, desenvolvimento e manutenção de sistemas de software, com aplicação de tecnologias e práticas de gerência de projetos e outras disciplinas, visando organização, produtividade e qualidade. Mais detalhes em: <<http://bit.ly/17TRZF0>>. Acessado em 17/10/2015.

Esta tese de Doutorado tem como objetivo estabelecer um conceito narrativo para que este possa servir como base na formulação de um projeto de criação e desenvolvimento de um jogo digital, tendo como principal meta a apresentação de uma metodologia que aplica uma técnica de programação específica, auxiliando assim no processo de construção narrativa e desenvolvimento tecnológico. O Capítulo 3 apresenta estes pontos e busca refletir como uma técnica de programação pode ter influência na narrativa de um jogo digital.

E para embasar a tese aqui apresentada, o Capítulo 4 apresenta um estudo de caso que tem como objetivo aplicar, de forma prática, os pontos teóricos abordados no decorrer da pesquisa e mostrar como um jogo pode ser pensado e construído com base nos pontos estudados e apresentados. Soma-se a este estudo de caso, três apêndices (A, B e C) que mostram a documentação detalhada de todo o projeto.

Parte I

Fundamentos Conceituais

1 Fábula de Jogo, Narrativa e Projeto de Jogos

*Basta alguém digitar o
encantamento correto em um
teclado e logo uma tela de
visualização ganha vida,
mostrando coisas que não
existiam nem poderiam existir.*

— Frederick P. Brooks Jr.

1.1 Fábula de Jogo

Games são obras narrativas. Independente da natureza do jogo digital, até mesmo um simples puzzle lógico traz em si elementos narrativos que contam uma história, uma aventura ou simplesmente um processo de descoberta.

Em um jogo digital, além de você poder jogar uma história, você pode interpretá-la de várias maneiras. A cada nova etapa vencida, uma nova descoberta se faz. E não se pode negar que a narrativa encontra no game um habitat fértil (SANTAELLA, 2007).

Fértil, pois o ambiente digital, onde acontecem as dinâmicas de jogo, propicia que a multilinearidade narrativa inerente a este meio esteja presente e seja aplicada, lida e relida sempre que um novo ato de jogar aconteça.

Contudo, é preciso aqui começar a definir um caminho a ser percorrido na presente pesquisa e perguntar: mas o que são de fato obras narrativas? De acordo com (BARTHES, 2011), “a narrativa pode ser sustentada pela linguagem articulada, oral ou escrita, pela imagem, fixa ou móvel, pelo gesto ou pela mistura ordenada de todas estas substâncias; está presente no mito, na lenda, na fábula, no conto, na novela, na epopéia, na história, na tragédia, no drama, na comédia, na pantomima, na pintura, no vitral, no cinema, nas histórias em quadrinhos, na conversação”. Como a primeira edição da obra citada foi escrita em 1973, o game, o videogame e os jogos digitais não estão na lista do autor. Fosse o livro escrito por Barthes alguns anos depois, estariam, pois games contam histórias que podem ser jogadas.

Nesta tese usaremos o conceito de fábula proposto por (BAL, 1997) para estabelecer os parâmetros de análise na narrativa. E dentro de um estudo aprofundado sobre o tema,

a autora apresenta o conceito de fábula. Como podemos definir o que é uma fábula dentro deste contexto? Antes é preciso saber como o termo é usado.

Na Língua Portuguesa, a palavra “fábula” apresenta diversos significados, como: narrações alegóricas cujas personagens são, por via de regra, animais, e que encerra uma lição de moral, com as fábulas de La Fontaine. Mitologia, lenda: os deuses da fábula. Narração de coisas imaginárias: ficção. Fabulação. Assunto de crítica. Enredo. Quantia ou importância muito elevada: grande soma de dinheiro (PORTELLA, 1983).

Já o Moderno Dicionário da Língua Portuguesa, Michaelis¹, define fábula destas seis formas: 1 - Pequena narrativa em que se aproveita a ficção alegórica para sugerir uma verdade ou reflexão de ordem moral, com intervenção de pessoas, animais e até entidades inanimadas. 2 - Narração imaginária, ficção artificiosa. 3 - Narrativa ou conjunto de narrativas de ideação mitológica; mito. 4 - Entrecho ou urdidura de qualquer obra de ficção. 5 - Os elementos de deformação da realidade nas composições do gênero épico ou de invenção. 6 - Mentira.

A segunda definição de “fábula” (narração imaginária, ficção artificiosa) contribui para o uso da palavra fábula no contexto deste trabalho de pesquisa, assim como uma definição de Portella: narração de coisas imaginárias, ficção. Este é o sentido de fábula que adotaremos neste trabalho.

Tendo a narrativa como um dos pontos relevantes deste trabalho, vamos assumir que, dentro do contexto da Língua Portuguesa, o conceito de enredo e narrativa para tratar especificamente da fábula e construir aqui um raciocínio para a apresentação do conceito de Fábula de Jogo.

Cabe aqui considerar uma fábula como um agrupamento específico de uma série de acontecimentos. A fábula, como um todo, constitui um processo, embora cada evento também possa ser chamado de processo, ou pelo menos parte dele. É fundamental distinguir as três fases da fábula: a possibilidade (ou virtual), o evento (ou realização) e o resultado (ou conclusão) do processo. Nenhuma dessas três fases é indispensável. Uma possibilidade pode ou não ser realizada. E mesmo que o evento aconteça, nem sempre é garantida uma conclusão satisfatória (BAL, 1997).

Cabe aqui um exemplo específico desta do universo dos games. O jogo Super Mario Bros, lançado em 1985 pela Nintendo,² conta com oito mundos e, cada um destes mundos possui quatro fases. O objetivo final do personagem principal é enfrentar, no oitavo e último mundo, o vilão chamado Bowser, salvar a Princesa Peach e libertar os habitantes do Reino

¹ Para mais informações, acesse a versão online do Dicionário Michaelis em <http://michaelis.uol.com.br>. Acessado em 22/12/2014.

² Para mais informações sobre o jogo Super Mario Bros, acesse: <http://bit.ly/1HQ40br>

dos Cogumelos (Figura 1).

Figura 1 – Super Mario Bros e Princesa Peach



Fonte: Nintendo of America

Para chegar ao objetivo final, Mario, o personagem principal, vive uma sequência de eventos e acontecimentos, formando assim a narrativa presente no jogo. Contudo, a conclusão do objetivo final, nem sempre é satisfatória e alguns eventos podem não acontecer em um primeiro momento, pois o jogador precisa aprender como vencer os vilões para assim salvar a Princesa Peach. Apesar de uma conclusão não satisfatória, o jogo continua até a conclusão satisfatória.

Como podemos ver, a história do jogo é uma narrativa. E, de acordo com (BREMONT, 2011), “toda narrativa consiste em um discurso integrado em uma sucessão de acontecimentos de interesse humano na unidade de uma mesma ação”. Olhando para a história dos games, desde o seu início, até os dias atuais, não é difícil fazer esta analogia direta: um jogo digital é composto, dentre tantos elementos, de um agrupamento específico de uma série de acontecimentos. Estes acontecimentos, apesar de serem jogados quando o game está pronto, passou por um planejamento prévio, um processo de criação e escrita e construção de uma narrativa. E é este processo de criação e de definição da sequência de acontecimentos que daremos o nome aqui de Fábula de Jogo.

E neste ponto vou propor que, uma Fábula de Jogo [FDJ] é, portanto, uma sequência de acontecimentos narrativos que podem ser jogados construindo assim a narrativa presente no jogo digital. Estes acontecimentos narrativos devem estar documentados de alguma forma durante o processo de criação, antes de serem programados para serem transformados em um jogo digital. Veremos mais sobre o processo de documentação no decorrer desta pesquisa.

Sendo a proposta de Fábula de Jogo ser uma sequência de acontecimentos narrativos, o que aconteceria se isolássemos estes acontecimentos, ou seja, separarmos em acontecimentos

independentes?

1.2 Unidade Mínima de Fábula

Aqui propomos um outro termo para este trabalho. O simples acontecimento, fora da sequência de acontecimentos narrativos presentes na Fábula de Jogo, será chamado de Unidade Mínima de Fábula [UMF].

Para exemplificar o que pode ser uma Unidade Mínima de Fábula, iremos usar como exemplo o mesmo jogo já citado anteriormente (Super Mario Bros), composto por uma sequência de acontecimentos narrativos que formam assim a Fábula de Jogo. Durante o jogo, o personagem principal repete uma série de ações, como correr, pular, morrer, aumentar de tamanho, diminuir de tamanho. Estes fatos isolados, que podem ser repetidos ou não durante o período de um tempo de jogo, podem ser classificados como Unidades Mínimas de Fábulas. Estas UMFs podem estar dentro ou fora da sequência de acontecimentos narrativos. Este é um conceito importante para este trabalho, pois uma UMF pode então passar por um processo de programação e construção algorítmica, que será usada mais adiante.

E, por falar em narrativa e algoritmo, (GREIMAS, 2011) nos diz que “a narrativa deve ser considerada como um algoritmo, isto é, como uma sucessão de enunciados cujas funções simulam linguisticamente um conjunto de comportamentos orientados para um objetivo”. É como nos jogos digitais, que sempre possuem um objetivo: não existe um game sem um objetivo de jogo definido.

1.3 Processo de Projeto de Jogo

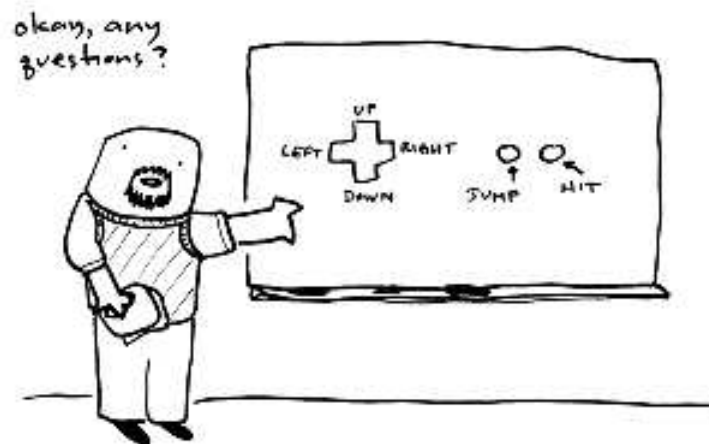
Como um jogo digital começa a ser desenvolvido? Jogos de qualquer natureza começam a serem desenvolvidos com a formulação de uma ideia inicial. Esta ideia é então submetida a uma série de ataques, mudanças, questionamentos e por fim, se transforma em um projeto a ser executado. Esta ideia inicial é o ponto de partida para a criação da Fábula de Jogo [FDJ], que determinará a sequência de eventos que darão a forma narrativa do game.

Projetos de desenvolvimento de software de qualquer natureza devem começar com uma declaração documentada do que se trata tal projeto para atender a uma necessidade inicial de uma empresa, um indivíduo ou um problema a ser resolvido. Esta declaração pode ser um simples registro de ideias, bem como complexos diagramas de fluxos, mapas mentais ou outros recursos, mas independente da matriz adorada, é uma documentação. Este documento de projeto dará a diretriz do que será produzido e desenvolvido. Estas diretrizes podem variar em tamanho, modelo e formato, mas visam sobretudo declarar o início do projeto.

Sendo o projeto de um game um projeto de software, pode-se então usar a mesma sistemática documental para o projeto de um jogo digital. Contudo, quando assunto é games, este documento pode fazer parte também do processo de criação do jogo digital. Este modelo de documento é conhecido como documento de game design (game design document, em inglês). Esta atividade de criação de um projeto de game é o que chamamos de game design. Atividade esta realizada pelo projetista de jogos, ou game designer, como é popularmente conhecido.

Um projeto de criação e desenvolvimento de um jogo digital difere em diversos aspectos de um projeto de software, sobretudo no que diz respeito a questões de arte, design e narrativa. Projetar um jogo é projetar um sistema de entretenimento que, segundo (SOMMERVILLE, 2011), “são sistemas cuja utilização principal é pessoal e cujo objetivo é entreter o usuário. A maioria desses sistemas é de jogos de diferentes tipos. A qualidade de interação com o usuário é a característica particular mais importante dos sistemas de entretenimento”. Esta interação se dá, sobretudo, com a experiência de jogo. E a experiência de jogo só pode ser criada a partir de uma Fábula de Jogo.

Figura 2 – Regras simples para o projeto de um jogo



Fonte: GamerCareerGuide

Podemos dizer que o projeto de criação e desenvolvimento de um jogo começa com a definição do seu conceito, que tem como finalidade definir a Fábula de Jogo (Figura 2). Para (NOVAK, 2010), “a fase de desenvolvimento do conceito começa quando a ideia do game é criada e termina quando é tomada a decisão de começar a planejar o projeto. Durante a fase de conceito, a equipe de desenvolvimento pode ser bem pequena, podendo incluir apenas um designer, um programador, um ilustrador e um produtor. O objetivo do desenvolvimento do

conceito é decidir em que consiste o game e transmitir essa ideia a outras pessoas por escrito”. Note que, esta atividade de definição de conceito já é uma atividade de game design, ou seja, de projeto de jogo. É, portanto, o projeto da Fábula de Jogo, ou o projeto da sequência de acontecimentos narrativos.

A atividade de game design pressupõe basicamente o processo pelo qual um jogo digital é criado conceitualmente. É uma atividade que tem como princípio básico explorar todas as possibilidades criativas e interativas do projeto sem se preocupar com a produção e implementação final do game. É o estágio inicial do projeto onde todas as ideias são catalogadas e ordenadas.

São estas ideias que no momento correto serão implementadas e codificadas para então se transformar em um game para as pessoas jogarem. De acordo com (SCHELL, 2011), “quando as pessoas jogam, elas vivenciam uma experiência. É com essa experiência que o designer de jogos se preocupa. Sem experiência, o jogo é inútil”. Portanto, podemos afirmar que o projetista de jogos é um projetista de experiências em games e do ato de jogar este jogo. Em outro nível, é o projetista das Fábulas de Jogos.

Estas experiências são criadas, discutidas e projetadas de acordo com os métodos mais diversos, contudo, acabam sempre sendo organizadas em um documento. Os profissionais da área chamam este documento de documento de game design (game design document - GDD), ou Documento de Game Design. Neste trabalho vamos chamar de Documento de Projeto de Jogo [DPJ]. Um documento desta natureza documenta ideias, conceitos, rascunhos e possibilidades de jogos. O Documento de Projeto de Jogo é a materialização do processo de criação do projeto de um jogo. Esta documentação pode ser construída de diversas formas, tanto individualmente para jogos pequenos e independentes, quanto para grandes jogos comerciais.

De acordo com (RABIN, 2012), “os conceitos de jogo normalmente não são fruto da imaginação de um designer. Em geral, são decisões empresariais lógicas ou óbvias baseadas sucessos anteriores ou negócios”. Apesar desta afirmação ter uma relação direta com grande indústria games, e não com o movimento de games independentes, que pode sim produzir jogos de sucesso que são produzidos por um pequeno time ou até mesmo por uma única pessoa. E é o game designer a pessoa responsável pela produção da documental conceitual do jogo.

Segundo (ROGERS, 2012), “o papel do designer é sempre o mesmo: criar as ideias e regras que compreendem o jogo”. E como estas ideias relativas aparecem, acontecem ou nascem? (JOHNSON, 2011) nos diz que “ideias surgem dentro de seus ambientes de trabalho e em torno deles, com todas as pressões, distrações e responsabilidades cotidianas, além da constante supervisão, que a vida profissional tantas vezes envolvem”. A documentação de

uma ideia serve para começar o processo de materialização de algo intangível. E podemos dizer que este “intangível” é o início da formulação de uma Fábula de Jogo.

Tudo gira em todos das ideias. Estas ideias precisam ser de alguma forma implementadas, transformadas em código para depois serem jogadas. As ideias precisam então se transformar em um jogo digital. São estas ideias que ajudam a compor o conceito do jogo. Mas o que é um conceito de jogo? Para (CHANDLER, 2012), “um conceito de jogo começa com um pergunta e um problema a ser resolvido”. Por exemplo: pode um Zumbi de Laser derrotar um Cowboy do Espaço?

De acordo com (SCHELL, 2011), “antes de começar a propor ideias, você precisa certificar-se por que está fazendo isso, e formular um problema é uma maneira de afirmar isso com clareza. Bons enunciados do problema informam tanto seu objetivo como suas limitações”. Tanto este objetivo, quando as limitações propostas por Schell podem ter um grande impacto no processo de produção de um jogo digital.

(SCHUYTEMA, 2008) explica que “primeiro, o conceito essencial é delineado em um breve documento de visão geral que explica os princípios mais importantes do games e alguns de seus principais recursos”. Contudo, não podemos perder de vista, conforme afirma (SCHELL, 2011) “documentos de jogos têm exatamente duas finalidades: memória e comunicação”.

Mas como testar então os objetivos e limitação das ideias propostas durante o processo de criação? Como validar o que foi documentado no Documento de Projeto de Jogo [DPJ] para que se possa ter certeza de que, o que está ali escrito, em linguagem natural, pode ser codificado e ser transformado em um game e em linguagem de máquina? São estas ideias de um jogo que, ao entrar no processo de produção, são testadas, codificadas e exploradas a fim de buscar a solução concreta para algo abstrato.

Mas cabe aqui perguntar: existem problemas relativos à documentação para a produção de um jogo digital?

1.4 Sentido da Documentação e Problemas Invisíveis

Entendendo a produção de um jogo digital como um projeto de software, com uma declaração e intenção de produção documentada do início do projeto, com os registros das ideias que surgem a medida que o game e o projeto começa a ser concebido, vemos que a documentação passa a ter um valor fundamental, pois é nela que estará contida os principais registros das mais diversas ideias a serem trabalhadas.

Contudo, a documentação não reflete o jogo em sua totalidade. Podemos dizer até

que a documentação em nada reflete o jogo final, pois não podemos jogar uma documentação. Melhor dizendo: o Documento de Projeto de Jogo [DPJ] não é jogável. A documentação serve seu propósito de comunicar as principais ideias e arquivar a memória do projeto. O DPJ representa o grande alicerce do projeto e, de acordo com (LEMES, 2009), “este documento é a espinha dorsal de todo e qualquer projeto de um game. É o que define todos os pontos de um jogo digital e guia todas as equipes envolvidas no processo de produção de um game.”

Para (FEIJÓ; CLUA; SILVA, 2010), “antes de desenvolver um jogo, deve-se produzir um documento de design que irá conter a descrição da jogabilidade, portanto as regras em detalhes. Esse documento pode ser encarado como uma espécie de manual de instruções para os futuros desenvolvedores do jogo. De fato, tão importante é esse documento, que o processo de desenvolvimento não pode começar sem ele”. Aqui o autor define o DPJ como documento de design, mas, em suma, é o mesmo tipo de documentação.

A importância está no fato de ser um documento que dá origem a todo o processo de criação e desenvolvimento do game, registrando todas as ideias, detalhes e pormenores que darão origem ao jogo digital que começa então a ser planejado. E o planejamento é o início do desenvolvimento.

Toda esta documentação tem o seu papel importante em um projeto, pois é nela que se apoiarão decisões das mais diversas. (BROOKS, 2009) afirma que “em muitos projetos de software, as pessoas começam a organizar reuniões para debater a estrutura e, depois, começam a escrever os programas. Não importam as dimensões do projeto. O gestor começa de imediato a formalizar, ao menos, pequenos documentos que servirão como sua base de dados. E ele acaba precisando de documentos muito parecidos com aqueles dos quais muitos outros gerentes dependem”.

E qual o sentido da documentação formal de software? Para (BROOKS, 2009) “em primeiro lugar, porque escrever as decisões é fundamental. Apenas quando se escreve é que as lacunas aparecem e as incoerências afloram. O ato de escrever acaba por requerer uma centena de pequenas decisões, e a existência delas distingue as políticas claras e exatas daquelas confusas. Em segundo lugar, porque os documentos comunicarão as decisões para outros. O gerente será continuamente surpreendido pelo fato de que as decisões que ele tomou para o conhecimento de todos são totalmente desconhecidas por algum membro da equipe. Uma vez que seu trabalho fundamental é manter todos na mesma direção, sua principal tarefa diária é a comunicação, não a tomada de decisão, e seus documentos irão aliviar imensamente essa carga”.

Documentação faz sentido em qualquer projeto de desenvolvimento de jogos, mas ela é fundamental? A documentação, por mais essencial que seja, também apresenta seus

problemas. A documentação, escrita em formato de texto e podendo conter elementos gráficos, contrapõe-se ao jogo digital, que é jogado.

Em primeiro lugar, podemos nos ater a questões burocráticas. Excesso de documentos podem burocratizar um processo que, antes de mais nada, precisa ser criativo. Em segundo lugar, podem causar interpretações diferentes de um mesmo ponto de vista, como sugere (FRIED; HANSSON, 2012) “o problema das abstrações (como relatórios e documentos) é que elas criam a ilusão de que todos chegam às mesmas conclusões sobre o que leem. No entanto, 100 pessoas podem ler as mesmas palavras e pensar, cada uma, uma coisa diferente. Por isso, o melhor é ter algo real logo de cara. Só assim se obtém uma compreensão verdadeira”.

A documentação de um projeto de um game pode variar em forma e tamanho, pois não existe uma fórmula fechada para sua redação, o que existem são modelos variados usados nas mais diversas situações e tipos de jogos. (ROGERS, 2012) diz que “um Documento de Game Design deveria ser longo o suficiente para descrever precisamente o que acontece um um jogo”. Esta descrição pode ter o tamanho de 10 páginas ou 300, dependendo do projeto.

Mas aqui um ponto é fundamental observar: independente de sua forma ou tamanho, este documento tem como objetivo a comunicação entre toda a equipe de desenvolvimento e o registro de ideias e decisões tomadas durante sua preparação.

A documentação também serve para ajudar o desenvolvedor final, ou o time de programadores, que precisam trabalhar na construção do produto digital que terá como base toda a documentação produzida e validada. De acordo com (FARREL, 2010), “quando programadores começam a planejar a lógica de um programa de computador, precisam de instruções conhecidas como documentação de programa. Usuários finais nunca veem a documentação de um programa; programadores usam-na ao planejar ou ao mesmo modificar programas”. Isso mostra então a importância da documentação, mesmo que este processo apresente diversas interpretações, como citado anteriormente.

Mas aqui temos então uma dicotomia interessante e que será explorada no decorrer desta tese. Se a documentação não pode ser jogada, e é um processo de registro das ideias a serem desenvolvidas e implementadas, o DPJ pode ser visto como o momento inicial da especificação de software, que, para (SOMMERVILLE, 2011), é onde a “funcionalidade do software e as restrições a seu funcionamento devem ser definidas”.

O DPJ trata das abstrações do game e dos processos de criação sem a preocupação de limitar tecnicamente (inicialmente) como o jogo será construído. Mas este documento deve ser encarado com um documento em constante evolução e versões. E a primeira versão deve conter o registro das principais ideias com a finalidade de apresentar a Fábula de Jogo.

Para (RABIN, 2012), “antes que o trabalho sério possa começar, o conceito central do

jogo deve se escrito. A forma segue a função, ou seja, o propósito do documento de conceito dita a forma que este terá. Este documento serve a vários propósitos, mas na fase de conceito, seu principal objetivo é comunicar uma visão do jogo para que todas as partes envolvidas possam entrar num acordo sobre ele”.

Uma vez que as partes envolvidas já conheçam os elementos conceituais presentes, este documento serve fazer parte do processo de planejamento do jogo digital que, para (CHANDLER, 2012), “o planejamento do jogo é onde as informações são reunidas mostrando como tudo será realizado”.

Com o Documento de Projeto de Jogo [DPJ] produzido, qual a próxima fase dentro do processo de desenvolvimento de um game? Partir para o desenvolvimento e implementação final ou testar as ideias presentes nesta documentação?

Testar as ideias presentes no DPJ faz com que se ganhe tempo, eliminem-se os erros de abstração, delimite-se o que será codificado, analise as possibilidades conceituais presentes e garanta que o projeto esteja o mais próximo possível de suas viabilidades técnicas.

Ressaltando que no DPJ estará presente toda a Fábula de Jogo, ou seja, a sequência de acontecimentos jogáveis que formam a narrativa e a experiência de jogo.

E como testar as ideias presentes no Documento de Projeto de Jogo [DPJ]? Como determinar que elementos presentes na Fábula de Jogo [FDJ] e documentados no DPJ sejam passíveis, ou não, de serem codificados a ponto de serem transformados em um jogo digital?

Usando a Programação Exploratória [PXP], como veremos no próximo capítulo.

Parte II

Programação Exploratória e Fábula PXP

2 Programação Exploratória de Computadores

O artista só tem um meio de representar-se a obra na qual trabalha: é preciso que ele a faça.

— Maurice Merleau-Ponty

2.1 Algoritmos e Programação

O que é programar um computador? Como se dá a programação de computadores? Qual o conhecimento necessário para que uma pessoa possa criar códigos para uma máquina funcionar de acordo com suas necessidades? Para (BROOKS, 2009), “um programa de computador é uma mensagem de um homem para a máquina. A rígida sintaxe marcial e todas as escrupulosas definições existem para tornar absolutamente claros os comandos para o estúpido mecanismo. Mas um programa, uma vez escrito, tem outra face, aquela que conta sua história ao usuário humano. Pois mesmo para o mais privado dos programas, alguma comunicação é necessária”.

De acordo com (FEIJÓ; CLUA; SILVA, 2010), “programar não é lidar com hardware, é lidar com processos, raciocínio, métodos e estratégias de desenvolvimento envolvendo artefatos abstratos”. E, um desses artefatos, ou o mais importante deles, é a lógica de programação.

Uma das definições que nos remete à lógica de programação é “a maneira pela qual se representa em linguagem corrente, ou por meio de símbolos que expressam essa linguagem, instruções, que compõem um programa a ser executado por um computador” (CARBONI, 2003). De acordo com (PFAFFENBERGER, 1998), “programação é um processo que consiste em especificar instruções que indicam ao microprocessador o que deve ser feito”. O conhecimento da lógica de programação é um dos passos na direção de como se dá a programação de computadores, mas não é o único.

O conhecimento da lógica¹ é um conhecimento que antecede, portanto, a lógica de programação. E um dos conceitos fundamentais que norteiam a lógica de programação é o conceito de algoritmo. Para isso é preciso entender e compreender que “um algoritmo é

¹ “Lógica é a ciência que estuda princípios e métodos de inferência, tendo o objetivo principal de determinar em que condições certas coisas e seguem (são consequência), ou não, de outras” (MORTARI, 2001).

um conjunto finito ordenado de passos executáveis não ambíguos de um processo que tem um término” (BROOKSHEAR, 2000). Uma forma que se usa para ilustrar o conceito de algoritmo é o encadeamento de instruções básicas para se realizar uma atividade, ou seja, uma sequência de passos para se realizar algo com um objetivo definido.

Contudo, o conceito de algoritmo vem sendo trabalhado por diversos autores e vale aqui um registro de diversas definições, mas que em suma, acabam por dar o mesmo sentido ao significado da palavra algoritmo:

“Algoritmo é uma sequência de passos que visa atingir um objetivo definido” (FORBELLONE; EBERSPÄCHER, 1993).

“Algoritmo é a descrição de uma sequência de passos que deve ser seguida para a realização de uma tarefa” (ASCENCIO; CAMPOS, 2008).

“Algoritmo é uma sequência finita de instruções ou operações cuja execução, em tempo finito, resolve um problema computacional, qualquer que seja sua a instância” (SALVETTI; AZEVEDO, 1972).

“Algoritmo são regras formais para a obtenção de um resultado ou da solução de um problema, englobando fórmulas de expressões aritméticas” (MANZANO; OLIVEIRA, 2000).

“Ação é um acontecimento que, a partir de um estado inicial, após um período de tempo finito, produz um estado final previsível e bem definido. Portanto, um algoritmo é a descrição de um conjunto de comandos que, obedecidos, resultam numa sucessão finitas de ações” (FARRER, 1989).

Mas quais elementos formam então um algoritmo? “Todo algoritmo é composto por instruções finitas e bem definidas que tem como objetivo de resolver um problema proposto. É um caminho para a solução de um problema. As instruções escritas em linguagem de programação² são chamadas de programas” (CARBONI, 2003).

E qual a importância de um algoritmo no processo de programar um computador? De acordo com (FEIJÓ; CLUA; SILVA, 2010), “o problema central da computação consiste em encontrar algoritmos que solucionem diversos problemas da melhor forma possível para as mais diversas situações do dia a dia”. E a Programação Exploratória [PXP], a ser definida logo adiante, ajuda neste processo de descoberta.

Programar é fazer um computador funcionar. É dar poder a estruturas tecnológicas que sem o código para nada servem. Um computador, sem o código que o faz funcionar, seria

² Linguagem de programação é um método padronizado para comunicar instruções para um computador. É um conjunto de regras sintáticas e semânticas usadas para definir um programa de computador. Mais detalhes em: <http://bit.ly/fzAOqs>. Acessado em 05/06/12.

apenas mais um apoio de papel. Um aparelho sem suas reais funcionalidades. E programar um computador é a função do programador, profissional que transforma o pensamento abstrato e algo concreto. (BROOKS, 2009) nos diz que “o programador, como o poeta, trabalha apenas levemente deslocado de um ambiente de pensamento puro. Ele constrói seus castelos no ar, de ar, criando a partir de sua imaginação. Poucos meios de criação são tão flexíveis, tão fáceis de polir e retrabalhar, tão prontamente capazes de produzir grandes estruturas conceituais”.

Programar, de certa forma e dependendo do ponto de vista, é ter um grande poder nas mãos. Conhecer o processo e conseguir aplicá-lo da forma correta para atender a necessidade de um problema específico é muito valorizado na época em que estamos vivendo.

“É o aspecto mole, impalpável e simbólico o verdadeiro portador de valor no mundo pós-industrial dos aparelhos. Transformação de valores; não é o objeto, mas o símbolo que vale. Por conseguinte, não mais vale a pena possuir objetos. O poder passou do proprietário para o programador de sistemas. Quem possui o aparelho não exerce o poder, mas quem o programa e quem realiza o programa. O jogo com símbolos passa a ser o jogo do poder” (FLUSSER, 2002).

Programar é criar um programa. Criar um programa é codificar a máquina para executar uma tarefa predeterminada. Computadores são máquinas programáveis e só podem funcionar e executar os programas nele contidos por intermédio da computação, do cálculo, dos procedimentos lógico- matemáticos da máquina. De acordo com (MARÇULA; FILHO, 2000), o “computador é uma máquina que recebe e trabalha os dados de maneira a obter um resultado. Para realizar isso, ele é programável, ou seja, responde a um grupo de comandos específicos (instruções) de uma maneira bem definida e pode executar uma lista pré-gravada desses comandos. Esta lista é chamada de programa”. E o programa de computador — também chamado de software — é “um conjunto de instruções que, quando executadas, produzem o desempenho desejado e dados que permitem que os programas manipulem adequadamente a informação” (PRESSMAN, 2011).

Podemos chamar então este conjunto de instruções de passos computacionais, ou por que não dizer passos de Turing³. O passo computacional é a manifestação da execução de uma instrução do algoritmo. Trabalhar na criação e no desenvolvimento de um software é então criar, organizar, reordenar e estabelecer passos computacionais.

Mas como devemos então começar o trabalho de criação e desenvolvimento de um

³ Conforme definição de Marcond Marchi, no Grupo de Estudo de Modelagem em Software da PUC-SP, onde o pesquisador afirma que os passos computacionais podem ser chamados também de passos de Turing. Alan Mathison Turing, matemático, lógico e cientista da computação britânico, influente e fundamental no desenvolvimento da ciência da computação, formalizou o conceito de algoritmo e computação com a máquina de Turing.

software, seja ele de qualquer natureza? Quais informações são necessárias para se começar o desenvolvimento? Para (BROOKS, 2009), “a essência de uma entidade de software é um construto de conceitos que se interligam: conjuntos de dados, relações entre itens de dados, algoritmos e chamadas de funções. Essa essência é abstrata, pois o construto conceitual é o mesmo sob muitas representações diferentes”.

O correto seria dizer que o desenvolvimento deveria começar pelos processos estabelecidos pela Engenharia de Software⁴, mas este não é o foco deste trabalho.

O desenvolvimento técnico de um jogo digital pode começar com a criação um artefato que seja executável em um computador (que não pode ser confundido com um protótipo de jogo), após a documentação ser escrita e a Fábula de Jogo [FDJ] ser criada usando como base o Documento de Projeto de Jogo [DPJ].

O DPJ registra as ideias, conceitos, narrativa e outros pontos. Mas estas ideias precisam então ser testadas por uma máquina. Uma forma de realizar estes testes tendo como base o DPJ é usando para isso a Programação Exploratória.

2.2 Programação Exploratória

O que é Programação Exploratória [PXP] e qual a sua utilidade? O PXP ajuda na resolução de problemas ou na produção de software? Como podemos usar a Programação Exploratória?

Tendo como base o Documento de Projeto de Jogo [DPJ], qual a próxima fase dentro do processo de produção de um jogo digital? As ideias, conceitos e elementos contidos neste documento podem ser transformados em um game?

A proposta desta tese é que, após a produção do DPJ, um documento escrito em linguagem natural podendo conter gráficos e imagens diversas, seja produzido então um Documento Executável de Jogo [DEJ], que é um documento complementar ao DPJ. O conceito de DEJ será melhor trabalhado no próximo capítulo.

E como devemos começar? Para (STAA, 1987), “é conveniente desenvolver-se um protótipo cobrindo somente as porções mais críticas do programa. Ao desenvolvermos o protótipo, estaremos aprendendo simultaneamente qual é o problema a ser resolvido e quais são as possíveis boas soluções. O resultado do desenvolvimento do protótipo é um programa parcial que dá uma solução a alguns aspectos fundamentais, porém sem preocupar-se com a qualidade

⁴ Engenharia de software é uma disciplina de engenharia cujo foco está em todos os aspectos da produção de software, desde os estágio iniciais da especificação do sistema até sua manutenção, quando o sistema já está sendo usado (SOMMERVILLE, 2011).

desta solução. O objetivo do protótipo é servir como um instrumento de aprendizado e de auxílio à definição do problema. Esta definição poderá ser avaliada e aprovada através de experimentos utilizando o protótipo. De posse da definição aprovada (pelos responsáveis pelo projeto) partimos para o desenvolvimento da versão de produção do programa. Note-se que esta versão poderá diferir bastante das soluções utilizadas pelo protótipo”.

Na raiz da prototipação sugerida por Staa está a Programação Exploratória, que trata justamente de explorar, testar e construir um software tendo como base a experimentação, o teste, a exploração do código, sem a preocupação com o resultado final, e sim com o processo de construção do software a ser desenvolvido. Este processo pode resultar em procedimentos e novos processos que atendam aos requisitos iniciais do software a ser desenvolvido.

Para entender o conceito de Programação Exploratória, é fundamental compreender o significado da palavra “explorar”, que, em suma, quer dizer examinar, analisar, estudar e pesquisar.

A Programação Exploratória trabalha na construção de um código-fonte⁵ de forma que este ato de construção seja objeto de análise, pesquisa e estudo. É um código artesanal, monolítico (estruturado) e não pode ser encarado como o código final de projeto de software. É um código de estudo a ser descartado. Para (BROOKS, 2009), “o descarte e redesenho podem ser feitos de uma vez só ou por partes, mas serão feitos. Portanto, inclua em seus planos o verbo descartar. Você terá que fazer isso de qualquer jeito”.

Para entender o que é um código monolítico, podemos analisar a definição de sistema monolítico que, segundo (TANENBAUM; WOODHULL, 2000), é a “organização mais comum dentro da estrutura de um sistema operacional. Esta abordagem poderia muito bem ser subintitulada com ‘a grande bagunça’. A estrutura é tal que não há nenhuma estrutura”. É um código experimental, um artefato que sofre alterações ao longo do tempo de estudo e de seu desenvolvimento, pois, de acordo com (BROOKS, 2009), “a entidade de software é constantemente sujeita a pressões em prol de uma mudança”.

A Programação Exploratória é essencial quando se tem um problema complexo que precisa ser compreendido e resolvido, mas ainda não existe uma solução ou recomendação já estabelecida. A PXP é uma técnica onde o próprio processo de programação e entendimento do problema gera uma resposta, em forma de código, para tal problema. Esta resposta, ou solução propriamente dita, é parte do processo de desenvolvimento e também do entendimento do problema.

⁵ Código fonte é o conjunto de palavras ou símbolos escritos de forma ordenada, contendo instruções em uma das linguagens de programação existentes, de maneira lógica. Mais detalhes em <http://bit.ly/KuoZH6>. Acessado em 05/06/12.

A técnica de Programação Exploratória exige a busca de uma solução para um problema apresentado. Este problema pode ser parte de um problema maior que no todo, pode vir a ser o software, ou o jogo digital completo. Ao ajudar a aprofundar a resolução do problema, expõe falhas e possíveis enganos. Por algumas vezes, o problema pode se apresentar de forma simples e bem-compreendido, mas programador pode não ter uma ideia clara de como alcançar sua solução. A técnica de PXP ajuda neste sentido.

Para alcançar sua solução, o programador precisa atacar o problema em busca de uma lógica que o resolva. E este problema pode ser um simples conceito apresentado na Fábula de Jogo, presente no Documento de Projeto de Jogo que precisa ser codificado e programado ou algo muito mais complexo.

2.3 PXP e o Design

Sendo então a Programação Exploratória uma técnica de construção de um código que busca o entendimento e a solução de um problema, podemos dizer que esta busca, este processo exploratório é um estudo do problema. Fazendo um paralelo com o desenho e também o design, podemos dizer que a técnica de Programação Exploratória produz um rascunho de software. Entendo aqui o rascunho como um processo do estudo e não o rascunho de um cálculo, pois para a máquina, os cálculos precisam ser exatos.

Um artista ou um designer, quando começar a planejar sua obra, seja uma ilustração, um pintura, um projeto ou um produto, o primeiro passo no processo de planejamento da execução desta obra é realizar um rascunho, um estudo, um esboço. Contudo, se estamos falando de arte, pintura e ilustração, o mínimo que se espera do artista é que ele saiba desenhar minimamente. O rascunho, o desenho, faz parte do planejamento da obra de arte. O esboço é um excelente modo de registrar ideias, pensamentos e funcionalidades que poderão ser implementadas no decorrer do desenvolvimento da obra.

De acordo com (MUNARI, 1998), “ao longo do desenvolvimento do projeto, o designer utiliza vários tipos de desenhos, desde o simples esboço para fixar um pensamento útil ao projeto, até desenhos de construção, perspectivas, axonométricas, desenho explodido e fotomontagens”. Ou seja, o esboço é parte fundamental do desenvolvimento do projeto.

Para (GOSCIOLA, 2003), “a roteirização não é a obra em si, mas um dos processos de sua criação. Para a realização de uma obra audiovisual, seja linear ou não-linear, um vídeo ou uma hipermídia, a obra é o que importa como resultado final, mesmo o roteiro seja alterado durante a sua realização. No caso de um livro ou uma pintura, comumente utiliza-se o termo esboço para a sua roteirização. O esboço pode não ser a criação de uma hipermídia,

mas certamente, é um instrumento eficiente de criação do roteirista. Para o roteiro de de um escultura. por exemplo, faz-se um esboço ou um projeto”.

Como podemos ver, o esboço é parte inerente do planejamento de uma obra, seja qual seja o seu suporte final.

Os esboços fazem parte de um processo de estudo que tem como objetivo concretizar uma ideia. E o que melhor do que experimentar muitas possibilidades para trabalhar na concretização de algo? Josef Albers⁶, artista americano nascido na Alemanha, foi um dos professores da Bauhaus⁷ e tinha como metodologia pedagógica e de trabalho, a experimentação, algo similar à Programação Exploratória. Segundo (WICK, 1989), as máximas da metodologia de Albers era “aprender, e não ensinar; o princípio do *trial and error* como condição do processo criativo; aprender descobrindo, como elemento necessário à educação da criatividade”.

Tentar e errar para descobrir. Descobrir para criar. “Construir inventando e atentar descobrindo: estes se desenvolvem - pelo menos a princípio - por meio da experimentação livre de perturbações, livre de influências, despreconceituosa, portanto, que (em primeiro lugar) consiste na experimentação lúdica” (WICK, 1989).

Voltando para o pensamento a respeito do software, mas ainda olhando para a questão da criatividade, (BROOKS, 2009) nos diz que “o software pode ser modificado com facilidade pois é pura matéria do pensamento, infinitamente maleável”.

Podemos dizer então que a Programação Exploratória trabalha na construção de um rascunho do software (entendendo rascunho como um processo de estudo e não o rascunho do cálculo), algo que servirá de base para o planejamento final do produto codificado e resolução de problemas, mas é algo que não é o software final. E, assim como se espera que um artista saiba desenhar, espera-se também que o desenvolvedor que trabalhará no código saiba programar e entenda os princípios básicos da programação de computadores. Também é preciso envolver-se plenamente na produção do código, experimentar o código e testar para inovar.

(MERLEAU-PONTY, 1999) afirma que “o artista só tem um meio de representar-se a obra na qual trabalha: é preciso que ele a faça”. Se o artista precisa viver sua obra, fazendo-a; o programador precisa vivenciar seu código, programando-o e escrevendo-o. A Programação Exploratória trabalha no fazer o código ao mesmo tempo em que ele é pensado e trabalha na busca pela solução e resolução de problemas. É um bloco de rascunho para testar ideias usando o computador como ferramenta primária de trabalho.

⁶ Josef Albers (1888 - 1976) foi professor da Bauhaus de 1923 até o seu fechamento, em 1933.

⁷ Escola de design, artes plásticas e arquitetura de vanguarda que funcionou em 1919 e 1933 na Alemanha.

Figura 3 – Estudos para a Natividade — Leonardo da Vinci



Fonte: Obra pertencente ao Metropolitan Museum of Art de Nova York

A Programação Exploratória trabalha na construção de um código que é em si mesmo um objeto de análise, pesquisa e estudo de um problema específico a ser resolvido e não de estudo da programação de computadores em si. Aprender a programar, então, é a etapa inicial do processo, que antecede portanto a Programação Exploratória. O esboço é o primeiro passo no planejamento da obra de arte (Figura 3).

(HONAN, 2012), em uma reflexão sobre o desenvolvimento de software, diz que “o código é como um poema; ele tem que seguir certos requerimentos estruturais, e ainda assim essa estrutura pode virar arte. Mas código é arte que faz alguma coisa. É a construção de algo novo a partir de nada além de uma ideia”. Usando o computador como ferramenta de produção e criação, é por meio da codificação que então podemos desenvolver e materializar

em ideia. Mesmo que esta materialização não seja matéria, e sim um conjunto de códigos armazenados e executados em uma máquina. Uma materialização virtual.

Por “codificação, entendemos digitar o código-fonte comentado, interpretá-lo completamente antes da compilação para assegurar que ele faz o que é concebido para fazer, compilá-lo para então executá-lo com outro código com base nos caos de teste informais. As palavras implementação e programação também são utilizadas para codificação. A palavra desenvolvimento algumas vezes também é utilizada, embora ela também possa ser aplicada a outras fases da codificação” (BRAUDE, 2005).

Terry Winograd⁸, em entrevista concedida à (PESSIS-PASTERNAK, 1993), afirma que “os melhores especialistas em informática têm, de fato, uma veia poética. Um programa de computador, como toda criação artística, tem muito mais possibilidades de aperfeiçoamento quando o seu criador é movido por um espírito poético”. O espírito poético aqui pode ser entendido como a vontade inata de fazer algo, de se construir uma obra por meio da codificação de uma máquina. E no processo de construção de uma codificação criativa, passa então pela Programação Exploratória.

Para HILLIS, “o computador não é apenas uma calculadora, uma câmera ou um pincel avançados; na verdade, é um aparelho que acelera e amplia nossos processos de pensamento”.

Esta ampliação, defendida por Hillis, é o que possibilita a exploração de novas formas de tratar velhos problemas.

2.4 Prática da Programação Exploratória

Quais os fundamentos necessários para se programar um computador? Ou melhor dizendo: quais os conhecimentos necessários para uma pessoa programar um computador? Dominar a linguagem de programação e os seus fundamentos, focando sobretudo nas ideias e não nos artefatos⁹. É preciso que, o praticante da Programação Exploratória tenha proficiência em programação. Raciocinar antes de programar e pensar em vários modelos.

A prática da Programação Exploratória é uma técnica de refinamento contínuo em busca da lógica para se descobrir a resolução de um problema proposto. Algumas abordagens neste sentido podem ajudar, no sentido de buscar o entendimento do todo, em busca do objetivo a ser alcançado. Para (BARBOSA, 2001), “a abordagem top-down, apresentada por

⁸ Professor de Ciência da Computação da Universidade de Stanford. Interesses profissionais e publicações estão em seu site pessoal: <http://hci.stanford.edu/winograd>.

⁹ Artefato é um dos vários tipos de subprodutos concretos produzido durante o desenvolvimento de software. Alguns artefatos ajudam a descrever a função, arquitetura e o design do software. Outros artefatos estão relacionados com o próprio processo de desenvolvimento, tais como planos de projetos, processos de negócios e avaliações de risco. Mais detalhes em <http://bit.ly/19WU1Uz>. Acessado em 27/08/13.

Wirth em 1971, é recomendada pela maioria dos especialistas e consiste no desenvolvimento por refinamentos sucessivos, exibindo a representação de cada etapa”.

A Programação Exploratória também é um processo de refinamento contínuo, mas não o refinamento do produto final e sim de um produto de um estudo que tem como objetivo solucionar algo incompleto ou em construção.

Apesar da prática da Programação Exploratória estar focada na resolução de um problema estabelecido, necessita de alguém qualificado para escrever o código que busca a resolução do problema. Um problema que deve ser refinado constantemente. (HILLIS, 2000) diz que “um programador qualificado é como um poeta que pode expressar em palavras aquelas ideias que as outras pessoas consideram inexprimíveis. Sendo um poeta, você pressupõe que há uma certa quantidade de conhecimento e experiência compartilhados com seu leitor. O conhecimento e a experiência que o programador e o computador tem em comum é o significado da linguagem de programação”.

Mas temos que colocar um ponto importante aqui: para aprender a programar, é preciso fazer. Ou seja, a Programação Exploratória só faz sentido quando já se sabe programar e parte-se então para a solução de um problema específico. E para se descobrir a lógica por trás deste problema, é fundamental que a busca da lógica seja atacada pelo teste, pelas múltiplas escolhas e pela criatividade. Mas é preciso fazer para aprender. Para (FEIJÓ; CLUA; SILVA, 2010), "assim como é impossível aprender a pintar apenas lendo um livro de arte, é impossível aprender a programar apenas lendo um livro de programação".

Mas existe uma linguagem de programação adequada para a prática da Programação Exploratória? Buscar a solução de um problema computacional é escrever um algoritmo que solucione este problema. Para (BARBOSA, 2001), “os métodos utilizados atualmente no desenvolvimento de algoritmos são baseados na programação estruturada, que tem por objetivo diminuir a complexidade do projeto de programas computacionais, sendo uma de suas propostas o uso de um número limitado de construtores lógicos. Sendo o objetivo principal da programação estruturada verificar ou garantir a corretude de um programa antes mesmo de terminá-lo”.

Em outras palavras, é preciso o bom entendimento do que se quer realizar para assim começar a fazê-lo. Mas é preciso fazer da maneira correta. Esta corretude é justamente a lógica por trás do problema. O problema só será resolvido quando o algoritmo construído esteja correto, cada etapa dos passos computacionais em seu devido lugar. Esta é uma tarefa complexa, como aponta (HILLIS, 2000): “qualquer pessoa que já tenha escrito - ou depurado - um programa sabe que dizer a um computador o que você quer fazer não é tão fácil quanto parece. Cada detalhe da operação desejada tem de ser precisamente descrito”.

Esta descrição precisa é o processo de programação do computador. Mas o processo não é certo, ou seja, para chegar nesta precisão, é preciso experimentar alguns caminhos de tentativa e erro a fim de estabelecer o melhor método de descrição. E esta descrição precisa é o algoritmo gerado a partir da proposição de um problema a ser solucionado. Uma vez solucionado o problema, este algoritmo fará pare de um programa que espera-se, que funcione plenamente. De acordo com (FEIJÓ; CLUA; SILVA, 2010), “um programa não pode ter nenhuma ambiguidade e nenhuma incorreção ou imprecisão”. Se está impreciso, ainda não é um programa. Pode vir a ser parte do estudo de um Documento Executável de Jogo [DEJ] como veremos no próximo capítulo.

É a explicação correta dos passos computacionais que fazem com que o computador faça aquilo que o programador quer que ele faça. Nem mais e nem mesmo. Como aponta (FEIJÓ; CLUA; SILVA, 2010), “os computadores são cegos e eles vão fazer as coisas exatamente conforme foram solicitados a fazer, levando exatamente ao pé da letra aqui que foi dito a eles”.

É por isso que durante o processo de Programação Exploratória muitas vezes a máquina não vai responder o que queremos e precisamos. Mas o interessante é que, a máquina fará exatamente aquilo que for orientada a fazer, ou seja, programada. Mas o que de fato ela pode fazer? Qualquer coisa, desde que seja devidamente orientada para isso. É como nos mostra (HILLIS, 2000), que diz que “a magia de um computador está em sua capacidade de se tornar quase qualquer coisa que você possa imaginar, desde que você explique exatamente o quê. O problema está em explicar o que você quer. Com a programação certa, um computador pode se transformar em um teatro, um instrumento musical, um livro de referência, um adversário no jogo de xadrez”.

E um ponto importante neste processo é a criatividade.

Programar é construir um conjunto de instruções para a máquina para resolver um problema específico e estruturado. Para (FEIJÓ; CLUA; SILVA, 2010), “aprender a formalizar e objetivar ao máximo todos os problemas e conceitos é uma das primeiras coisas que um aprendiz de computação deve fazer”. E para resolver problemas os seres humanos precisam de algo que é inerente à sua própria existência: a criatividade. Ser criativo é estabelecer relações improváveis na busca por algo novo e inesperado. Para (GLEICK, 2013), “os humanos solucionam os problemas com a intuição, a imaginação e lampejos de criatividade”.

Mas o que é a criatividade? Como ele se manifesta e como pode ser usada na técnica de Programação Exploratória? Para (OSTROWER, 1999), “a criatividade é portanto um potencial em aberto, abrangente, vindo a manifestar-se nas pessoas através de certas inclinações, interesses, aptidões. Poderíamos considerá-la também uma espécie de receptividade

interior. As pessoas são sensíveis em certas áreas de ser, parecem ter uma pele mais fina, poder ser diretamente atingidas e, sobretudo, podem auscultar um saber que aflora da intuição, da mais íntima presença da personalidade. As inclinações surgem espontaneamente, podendo voltar-se para as artes e artesanato, para as ciências sociais e humanas, para os muitos campos da pesquisa científica ou da tecnologia”.

E é possível usar a criatividade para se programar computadores?

“A arte da programação ‘gratifica anseios criativos construídos profundamente dentro de nós e deleita sensibilidades que temos em comum com todos os homens’, fornecendo cinco tipos de alegria:

- 1 - A satisfação de construir algo;
- 2 - A felicidade de se construir algo útil para os outros;
- 3 - O fascínio da montagem de objetos complexos, como em um quebra-cabeça com peças móveis que se interconectam;
- 4 - A alegria da aprendizagem constante, que vem da natureza não-repetitiva da tarefa;
- 5 - A delícia de se trabalhar em um meio tão maleável - pura matéria do pensamento - que ainda assim existe, movimenta-se e funciona de uma maneira impossível para palavras simplesmente escritas” (BROOKS, 2009).

Como podemos ver, a criatividade é um ponto importante da Programação Exploratória. Mas dentro do contexto desta tese, onde será usada de forma prática a Programação Exploratória? No Documento Executável de Jogo [DEJ] como veremos no próximo capítulo.

3 Fábula PXP

A magia de um computador está em sua capacidade de se tornar quase qualquer coisa que você possa imaginar, desde que você explique exatamente o quê.

— Daniel Hillis

3.1 Documento Executável de Jogo

Uma vez que o Documento de Projeto de Jogo [DPJ] tem a função documental, de registro do processo criativo, da Fábula de Jogo [FDJ] e também de comunicação entre os membros da equipe de desenvolvimento, qual a garantia de que este documento reflita, no jogo programado, as ideias e a narrativa ali presentes?

E mais: como determinar que os elementos presentes na Fábula de Jogo [FDJ] sejam passíveis, ou não, de serem codificados a ponto de serem transformados em um jogo digital?

O Documento Executável de Jogo [DEJ] surge para validar as ideias que foram registradas, documentadas e ordenadas no DPJ. O processo de documentação é abstrato e serve para registrar o pensamento puro do projetista do jogo. A construção do DEJ tem a função de validar estas ideias, que, por sua vez faz com que o abstrato se torne real.

As informações presentes no DPJ serão filtradas pelo responsável (que pode ser uma equipe) pela criação do DEJ. Estes dados filtrados serão combinados e recombina- dos para gerar assim o entendimento de todo o projeto, aplicar suas regras e colocar o conhecimento gerado para trabalhar em função da codificação. É neste momento que a Programação Exploratória [PXP] entra em cena.

A construção do DEJ não é o fim, e sim o meio. É parte do processo documental e criativo, ajudando na construção narrativa, validando ideias presentes na Fábula de Jogo e propondo novas soluções. É um processo iterativo de construção de um documento complementar ao DPJ.

O ciclo de trabalho pode ser representado na sequência abaixo:

Ideia inicial \Rightarrow (DPJ \Leftrightarrow DEJ) \Rightarrow **Projeto de Desenvolvimento.**

O ciclo contínuo DPJ & DEJ sugere que estes dois documentos são complementares e dependentes um do outro.

É a partir do Projeto de Desenvolvimento que se começa então a trabalhar no produto final e parte-se então para a construção dos possíveis protótipos. O protótipo não é resultado de um estudo, e sim parte do processo de produção amparado nas bases e soluções propostas pelo DEJ.

A construção do Documento Executável de Jogo [DEJ] é a materialização do pensamento do game designer, o projetista do jogo, enquanto o Documento de Projeto de Jogo [DPJ] é a expressão do seu pensamento, ideias e registro dos lampejos criativos. Da combinação DPJ & DEJ podem nascer descobertas inesperadas em função associação de ideias, dados e informações presentes no projeto.

Mas é importante frisar: o Documento Executável de Jogo não é o protótipo. O protótipo tem outra natureza, que é trabalhar em um projeto já concebido. o DEJ além de validar as ideias do DPJ, pode sugerir novas soluções criativas e técnicas, uma vez que estará em contato direto com o suporte final do jogo digital: a máquina.

Fazendo uso da técnica de Programação Exploratória [PXP], o game designer pode então contar com mais um recurso em seu processo de criação, a construção de um Documento Executável de Jogo [DEJ] que vai ajudá-lo a ir além dos estados iniciais de sua criação e seu projeto. Poderá colocar em prática o que foi concebido de forma abstrata.

Do ponto de vista prático, o projetista do jogo terá que escolher uma ferramenta de trabalho, uma linguagem de programação e uma engine de jogo para fazer este trabalho. Esta escolha é fundamental pois é via a engine de jogo escolhida que o game designer fará toda a interpretação do DPJ e trabalhará na criação do DEJ, utilizando para isso a técnica de PXP.

O Apêndice C desta tese apresenta um DPJ que foi construído com base em sessões de brainstorm (Anexo A) em sua primeira versão e passou por um processo de PXP & DEJ o que resultou em mudanças em diversos pontos da Fábula de Jogo, dando origem a uma segunda versão do DPJ, presente em sua totalidade neste anexo.

Esta segunda versão do DPJ (Apêndice C) apresenta a Fábula PXP, ou seja, a Fábula de Jogo que passou pela ação direta da Programação Exploratória [PXP] durante o processo de produção do Documento Executável de Jogo [DEJ].

Os anexos presentes nesta tese mostram de forma prática o produção de um DPJ & DEJ utilizando a engine de jogo Unity 3D¹.

¹ Unity 3D é um motor de jogo 3D proprietário criado pela Unity Technologies, mais informações em: <http://unity3d.com/pt>.

O Documento Executável de Jogo [DEJ] é portanto um documento codificado, usando para isso uma linguagem de programação e uma engine de jogo (a ser escolhida pelo projetista do jogo) que visa validar, ampliar ou alterar a Fábula de Jogo [FDJ] presente no Documento de Projeto de Jogo [DPJ] usando para isso a Programação Exploratória [PXP]. A PXP ataca o projeto, identificando as Unidades Mínimas de Fábula [UMF] e codificando-as a ponto de validar, ampliar ou alterar.

O resultado deste processo é uma Fábula de Jogo que passou pela ação direta da Programação Exploratória, dando origem à Fábula PXP.

3.2 Prototipação

De acordo com (NOVAK, 2010), “a definição usual de um protótipo no setor de games é algo como o seguinte: um item de software operacional que captura na tela a essência do que torna o game especial, o que o diferencia dos demais e o que tornará bem-sucedido”. Nos dias atuais, muitos recursos comuns em todos os jogos estão disponíveis em um grande número de engines comerciais. E é o diferencial que faz com que um jogo possa atingir a popularidade.

Mas para que serve de fato um protótipo? Qual sua função no projeto de desenvolvimento de um jogo digital? “O protótipo nos permite gerenciar as expectativas de nossos clientes envolvendo-os profundamente no projeto desde o princípio, e construir um sistema rapidamente, mas atendendo as necessidades da empresa” (ARTHUR, 1991).

Apesar da visão focada no cliente, podemos entender como cliente aqui a equipe de desenvolvimento de um jogo digital e construir rapidamente o sistema com um processo rápido de validar as soluções apresentada pelo Documento Executável de Jogo [DEJ]. E na produção de um game, o protótipo digital é a expressão e o resultado final da Fábula PXP.

Para (SOMMERVILLE, 2011), “um protótipo de uma versão inicial de um sistema de software, usado para demonstrar conceitos, experimentar opções de projeto e descobrir mais sobre o problema e suas possíveis soluções”. Contudo, devemos olhar esta experimentação e de descoberta sobre o problema diz respeito à implementação. Dentro desta visão proposta por Sommerville, a prototipação se aproxima muito do Documento de Projeto de Jogo; mas vale ressaltar que um DEJ não é um protótipo, e sim um documento codificado complementar do DPJ.

Mas um protótipo deve focar nos problemas e soluções? Se o protótipo digital de um projeto de game tem como base o Documento de Projeto de Jogo [DPJ] e o Documento Executável de Jogo [DEJ] que serviu para delinear as ideias gerais do projeto e a sequência de acontecimentos narrativos (Fábula de Jogo), problemas ainda poderão surgir durante o

processo de implementação. Pois o protótipo será a materialização das ideias e a expressão da Fábula PXP. E estas ainda ideias poderão ser validadas pela equipe de desenvolvimento do game e serve justamente para se ganhar agilidade. “Dado um protótipo, os usuários sempre poderão lhe dizer o que não gostam. Isso economiza muito esforço que poderia ter sido gasto desenvolvendo o sistema errado” (ARTHUR, 1991).

(NOVAK, 2010) apresenta a opinião de Tracy Fullerton, professora Escola de Artes Cinemáticas da USC², “os protótipos proporcionam uma oportunidade de testar ideias que o atraem, mesmo que pareçam inadequadas. São uma oportunidade de aprender e, para os designers de games, de experimentar mecânicas totalmente novas. Costumo recomendar a criação de protótipos no estágio mais precoce possível de uma ideia. Recomendo também testar esse protótipo, por mais primário ou inadequado que possa parecer”. Este testes iniciais tendem analisar se a ideia apresentada pode ou não ser inserida no jogo ou não, e é justamente a função do Documento Executável de Jogo [DEJ], validar a Fábula de Jogo.

Como esta é uma nova proposta de trabalho, a palavra protótipo tende a ser confundida com um DEJ, mas são artefatos diferentes, apesar de toda a proximidade.

E como se dá o processo de desenvolvimento de um protótipo? Quais as etapas que devem ser consideradas na produção de um protótipo, independentemente do seu tamanho? (SOMMERVILLE, 2011) fornece um caminho, como vemos nas quatro etapas da Tabela 1.

Tabela 1 – Atividades de uma prototipação.

Atividade	Resultado
Plano de prototipação	Estabelecer os objetivos do protótipo
Definição geral	Definir as funcionalidades do protótipo
Protótipo executável	Desenvolver o protótipo
Relatório de avaliação	Avaliar o protótipo

Fonte: Produzido pelo autor

Os protótipos devem ser produzidos com o objetivo de testar o projeto e suas funcionalidades. Em muitos casos, basta conhecer os recursos disponíveis em uma determinada ferramenta que será usada para se resolver os problemas aparentes do protótipo e sua construção. Em função disso, muitas vezes podemos usar a prototipação rápida. De acordo com (ARTHUR, 1991), “para ter sucesso no protótipo rápido e no desenvolvimento rápido, devemos passar da codificação própria à composição de sistemas a partir de bibliotecas de ferramentas existentes”.

Assim como um DEJ pode ser construído usando uma engine de jogo, o protótipo

² Mais sobre a USC: <http://cinema.usc.edu/>

pode seguir a mesma linha de trabalho, pois uma engine de jogo com a Unity 3D apresenta um grande número de bibliotecas para situações comuns em um grande número de jogos digitais.

Mas o protótipo deve então cobrir 100% das necessidades de desenvolvimento do jogo digitais ou não? Existe um ponto onde a produção do protótipo deve ser encerrada para começa-se então a codificação final? Para (ARTHUR, 1991), “o objetivo do protótipo é entregar os 20% da funcionalidade de um sistema que atenda a 80% das necessidades do cliente”.

Portanto, é fundamental entender que, um protótipo não é um DEJ, apesar de sua semelhanças e similaridades. O DEJ está presente no processo de concepção do game e é mais uma ferramenta na mão do game designer. O protótipo está presente na fase de implementação do projeto já testadas e apresentadas no DEJ.

3.2.1 Braid e a prototipação de ideias

Braid é um jogo independente criado por Jonathan Blow e lançado originalmente para Xbox 360 em 2008, usando o serviço Xbox Live Arcade³. Nos anos seguintes ganhou versões para Windows, Mac OS X, Linux e também está presente na PlayStation Network.

O enredo do jogo é simples, porém enigmático. Trata-se de uma ideia muito explorada nos universos dos games: um homem deve salvar a princesa de um monstro. Para isso, ao longo de sua jornada no game, o personagem encontra diversas pistas que levam o jogador a interpretá-las das mais diversas formas. Apesar disso, o jogo é um game do tipo plataforma onde o jogador pode correr, pular e escalar áreas determinadas.

Um ponto importante a ser relatado aqui é o modo como o jogo foi produzido. Ao analisar a metodologia, fica claro que o autor do jogo o produziu com base na prototipação intensa e refinamentos contínuos. E esta prototipação tem uma grande aproximação com um DEJ.

A imagem abaixo apresenta o protótipo inicial da funcionalidade do game à esquerda e como o jogo ficou depois de passar pelo excelente trabalho de arte. As características funcionais permaneceram as mesmas, mas a forma ganhou uma qualidade impressionante. A arte, neste caso, veio para agregar visualmente o que antes já havia sido testado e implementado. Apesar de não usar o termo programação exploratória, o autor sugere que a prototipação foi fundamental para a criação dos mais diversos pontos do jogo como mecânica, jogabilidade e demais funcionalidades do produto.

³ Xbox Live Arcade é uma loja online que é controlada pela Microsoft e usada para distribuir digitalmente jogos digitais, a maioria jogos casuais (em contraste aos jogos 3D comuns)



Figura 4 – O protótipo de Braid e sua versão final.

A agregação da arte só foi possível em função das funcionalidades propostas, e não ao contrário. O game não foi criado e programado em função da arte, e sim a arte foi feita em função da programação existente.

Na Game Developers Conferene de 2007, Jonathan Blow apresentou uma sessão intitulada Indie Prototyping, Braid and Making Innovative Games⁴ e discutiu a metodologia por trás da prototipagem de jogos indie. Ele apresentou como concebe, desenvolve e testa o conceito do game em formato jogável. E apresenta pistas quando um protótipo ganha sua forma final.

Nota-se, portanto, uma grande semelhança com a produção de um Documento Executável de Jogo com a prototipação proposta e praticada por Blow.

⁴ Para mais informações e para assistir a apresentação, acesse: <http://bit.ly/1whQaq2>. Acessado em 17/01/2015

3.3 Linguagens de programação e Engines

A linguagem de programação⁵ tem um papel importante no processo de construção do Documento Executável de Jogo [DEJ] e também na prototipação, pois a linguagem escolhida pode, por si só, resolver problemas conceituais presente no Documento de Projeto de Jogo [DPJ] que de outra forma poderia exigir um tempo maior.

Além da escolha da linguagem de programação, é fundamental a escolha da ferramenta que auxiliará na produção do Documento Executável de Jogo [DEJ] e suas diversas versões. Esta escolha pode fazer com que o game designer ganhe tempo na solução de problemas e encontrar maneiras de propor novas possibilidades de jogo, alterando assim sua ideia original. No Apêndice C desta tese é apresentado um Documento de Projeto de Jogo [DPJ] com comentários que nasceram após a ação da Programação Exploratória [PXP] usada na construção do Documento de Projeto de Jogo [DEJ]. Para a construção do DEJ foi usada a engine de jogo Unity 3D e a linguagem de programação C#⁶.

Uma das grandes vantagens da Unity 3D, escolhida para este experimento, está em seu editor visual, que torna possível visualizar todo o projeto do jogo e criar um game completo a partir do ambiente de desenvolvimento da própria engine.

A escolha da linguagem é importante durante a construção do DEJ, contudo, mais importante ainda é conhecer esta linguagem para poder trabalhar na construção do Documento Executável de Jogo.

Para (KERNIGHAN; PIKE, 2000), “assim como os idiomas naturais, as linguagens de programação tem idiomas, maneiras convencionais pelas quais os programadores experientes escrevem código comum. Uma parte central do aprendizado de qualquer linguagem é o desenvolvimento da familiaridade com seu idioma”. E a melhor de desenvolver esta familiaridade é, além de estudar, fazer algo funcional e dentro do contexto do processo proposto nesta pesquisa, é justamente construir um DEJ.

(HILLIS, 2000) explica que “como uma linguagem humana, uma linguagem de programação possui um vocabulário e uma gramática, mas, ao contrário da linguagem humana, cada palavra e frase da linguagem de programação tem um significado exato”. Este significado exato pode ser o diferencial no momento que o desenvolvedor escolhe a linguagem de

⁵ Linguagem de programação um método padrozinado de comunicação de instruções para um computador usando para isso um conjunto de regras sintáticas e semânticas usadas para definir um programa. Mais informações em: <http://bit.ly/1hhPm1q>. Acessado em 17/01/2015.

⁶ C# é uma linguagem de programação orientada a objetos, desenvolvida pela Microsoft como parte da plataforma .NET. A sua sintaxe orientada a objetos foi baseada no C++ mas inclui muitas influências de outras linguagens de programação, como Object Pascal e Java. Mais detalhes em: <http://bit.ly/1t5Esx7>. Acessado em 17/01/2015

programação para realizar o seu DEJ.

Mas vale lembrar o que já foi dito anteriormente, é preciso saber programar para assim escolher a linguagem de programação a ser usada em um Documento Executável de Jogo. Como aponta (FEIJÓ; CLUA; SILVA, 2010), "aprender a programar e a escrever algoritmos é algo que independe da linguagem (de programação), uma vez que há muitas semelhanças entre elas".

A linguagem de programação escolhida estará à serviço da Fábula de Jogo presente no Documento de Projeto de Jogo. E é com esta linguagem de programação e a engine escolhida que o projetista do jogo construirá um Documento Executável de Jogo usando para isso a programação exploratória, dando origem então à Fábula PXP.

3.4 Fábula PXP

A Fábula PXP é uma importante aliada do projetista de jogos quando este estiver trabalhando no projeto de um jogo digital.

Fábula PXP é a Fábula de Jogo [FDJ] que passou pela ação direta da Programação Exploratória [PXP] durante o processo de produção do Documento Executável de Jogo [DEJ].

O Documento Executável de Jogo [DEJ] é um documento codificado, que em sua construção foi usada uma linguagem de programação e uma engine de jogo (a ser escolhida pelo projetista do jogo) que visa validar, ampliar ou alterar a Fábula de Jogo [FDJ] presente no Documento de Projeto de Jogo [DPJ], usando para isso a Programação Exploratória [PXP].

A PXP ataca o projeto, identificando as Unidades Mínimas de Fábula [UMF] e codificando-as a ponto de validar, ampliar ou alterar a Fábula de Jogo, dando origem à Fábula PXP. A união de uma sequência de UMFs forma a Fábula de Jogo. E a Fábula de Jogo alterada, em função da ação da PXP, forma então a Fábula PXP.

E sabendo que, cada vez mais e mais pessoas se interessam pelo desenvolvimento de games, a Fábula PXP, defendida nesta tese, tem como objetivo ajudar na produção de projetos de jogos que trazem consigo a lógica contida nas ideias originais do jogo para testar e assim a validar as ideias contidas no Documento de Projeto de Jogo [DPJ].

A técnica de PXP então tem como objetivo ajudar na validação da narrativa contida na Fábula de Jogo, transformando-a em diversas direções.

A descoberta da lógica das ideias abstratas apresentadas em um DPJ faz com que a Fábula PXP seja uma técnica importante para o planejamento de um jogo digital desde os

seus momentos iniciais de concepção.

Além disso, ao aplicar técnica de Programação Exploratória [PXP] na Fábula de Jogo [FDJ] presente e documentada no Documento de Projeto de Jogo [DPJ] na produção de um Documento Executável de Jogo [DEJ] é possível, além de validar as ideias e a narrativa ali presentes, propor alterações e melhorias na FDJ em função da ação da PXP no DEJ. E esta técnica pode ajudar muito o projetista de jogos, tendo em vista que este profissional poderá usar uma técnica de programação para ajudar em seu projeto e processo de concepção criativa.

Isto significa que, uma vez que a Fábula de Jogo [FDJ] passa pela ação da PXP ela pode sofrer alterações e melhorias, criando-se assim a Fábula PXP, ou seja, a Fábula de Jogo que passou pelo ataque da técnica de Programação Exploratória que, validou alguns itens e propôs diversas melhorias, alterações na narrativa e identificação de pontos de mudança ou até de limitações técnicas.

Quais as vantagens de se usar a Fábula PXP?

- As ideias apresentadas no Documento de Projeto de Jogo [DPJ] são testadas e passam por uma validação usando a técnicas da Programação Exploratória [PXP]. Com o uso desta técnica, as ideias podem ser ampliadas, de acordo com o desenrolar da construção da narrativa, modificando a Fábula de Jogo. Uma Fábula de Jogo alterada em função da Programação Exploratória, dará origem à Fábula PXP;

- A técnica de Programação Exploratória [PXP] formaliza a ideia a ser especificada no Projeto de um Jogo Digital, tendo em vista que a especificação é a descrição sem algo, sem ambiguidades.

- O projetista de jogos ganha um aliado importante, ou seja, usar o suporte computacional e uma técnica de programação para testar o que foi concebido de forma abstrata.

A Fábula PXP responde portanto a pergunta que motivou esta pesquisa: como determinar que elementos presentes na narrativa criada em um projeto de um game sejam passíveis, ou não, de serem codificados a ponto de serem transformados em um jogo digital?

4 Estudo de Caso

Para ilustrar o processo da Programação Exploratória [PXP] aplicada ao Documento de Projeto de Jogo [DPJ] resultando então no Documento Executável de Jogo [DEJ], foram produzidos uma série de documentos que visam testar o processo sugerido por esta pesquisa.

O primeiro documento produzido foi um documento que registrou o processo de brainstorm para a criação, ordenação e registro das ideias iniciais do jogo.

O **Apêndice A - A Criação do Projeto RunnerChief** apresenta como as ideias foram registradas para depois serem transformadas em um Documento de Projeto de Jogo [DPJ].

Ao lançamento da ideia base do jogo, chamado então de RunnerChief, logo desenvolveu-se não só o conceito do game em si, mas uma narrativa própria, que tornou-se uma das bases para o desenvolvimento das mecânicas principais de jogo, seguidas por sua vez de maior detalhamento no decorrer do tempo de produção, ainda descrito inicialmente apenas no DPJ [Documento de Projeto de Jogo] e, portanto, sem aplicações práticas em qualquer meio.

Nas ideia inicial do jogo, RunnerChief é um corredor que esta em cima de um telhado e tem que correr para salvar o gato encantado chamado Xilef, que não pode cair nas mãos de malvado Agrag.

Para amparar todo o processo, foi criado um **Diário de Desenvolvimento (Apêndice B)** que teve como objetivo documentar o processo de criação, desenvolvimento e projeto do jogo proposto nesta tese. No total, foram 83 dias de trabalho no desenvolvimento do projeto RunnerChief aplicando os conceitos apresentados e estudados.

Muitos itens apresentados e trabalhados no decorrer dos 83 dias passaram pela ação da direta Programação Exploratória [PXP] na Fábula de Jogo [FDJ], resultando assim na Fábula PXP.

No decorrer do processo de desenvolvimento, abrangeu-se, na ordem apresentada abaixo, os seguintes itens:

1. Documento de Brainstorm
2. Diário de Desenvolvimento
3. Documento de Projeto de Jogo [DPJ] - versão 1
4. Documento Executável de Jogo [DEJ] - 9 versões

5. Lista de PXP
6. Documento de Projeto de Jogo [DPJ] - versão 2

Ainda assim, determinados elementos do processo correram em paralelo, de forma que:

- O Diário de Desenvolvimento correu junto a todo o processo;
- Os itens 4, 5 e 6 ocorreram juntos, onde o item 4 validava os itens 5 e 6, que recebiam por sua vez adendos ou modificações em função do processo de PXP presente no item 4;
- O item 5 foi criado a partir de uma análise dos elementos da Fábula de Jogo que necessitariam passar pela ação da Programação Exploratória. Além disso, foram identificadas uma série de Unidades Mínimas de Fábula (UMF) que passaram então pela ações da PXP.

No decorrer dos 83 dias de trabalho, foi possível detectar pontos como este:

Dia 46

1. Concluiu-se a revisão e complemento do conteúdo textual do DPJ - versão 2 com os respectivos comentários de PXP abrangendo-se, nessa etapa, todos os tópicos restantes, faltando assim apenas os rascunhos de planejamento de level design a serem atualizados;
2. Retificou-se alguns erros gramaticais nos documentos DPJ - versões 1 e 2, incluindo-se ainda, em ambos, pequena frase complementar no item 7.10 Rejogando e Salvando o Jogo;
3. Modificou-se, após reunião, o foco do quarto DEJ usando a técnica de PXP, tornando esse à aplicação da nova mecânica de jogabilidade decorrente da mudança de roteiro, reiniciando-se sua produção e passando-se, assim, ao próximo o cálculo base de aleatoriedade de objetos-obstáculo;
4. Foram adicionados, ao quarto DEJ usando a técnica de PXP, os sete assets de personagens referentes aos gatos da Brigada Arco-íris a serem controlados pelo jogador.

Os elementos apresentados no **Dia 46** de trabalho só foram possíveis em função da construção de DEJs sequenciais usando a técnica de Programação Exploratória. O **Apêndice B** mostra detalhadamente todos os dias de trabalho.

O **Apêndice C - Documentação do Projeto RunnerChief** apresenta um Documento de Projeto de Jogo que serviu como registro da primeira versão do documento (Pré PXP), ainda textual e sem a ação da Programação Exploratória [PXP], usada para a criação dos Documentos Executáveis de Jogo (Pós PXP). O mesmo documento registrou os pontos que sofreram as alterações do processo. Este pontos foram identificados como Pré PXP e Pós PXP.

No DPJ (Apêndice C) do projeto, detalhes de todos os itens, inclusive lógicas para solução de problemas foram apresentados.

A apresentação foi assim definida:

O projeto, chamado de RunnerChief, tem como principal objetivo o auxílio na produção da pesquisa de Doutorado na qual aborda-se o tema PXP [Programação Exploratória], como processo de validação de conceitos e ideias de um jogo por meio de testes concretos em DEJs pontuais, de forma a confirmar ou solicitar alinhamento de tais ideias à capacidade tecnológica e de produção.

RunnerChief, portanto, a princípio, busca colocar em prática as técnicas que vêm sendo pesquisadas, a fim de comprová-las ou melhor direcioná-las.

Diversos pontos do projeto foram alterados no decorrer do desenvolvimento e foram assim registrados no DPJ:

Pré PXP: Annes cozinhando magicamente em sua cozinha fantástica e à frente, os botões Jogar, Loja de Utensílios Mágicos, Especialidades do Chef, Cardápio e Sair

Pós PXP: Annes, junto a oito bichanos, cozinhando magicamente em sua cozinha fantástica e à frente, os botões Jogar, Loja de Utensílios Mágicos, Especialidades do Chef, Cardápio e Sair.

A construção do Documento Executável de Jogo, em suas nove versões presentes nesta pesquisa, também influenciaram a mecânica de jogo, como ilustrado abaixo:

Pré PXP:

Pontuação: o cálculo interno

O cálculo interno de pontuação é feito com base nas metas principais do jogo, que são:

1. Salvar o maior número de Gatos Mágicos possível;
2. Acumular o maior número de pontos, coletando as Moedas de Fliperama Mágicas, possível;

3. Alcançar a maior distância possível.

Dessa forma, cada Gato Mágico possui um valor de pontuação, bem como cada Moeda de Fliperama Mágica.

A fórmula então seria: $Pt = Gm + Pmfm + Dmp$

Onde:

“Pt” = Pontuação Total;

“Gm” = Gatos Mágicos (com seus diferentes valores);

“Pmfm” = Pontuação de Moedas de Fliperama Mágicas (também com seus diversos valores).

“Dmp” = Distância Máxima Percorrida.

Pós PXP:

Pontuação: o cálculo interno:

O cálculo interno de pontuação é feito com base nas metas principais do jogo, que são:

1. Coletar o maior número de Novelos Mágicos possível;

2. Acumular o maior número de pontos, coletando as Moedas de Fliperama Mágicas, possível;

3. Alcançar a maior distância possível.

Dessa forma, cada Novelo Mágico possui um valor de pontuação, bem como cada Moeda de Fliperama Mágica.

A fórmula então seria: $Pt = Nm + Pmfm + Dmp$

Onde:

“Pt” = Pontuação Total;

“Nm” = Novelos Mágicos (com seus diferentes valores);

“Pmfm” = Pontuação de Moedas de Fliperama Mágicas (também com seus diversos valores).

“Dmp” = Distância Máxima Percorrida.

O processo de trabalho

Usando a técnica de Programação Exploratória [PXP] na produção de um jogo digital, iniciou-se então o desenvolvimento do primeiro DEJ [Documento Executável de Jogo], com a

finalidade de testar elementos-chave do composto teórico que era nesse momento RunnerChief, descobrindo-se assim suas eventuais funcionalidades e disfunções no campo prático. E eis que, ao término da produção do terceiro DEJ, focado na busca por um cálculo funcional de aleatoriedade para a mudança de temas de cenário e seus elementos, verificou-se concretizada uma dinâmica de jogo não imaginada na concepção da ideia: Annes, o protagonista até então, encontrava-se notavelmente suscetível de queda das muitas plataformas de que eram compostos os variados cenários.

Essa facilidade resultava em um número de game overs mais frequente do que se gostaria, intrincando a progressão da jogabilidade e, assim, elevando o risco de frustração do jogador. Em face disso, algumas possibilidades de concerto seriam possíveis, como balanceamento detalhado da movimentação do protagonista, da progressão de velocidade do jogo e do posicionamento das plataformas de cenário, por exemplo. Todavia, o modo como Annes era “morto” e instantaneamente instanciado logo após no jogo, dentro dos DEJs, mostrou-se de alguma forma uma interessante ponta solta a ser não modificada, mas antes explorada de modo a contribuir para o enriquecimento e diferenciação da jogabilidade. Assim, após relacionar algumas ideias, surgiu a ideia de inverter o protagonismo da narrativa. Em lugar de Annes, o cozinheiro mágico, correr para salvar seus muitos gatos mágicos com a ajuda de Xilef, que lideraria sete dos muitos gatos mágicos para salvar o cozinheiro.

Nota-se claramente no ponto acima a presença da Fábula PXP.

Com essa simples, mas importante mudança narrativa, a jogabilidade poderia ser modificada de forma a aproveitar a dinâmica de “morte-respawn”¹ instantânea revelada no campo concreto e, assim, o jogador poderia controlar um total de sete gatos mágicos, um de cada vez, ao que quando um “morresse”, outro logo em seguida viria, concedendo então sete chances ao interator em termos de regras. Xilef continuaria cumprindo seu papel de NPC² entregador de power-ups, e alguns elementos pontuais do jogo seriam modificados sem, entretanto, alterar a essência do conceito inicial. Power-ups³ como Patins Mágicos foram substituídos por Sandálias Mágicas, e itens coletáveis como Gatos Mágicos - coerentes na narrativa de Annes - deram lugar a Novelos Mágicos, mais lógicos dentro da nova narrativa, mas com funções inalteradas. Por fim, tal mudança exigiu - e mostrou base fértil para - uma ampliação necessária ao enriquecimento do jogo. Dessa forma, obstáculos como Vassoura Mágica e Balde de Água Fria surgiram, do mesmo modo como, o raro power-up colecionável Novelo de Prata, capaz de reviver um gato “morto”, resultando em chances extras de

¹ Respawn é o momento que o personagem volta ao jogo, no mesmo local, após sua morte

² NPC: non-player character ou em português, personagem não jogável ou manipulável

³ Power-up, nos videogames, uma expressão para um item qualquer que aumenta o poder, velocidade ou outra característica de algum personagem ou veículo controlado pelo jogador. Mais detalhes em <<http://bit.ly/15JvQbS>>. Acessado em 26/01/2015.

continuação para o jogador.

Assim iniciou-se o quarto DEJ, aplicando essas mudanças até então ainda teóricas. Esse testando o funcionamento real da apelidada Mecânica da Brigada Arco-íris, o próximo, a implementação dos obstáculos repensados na dinâmica do jogo, um outro ainda, adicionando ao campo prático e ao contexto do jogo o comportamento dos power-ups renovados.

A mudança narrativa e suas alteradas mecânicas mostram-se funcionais no contexto de jogo digital e, por fim, foram validadas, permitindo a continuação segura da produção. A técnica de PXP mostrou-se, nesse caso, altamente colaborativa e prestativa ao desenvolvimento do conceito e do jogo em si, mostrando-se importante ferramenta, sem a qual poderia-se correr maior risco de se deparar com surpresas não muito agradáveis de não-funcionalidade num processo de produção a todo vapor de um jogo.

Com este experimento prático, apresenta-se então a influência da técnica de PXP na Fábula de Jogo, dando origem então à Fábula PXP.

Conclusão

O atual momento pelo qual passa a área de conhecimento que trabalha com produção e desenvolvimento de jogos digitais cresce à medida que a tecnologia de desenvolvimento se prolifera. São engines de games que automatizam diversos recursos comuns nos mais diversos jogos digitais e outras ferramentas que contribuem para o produtor de games desenvolver o seu trabalho.

Soma-se a isso uma indústria que cresce no mundo todo, inclusive no Brasil.

O **Capítulo 1** trata dos Fundamentos Conceituais para a produção de jogos digitais mantendo o foco no aspecto narrativo, no processo de projeto de jogo e na documentação de um projeto de jogo. Os fundamentos são de suma importância, pois criam o alicerce para a produção de um jogo.

Este capítulo apresenta os seguintes conceitos:

Fábula de Jogo [FDJ]: uma sequência de acontecimentos narrativos que podem ser jogados, construindo assim a narrativa presente no jogo digital.

Unidade Mínima de Fábula [UMF]: o simples acontecimento isolado, fora da sequência de acontecimentos narrativos presentes na Fábula de Jogo.

Documento de Projeto de Jogo [DPJ]: a materialização do processo de criação do projeto de um jogo digital. Um documento desta natureza documenta ideias, conceitos, rascunhos e possibilidades de jogos.

São três conceitos que tratam da fundamentação conceitual de um processo de produção de jogos digitais.

O **Capítulo 2** apresenta da técnica de Programação Exploratória e suas aplicações, técnica esta que é de fundamental importância para a exploração da Fábula de Jogo em forma de código.

Este capítulo apresenta o seguinte conceito:

Programação Exploratória [PXP]: trata justamente de explorar, testar e construir um software tendo como base a experimentação, o teste, a exploração

do código, sem a preocupação com o resultado final, e sim com o processo de construção do software a ser desenvolvido. Trabalha na construção de um código-fonte de forma que este ato de construção seja objeto de análise, pesquisa e estudo. É um código experimental, um artefato que sofre alterações ao longo do tempo de estudo e de seu desenvolvimento.

É a técnica de Programação Exploratória que trata de atacar a Fábula de Jogo e as Unidades Mínimas de Fábula para buscar a descoberta de sua lógica de funcionamento em forma de código, para ser então interpretado por uma máquina.

O **Capítulo 3** apresenta o Documento Executável de Jogo (artefato complementar ao Documento de Projeto de Jogo) e a Fábula PXP. Este capítulo tem como objetivo responder a pergunta que motivou a pesquisa: como determinar que elementos presentes na narrativa criada em um projeto de um jogo digital sejam passíveis, ou não, de serem codificados a ponto de serem transformados em um jogo digital?

O Documento Executável de Jogo [DEJ] trabalha na validação das ideias que foram registradas, documentadas e ordenadas no DPJ. O processo de documentação é abstrato e serve para registrar o pensamento puro do projetista do jogo. A construção do DEJ tem a função de validar estas ideias, que, por sua vez faz com que o abstrato se torne real.

Já a Fábula PXP é a Fábula de Jogo que passou pela ação da Programação Exploratória e sofreu alterações em relação a sua concepção original.

Este capítulo apresenta os seguintes conceitos:

Documento Executável de Jogo [DEJ]: documento codificado, complementar ao Documento de Projeto de Jogo, que tem como objetivo validar a Fábula de Jogo fazendo uso da Programação Exploratória.

Fábula PXP: é a Fábula de Jogo [FDJ] que passou pela ação direta da Programação Exploratória [PXP] durante o processo de produção do Documento Executável de Jogo [DEJ] e sofreu alterações.

A técnica de PXP então tem como objetivo ajudar na validação da narrativa contida na Fábula de Jogo, transformando-a em diversas direções.

A descoberta da lógica das ideias abstratas apresentadas em um DPJ faz com que a Fábula PXP seja uma técnica importante para o planejamento de um jogo digital desde os

seus momentos iniciais de concepção.

O **Capítulo 4** apresenta um Estudo de Caso de um projeto de construção de um jogo digital que teve como objetivo validar as ideias apresentadas nesta tese, sobretudo na apresentação prática da Fábula PXP.

Como complementação deste capítulo, existem ainda três Apêndices (A, B e C) que apresentam o projeto de criação do jogo RunnerChief, tendo como base e premissa de trabalho a Fábula PXP.

Analisando o cenário de desenvolvimento de jogos digitais no Brasil e o grande número de Faculdades e Universidades que oferecem cursos de graduação nesta área, a Fábula PXP pode ajudar não só os estudantes universitários a desenvolverem projetos com um maior rigor a aplicação criativa e técnica, bem como empresas iniciantes a projetarem jogos que possam a vir a se tornar grandes sucessos.

A Fábula PXP é uma resposta, uma inquietação e um modo de trabalhar um projeto de desenvolvimento de jogos digitais tendo como base não só a narrativa e a dimensão artística nesta fase do projeto, e sim aliar a esta etapa a técnica de Programação Exploratória para testar o que foi concebido de forma abstrata no momento de sua criação, alterando-a, testando-a, analisando-a e complementando-a de forma que, a arte possa influenciar a técnica e por sua vez, a técnica também possa influenciar a arte.

Referências

- ARTHUR, L. J. *Programando Shell no Unix*. Rio de Janeiro: LTC – Livros Técnicos e Científicos Editora, 1991. Citado 3 vezes nas páginas 51, 52 e 53.
- ASCENCIO, A. F. G.; CAMPOS, E. A. V. *Fundamentos da programação de computadores: algoritmos, Pascal e C/C++*. [S.l.: s.n.], 2008. Citado na página 38.
- BAL, M. *Narratology: Introduction to the Theory of Narrative*. 2nd. ed. Toronto: University of Toronto Press, 1997. Citado 2 vezes nas páginas 25 e 26.
- BARBOSA, L. M. *Ensino de algoritmos em cursos de computação*. São Paulo: EDUC, 2001. Citado 2 vezes nas páginas 45 e 46.
- BARTHES, R. *Análise estrutural da narrativa - 7ª edição*. [S.l.: s.n.], 2011. Citado na página 25.
- BRAUDE, E. *Projeto de software: da programação à arquitetura: uma abordagem baseada em Java*. Porto Alegre: Bookman, 2005. Citado na página 45.
- BREMOND, C. A lógica da narrativa. *BARTHES, Roland et al. Análise estrutural da narrativa. Petrópolis: Vozes*, 2011. Citado na página 27.
- BROOKS, F. P. *O mítico homem-mês: ensaios sobre engenharia de software*. Rio de Janeiro: Elsevier, 2009. Citado 7 vezes nas páginas 32, 37, 39, 40, 41, 43 e 48.
- BROOKSHEAR, J. G. *Ciência da computação: uma visão abrangente. – 5ª edição*. Porto Alegre: Bookman, 2000. Citado na página 38.
- CARBONI, I. d. F. *Lógica de programação*. São Paulo: Pioneira Thomson Learning, 2003. Citado 2 vezes nas páginas 37 e 38.
- CHANDLER, H. M. *Manual de produção de jogos digitais – 2ª edição*. Porto Alegre: Bookman, 2012. Citado 2 vezes nas páginas 31 e 34.
- FARREL, J. *Lógica e design de programação*. São Paulo: Cengage Learning, 2010. Citado na página 33.
- FARRER, H. *Algoritmos estruturados*. [S.l.]: Prentice-Hall, 1989. Citado na página 38.
- FEIJÓ, B.; CLUA, E.; SILVA, F. S. C. *Introdução à ciência da computação com jogos: aprendendo a programar com entretenimento*. Rio de Janeiro: Elsevier, 2010. Citado 6 vezes nas páginas 32, 37, 38, 46, 47 e 56.
- FLUSSER, V. *Filosofia da caixa preta: ensaios para uma futura filosofia da fotografia*. Rio de Janeiro: Relume Dumará, 2002. Citado na página 39.
- FORBELLONE, A. L. V.; EBERSPÄCHER, H. F. *Lógica de programação: a construção de algoritmos e estruturas de dados*. [S.l.]: Makron Books, 1993. Citado na página 38.

- FRIED, J.; HANSSON, D. H. *Reinvente sua empresa*. Rio de Janeiro: Sextante, 2012. Citado na página 33.
- GLEICK, J. *A informação: uma história, uma teoria, uma enxurrada*. São Paulo: Companhia das Letras, 2013. Citado na página 47.
- GOSCIOLA, V. *Roteiro para as novas mídias - Do game a TV interativa*. [S.l.]: Senac, 2003. Citado na página 42.
- GREIMAS, A. J. Elementos para uma teoria da interpretação da narrativa mítica. BARTHES, Roland et al. *Análise estrutural da narrativa*. Petrópolis: Vozes, 2011. Citado na página 28.
- HILLIS, D. *O padrão gravado na pedra: idéias simples que fazem os computadores funcionarem*. Rio de Janeiro: Rocco, 2000. Citado 4 vezes nas páginas 45, 46, 47 e 55.
- HONAN, M. *Como o Yahoo matou o Flickr e perdeu a internet*. 2012. <<http://gizmodo.uol.com.br/como-o-yahoo-matou-o-flickr-e-perdeu-a-internet/>>. [Online; accessed 10-Janeiro-2015]. Citado na página 44.
- JOHNSON, S. *De onde vêm as boas ideias*. Rio de Janeiro: Zahar, 2011. Citado na página 30.
- KERNIGHAN, B. W.; PIKE, R. *A prática da programação*. Rio de Janeiro: Campus, 2000. Citado na página 55.
- LEMES, D. d. O. Games independentes - fundamentos metodológicos para criação, planejamento e desenvolvimento de jogos digitais. *Programa de Pós-Graduação em Tecnologias da Inteligência e Design Digital. Dissertação de Mestrado. Orientação: Luís Carlos Petry. Pontifícia Universidade Católica de São Paulo - PUC-SP*, 2009. Citado na página 32.
- MANZANO, J. A. N.; OLIVEIRA, J. F. de. *Algoritmos: lógica para desenvolvimento de programação de computadores*. [S.l.]: Ed. Érica, 2000. Citado na página 38.
- MARÇULA, M.; FILHO, P. A. B. *Informática: conceitos e aplicações. - 2ª Edição*. Rio de Janeiro: Érica, 2000. Citado na página 39.
- MERLEAU-PONTY, M. *Fenomenologia da percepção. - 2ª Edição*. São Paulo: Martins Fontes, 1999. Citado na página 43.
- MORTARI, C. A. *Introdução à lógica*. São Paulo: Editora UNESP: Imprensa Oficial de Estado, 2001. Citado na página 37.
- MUNARI, B. *Das coisas nascem coisas*. São Paulo: Martins Fontes, 1998. Citado na página 42.
- NOVAK, J. *Desenvolvimento de games*. São Paulo: Cengage Learning, 2010. Citado 3 vezes nas páginas 29, 51 e 52.

- OSTROWER, F. *Acasos da criação artística - 2ª Edição*. Rio de Janeiro: Elsevier, 1999. Citado na página 47.
- PESSIS-PASTERNAK, G. *Do caos à inteligência artificial: quando os cientistas se interrogam*. São Paulo: Editora da Universidade Estadual Paulista, 1993. Citado na página 45.
- PFaffenberger, B. *Websters new world: dicionário de informática*. Rio de Janeiro: Campus, 1998. Citado na página 37.
- PORTELLA, O. O. A fábula. *Revista Letras*, v. 32, 1983. Citado na página 26.
- PRESSMAN, R. *Engenharia de software: uma abordagem profissional - 7ª Edição*. Porto Alegre: AMGH, 2011. Citado na página 39.
- RABIN, S. *Introdução ao desenvolvimento de games: vol. 4: a indústria de jogos: produção, marketing, comercialização e direitos autorais*. São Paulo: Cengage Learning, 2012. Citado 2 vezes nas páginas 30 e 33.
- ROGERS, S. *Level UP: um guia para o design de grandes jogos*. São Paulo: Blucher, 2012. Citado 2 vezes nas páginas 30 e 33.
- SALVETTI, D. D.; AZEVEDO, A. S. *Elementos de Programação – Fortran IV – 2ª Edição*. São Paulo: Companhia Editora Nacional, 1972. Citado na página 38.
- SANTAELLA, L. *Linguagens líquidas na era da mobilidade*. [S.l.]: Paulus, 2007. Citado na página 25.
- SCHELL, J. *A arte de game design: o livro original*. [S.l.]: Campus, 2011. Citado 2 vezes nas páginas 30 e 31.
- SCHUYTEMA, P. *Design de games: uma abordagem prática*. São Paulo: Cengage Learning, 2008. Citado na página 31.
- SOMMERVILLE, I. *Engenharia de Software*. São Paulo: Pearson Prentice Hall, 2011. Citado 5 vezes nas páginas 29, 33, 40, 51 e 52.
- STAA, A. V. *Engenharia de programas. – 2ª Edição*. Rio de Janeiro: Livros Técnicos e Científicos S.A., 1987. Citado na página 40.
- TANENBAUM, A. S.; WOODHULL, A. S. *Sistemas Operacionais: projeto e implementação - 2ª Edição*. Porto Alegre: Bookman, 2000. Citado na página 41.
- WICK, R. *Pedagogia da Bauhaus*. São Paulo: Martins Fontes, 1989. Citado na página 43.

Apêndices

APÊNDICE A – A Criação do Projeto RunnerChief

A.1 Documento de Brainstorm

Para ilustrar o processo da Programação Exploratória aplicada ao Documento de Projeto de Jogo [DPJ], que resultará então no Documento Executável de Jogo [DEJ], foram produzidos uma série de documentos que visam testar o processo sugerido por esta pesquisa.

O primeiro documento produzido foi um documento que registrou o processo de brainstorm para a criação, ordenação e registro das ideias iniciais do jogo.

[23/03/2014] Ideia inicial do jogo

Jogo estilo Jetpack Joyride¹ com visual no estilo 8 bits (Nintendo). O jogo terá apenas um único controle: o toque na tela.

Com esta premissa, já pressupõe-se que o jogo terá como suporte um tablet ou um telefone celular (smartphone).

RunnerChief é um corredor que esta em cima de um telhado e tem que correr para salvar o gato encantado chamado Xilef, que não pode cair nas mãos de malvado Agrag.

Ao longo de suas corridas, RunnerChief, cujo verdadeiro nome é Annes, acumula fichas de antigos fliperamas para melhorar seu desempenho durante o jogo. Ao correr, o personagem libera um pequeno arco-íris.

Controles

- Com um toque na tela o personagem pula;
- Com um toque mais longo na tela (segurando) o personagem se mantém um pouco mais no alto.

¹ Para conhecer o jogo, visite o endereço: <http://halfbrick.com/our-games/jetpack-joyride>

O que pode comprar com as moedas ou fichas no jogo?

- Tênis dos mais variados;
- Energéticos;
- Patins;
- Motos.

O game terá como principal referência visual os jogos de 8 e 16 bits dos anos 1980 e 1990.

[29/03/2014] Anotações para o projeto

Com base na ideia inicial, seguem-se as primeiras notas referentes ao projeto do jogo.

Nota 1: RunnerChief é um Endless Runner

Sendo RunnerChief um jogo ao estilo Jetpack Joyride, entende-se que tenha como característica principal ser um Endless Runner². Entretanto, o conceito base sugere que Annes, o protagonista, corra para salvar seu gato. Salvar o gato, a princípio, implicaria em um fim de jogo, conferindo uma característica “zerável”, onde conquista-se a meta principal e termina-se o cânone do jogo. Sendo assim, existe um conflito com objetivo inicial proposto e o gênero de jogo sugerido.

Isto posto, sugere-se uma reformulação do objetivo principal do jogo. A princípio, pensou-se em:

Annes deve correr para salvar seus muitos gatos mágicos aprisionados por Agrag em diversos lugares diferentes. Assim, Annes pode correr infinitamente, permitindo que o jogador avance no jogo o quanto puder.

Os gatos poderiam, por exemplo, ter cerca de 6 ou 8 desenhos ou estilos diferentes e apareceriam no jogo de forma mais ou menos aleatória segundo regras de geração ainda a serem definidas.

O único gato que não foi aprisionado é o esperto Xilef, que vez ou outra aparece flutuando e concede alguns power-ups (aumento de poder) a Annes, segundo regras que devem ainda serem determinadas.

² Endless Runner: conhecido com Jogo de Plataforma. Para mais informações, acesse: <http://bit.ly/13Uwwt4>

Ideias soltas 1:

1 - Neste processo de corrida, poderia ser usado movimentos de parkour?

Sim, parece ser possível usando animações pré-definidas a serem executadas de acordo com condições de jogo definidas na programação.

2 - É possível brincar com a gravidade? Correr pela parede para subir em algum prédio escalando as paredes?

A princípio, sim. Nesse caso, a própria engine cuidará dos cálculos de física requeridos para tanto, cabendo ao desenvolvedor apenas traçar as condições para tal e dizer como isso ocorrerá, por meio da programação do jogo.

Nota 2: Fliperama Mágico

A fim de que os itens coletáveis iniciais possam aderir melhor ao contexto do jogo, sugere-se que sejam algo como “Moedas de Fliperama Mágicas”. O elemento magia estaria de acordo com o universo fantasioso do jogo proposto inicialmente.

Nota 3: Magia para todos os lados

Seguindo a mesma linha das moedas de fliperama mágicas, pensou-se na mudança do item “energéticos” para “poções energéticas”, conferindo uma característica que se encaixa ao contexto mágico.

Os demais itens podem ter sua arte trabalhada a fim de comporem a identidade visual do jogo de forma sólida.

Observação:

No mais, neste primeiro momento de trabalho, as demais características, bem como os controles, parecem funcionais.

Ideias soltas 2

Obstáculos

Quanto aos obstáculos possíveis, compreendendo o gênero Endless Runner assumido pelo jogo, pergunta-se:

O jogo apresentará um número dinâmico de obstáculos, semelhante a Jetpack Joyride,

ou uma variedade mais simples, como em Canabalt?³. Depois de pensar sobre o assunto, concluí-se que o melhor é o meio termo.

Cenários

Sugere-se uma mudança de cenários mais ou menos aleatória e cíclica com diferentes temas baseados e de acordo com o contexto do jogo, como ocorrido em Jetpack Joyride.

Missões e Desafios

Imagina-se, até o momento, que a principal meta do jogador é salvar o maior número de Gatos Mágicos presente no jogo, acumulando assim a maior pontuação possível e para tanto deve permanecer vivo e alcançar a maior distância possível em uma partida. Assim, o jogador busca quebrar seu recorde a cada jogada.

Acompanhariam ainda, como ocorre já naturalmente nos jogos digitais e talvez ainda com mais ênfase na vertente mobile, os conhecidos achievements ou, conquistas.

Entretanto, para aumentar os elementos “rejogo”, “desafio” e “recompensa”, pergunta-se:

Seria interessante, no presente momento, e para o conceito de jogo proposto em específico, pensar em missões menores a serem completas pelo jogador, suscetíveis de gerar determinadas recompensas, uma vez mais como em Jetpack Joyride, onde o jogador é recompensado com estrelas e dinheiro a cada missão completa?

[02/04/2014] Anotações para o projeto

Ideias soltas 2 - Respostas

Para os obstáculos, a princípio sugere-se:

A distância natural entre alguns dos diversos telhados existentes no mapa, podendo variar estes, bem como suas distâncias, de acordo com o tema do cenário atual. Cair nestes vão implicaria em fim de jogo (Game Over) direto.

Bombas mágicas, arremessadas de forma semelhante aos mísseis em Jetpack Joyride, com a diferença de reagirem à física, possuindo dessa forma peso e podendo mesmo quicar no cenário. Tais bombas implicariam em fim de jogo (Game Over) direto.

³ Canabalt é um jogo criado por Adam Saltsman para um projeto experimental de jogabilidade, a forma como o game é jogado. Para mais informações, acesse: <http://bit.ly/1zPIihB>

Caixas de Armadilha Mágica, geradas no cenário vez ou outra, de forma um tanto rara, posicionando-se de forma semelhante às caixas de Canabalt. Ao tocar as caixas, uma reação-armadilha mais ou menos aleatória é lançada sobre o jogador, tais como:

- Perder determinadas quantidades de Moedas de Fliperama Mágicas;
- Deixar escapar um Gato Mágico;
- Explosão com Game Over. Tal explosão teria possibilidades reduzidas de ocorrer;
- Apenas um tropeço. Annes apenas perde diminuta parte de sua velocidade.

Missões e Desafios

Neste caso, os achievements iniciais poderiam ser algo na linha de:

Salvar X gatos em uma única partida;
Salvar X * 2 gatos em uma única partida;
Salvar X * 5 gatos em uma única partida.

Atingir X pontos em uma partida;
Atingir X * 2 pontos em uma partida;
Atingir X * 5 pontos em uma partida.

Salvar X gatos totais;
Salvar X * 2 gatos totais;
Salvar X * 5 gatos totais.

Ideias soltas 3

Quem é Annes?

Sendo o nome inicial do jogo “RunnerChief”, e “Chief” podendo significar variados tipos de “Chefe”, como chefe de polícia ou outros, imaginou-se o protagonista Annes como sendo uma espécie de Cozinheiro, ou Chef Mágico.

Power-ups

Para os Power Ups concedidos por Xilef, imagina-se, por hora:

- Poções Energéticas. Tais itens concederiam invulnerabilidade passageira a Annes;

- Nabo Mágico. O legume faria o jogo passar em câmera lenta por alguns segundos. O efeito é também suscetível de interrupção caso Annes entre em choque com quaisquer dos itens que implicam em fim de jogo (Game Over) direto. Pensou-se depois em ovo mágico ao invés de nabo mágico.

- Patins Mágicos. Fariam Annes flutuar, mudando a movimentação do personagem para algo semelhante à de Jetpack Joyride;

- Moto do Chefe. Uma moto mágica invulnerável a determinado número de bombas.

Pergunta: quanto ocorrer a queda nos vãos e o contato com as Bombas Mágicas, Annes perde seu veículo. Entretanto, seria interessante também a perda dos veículos ao choque com a Caixa de Armadilha Mágica? Ou, nesse caso, parece melhor manter os veículos? Neste momento pensou-se que perder o veículo é mais interessante.

APÊNDICE B – Diário de Desenvolvimento

B.1 Projeto RunnerChief

Este Diário de Desenvolvimento tem como objetivo documentar o processo de criação, desenvolvimento e projeto do jogo proposto nesta tese. No total, foram 83 dias de trabalho no desenvolvimento do projeto RunnerChief aplicando os conceitos apresentados.

Muitos itens apresentados passaram pela ação da direta Programação Exploratória [PXP] na Fábulo de Jogo [FDJ], resultando assim na Fábula PXP.

No decorrer do processo de desenvolvimento, abrangeu-se, na ordem apresentada abaixo, os seguintes itens:

1. Documento de Brainstorm
2. Diário de Desenvolvimento
3. Documento de Projeto de Jogo [DPJ] - versão 1
4. Documento Executável de Jogo [DEJ] - 9 versões
5. Lista de PXP
6. Documento de Projeto de Jogo [DPJ] - versão 2

Ainda assim, determinados elementos do processo correram em paralelo, de forma que:

- O Diário de Desenvolvimento correu junto a todo o processo;
- Os itens 4, 5 e 6 ocorreram juntos, onde o item 4 validava os itens 5 e 6, que recebiam por sua vez adendos ou modificações em função do processo de PXP presente no item 4;
- O item 5 foi criado a partir de uma análise dos elementos da Fábula de Jogo que necessitariam passar pela ação da Programação Exploratória.

Dia 1

1. Começa o processo de brainstorm com base na ideia inicial do jogo.

Dia 2

1. Foram aprovados os apontamentos descritos no dia anterior;
2. Respondeu-se algumas questões sobre possibilidades de física aplicadas ao jogo;
3. Adicionou-se imagens de referência para os Gatos Mágicos e o gato Xilef;
4. Foram adicionadas, validadas e respondidas algumas questões e sugestões sobre desafio, rejogo, recompensas e cenários.

Dia 3

1. Foram adicionadas sugestões de Power-ups, Obstáculos, Achievements e Missões e e seus possíveis funcionamentos;
2. Foram ainda sugeridos uma loja dentro do jogo e funcionamentos de Upgrade para os Power-ups.

Dia 4

1. Foram validadas as sugestões anteriormente colocadas, com exceção do item Nabo Mágico / Ovo Mágico, que ainda está sob análise para melhor resolução;
2. Foi ainda respondida a questão sobre o efeito secundário de perda de veículos do item Caixa de Armadilha Mágica.

Dia 5

1. Foi adicionada uma imagem de referência do jogo Bit.Trip Runner, sobre o arco-íris produzido por Annes.

Dia 6

1. Foi adicionada sugestão de resolução para o item Ovo Mágico / Nabo Mágico;
2. Foi adicionada uma abordagem geral sobre a situação atual do desenvolvimento da ideia e sua relação de completude com os tópicos do DPJ. Junto a isso, foram ainda colocadas algumas poucas sugestões quanto aos próximos passos.

Dia 7

1. Foram discutidos alguns elementos restantes da composição base do conceito do jogo, sendo acordados 2 destes, a saber: 2.17 projeto Artístico e 2.22 Sistema de Ajuda do DPJ. Os demais itens seguem em processo de construção.

Dia 8

1. Foram aprovados os itens restantes, 1. Sugestão de Resolução do Nabo Mágico / Ovo Mágico e 2.4 Público Alvo;
2. Foi ainda melhor explicado o item 2.8 Fluxo de Jogo adicionando-se também imagens de referência.

Dia 9

1. Iniciou-se a produção do Documento de Projeto de Jogo [DPJ], sendo preenchidos de início os itens no intervalo entre 1. Histórico do Projeto e 6.8 Escopo do Projeto.

Dia 10

1. Continuou-se o preenchimento do DPJ, dessa vez contemplando-se os itens entre 6.8.1 Número e Tipos de Cenários até 6.8.4 Número e Tipos de Objetos.

Dia 11

1. Continuou-se o preenchimento do DPJ, dessa vez completando-se os itens 7.1 Objetivos e 7.2.3 Estruturas dos Puzzles;
2. Organizaram-se ainda algumas anotações e formatações pelos itens do documento;
3. Sugeriu-se a mudança de colocação dos rascunhos de level design do item 7.3 Fluxo de Jogo, para o item 7.4 Fluxo de Telas.

Dia 12

1. Foram aprovados os itens completados entre 1. Histórico de Projeto e 7.2.3 Estruturas dos Puzzles;
2. Foi completo o item 7.5.1 A mecânica explícita.

Dia 13

1. Adicionaram-se alguns itens restantes ao tópico 7.5.1 A mecânica explícita;
2. Iniciou-se o preenchimento do tópico 7.5.2 A mecânica implícita;
3. Adicionaram-se mais um tipo de asterisco, dessa vez para notações de requerimento de PXP para preenchimento;
4. Completou-se o item 7.6 Física do Jogo, restando entretanto uma questão a ser sanada.

Dia 14

1. Continuou-se o preenchimento do tópico 7.5.2 A mecânica implícita;
2. Está em processo de resolução a questão pendente do item 7. Achievements, conquistas, do item 7.5.1 A Mecânica Explícita. Tal processo gerou ainda nova ideia para os achievements, que deve ser ainda pensada, e aprovada ou não.

Dia 15

1. Foram preenchidos e atualizados os itens de 6.6 Resumo do Fluxo de Jogo até 9.6 IA de NPCs, estando todos pendentes de aprovação e alguns suscetíveis de ampliação. Assim, restam, no momento, preencher os itens 7.3 Fluxo de Jogo e 7.4 Planilha de Fluxo de Telas.

Dia 16

1. Começou-se a produção do Fluxograma de Lógica Interna de RunnerChief, com base em todo o conteúdo desenvolvido e nas informações do item 6.6 Resumo do Fluxo de Jogo, com o objetivo de preenchimento do item 7.3 Fluxo de Jogo.

Dia 17

1. Continuou-se a produção do Fluxograma de Lógica de RunnerChief, a fim de preenchimento do item 7.3 Fluxo de Jogo.

Dia 18

1. Continuou-se a produção do Fluxograma de Lógica de RunnerChief. Terminada a construção do fluxo lógico de jogo, a imagem que contém o fluxograma está sendo trabalhada para apresentar de forma mais interessante acabamento visual.

Dia 19

1. Concluiu-se a produção dos fluxogramas lógicos e de telas de RunnerChief;
2. Foram revistos, ampliados e completados itens diversos do DPJ;
3. Foi revisada a formatação geral do texto, bem como o funcionamento dos links contidos no DPJ;
4. Foi concluída a primeira versão do DPJ, estando essa sujeita à aprovação.

Dia 20

1. Iniciaram-se os desenhos de rascunho e planejamento de modelos de Level Design que estão ainda pendentes no DPJ, contemplando-se, ao início, o primeiro dos quatro temas, a Vila Mágica.

Dia 21

1. Continuaram-se os desenhos de rascunho e planejamento de Level Design, contemplando-se, nesse dia, o segundo tema de cenário, Floresta Mágica.

Dia 22

1. Terminaram-se os desenhos de rascunho e planejamento das telas de Level Design, um para cada tema de cenário, contemplando-se portanto os modelos Vila Mágica; Floresta Mágica; Nuvens Mágicas, Dia; Nuvens Mágicas, Noite;
2. Os rascunhos de planejamento de telas de Level Design foram implementados no DPJ. Assim, o documento a princípio se torna de fato completo, ficando sujeito à aprovação.

Dia 23

1. Foi aprovada a primeira versão do DPJ, adicionando-se ainda alguns comentários a serem utilizados no processo de PXP. Tal versão é chamada de versão 001, visto que, após o processo de PXP, novas versões surgirão com base na já existente, apenas adicionando-se as possíveis alterações.

Dia 24

1. Deu-se início à Lista de PXP, documento que, conforme releitura do DPJ - versão 1, busca catalogar os pontos problemáticos da ideia desenvolvida para RunnerChief, bem como suas possíveis soluções teóricas e posteriores soluções e resultados práticos. A Lista de PXP vem assim embasar o processo de PXP, podendo ainda ser atualizada toda vez que encontrada uma solução prática nesse processo.

Dia 25

1. Foi efetuada pequena modificação no item 7.9 Opções de Jogo, do DPJ - versão 1, que encontrava-se incompleto em relação ao conceito do jogo;

2. Continuou-se a confecção da Lista de PXP. Terminada a principio catalogação dos principais pontos problemáticos, a lista agora encontra-se à disposição para revisão e eventuais correções e sugestões.

Dia 26

1. Foi ampliada, revisada e completa a Lista de PXP.

Dia 27

1. Iniciou-se a produção dos Documentos Executáveis de Jogos (DEJ) usando a técnica de PXP;

2. O primeiro DEJ foi iniciado, terminado e compartilhado a fim de documentação, acompanhamento, sugestões a aprovações. Tal DEJ tem como foco testes simples de pulo e gravidade, a fim de embasar os DEJs posteriores que, por sua vez, virão responder as reais perguntas da lista de PXP sobre esses aspectos.

Dia 28

1. Iniciou-se o segundo DEJ usando a técnica de PXP, com foco, a princípio, no cálculo de progressão de velocidade de jogo.

Dia 29

1. Continuou-se a produção do segundo DEJ usando a técnica de PXP. O foco, cálculo de progressão de velocidade de jogo, estendeu-se ao cálculo de instanci-
ação e posição de elementos do cenário, por notar-se o fato de ambos estarem intimamente ligados, tendo seus funcionamentos diretamente interdependentes.

Dia 30

1. Continuou-se a produção do segundo DEJ usando a técnica de PXP. O processo, no momento, é de correção de algumas lógicas de funcionamento de código, principalmente referentes à lógica de instanci-
ação e posição de elementos do cenário. O binômio direto velocidade de progressão X delay de instanci-
ação de elementos do cenário continua intimamente ligado mas, na lógica da programação do DEJ, suas relações não parecem se dar tão diretamente quanto a princípio se imaginava.

Dia 31

1. Concluiu-se e compartilhou-se o segundo DEJ usando a técnica de PXP, alcançando-se as fórmulas base de velocidade de progressão e delay de instanci-
ação de elementos de cenário. Tais fórmulas, entretanto, são ainda suscetíveis de modificação conforme a produção de novos DEJs, e evolução da produção de RunnerChief;
2. Iniciou-se discussão, via Google Groups sobre como serão apresentados os comentários pós-PXP no PDJ.

Dia 32

1. Iniciou-se a produção do terceiro DEJ usando a técnica de PXP, com foco, agora, no cálculo de aleatoriedade dos temas e elementos de cenário;
2. Definiu-se o modo de como serão apresentados os comentários pós-PXP no

DPJ;

3. Começou-se a definir o modo de como serão apresentadas as soluções na Lista de PXP;
4. Adicionou-se as resoluções parciais, nos comentários pós-PXP do DPJ, para os pontos 2. A Corrida, Cálculo de Progressão e 5. Mudança de Cenários, segundo resultados do DPJ 2, Cálculo de Velocidade de Progressão Base.

Dia 33

1. Continuou-se a produção do terceiro DEJ usando a técnica de PXP, mantendo-se o foco no cálculo de aleatoriedade dos temas e elementos de cenário;
2. Mantém-se ainda em processo de definição o modo de como serão apresentadas as soluções, parciais e final, na Lista de PXP.

Dia 34

1. Continuou-se a produção do terceiro DEJ usando a técnica de PXP, mantendo-se o foco no cálculo de aleatoriedade dos temas e elementos de cenário. Em tal etapa, concluiu-se um primeiro balanceamento da aleatoriedade dos elementos do cenário Vila Mágica, restando ainda três temas a terem seus respectivos elementos de cenário equilibrados;
2. Colocou-se latente, por hora, o processo de definição sobre o modo como serão apresentadas as soluções, parciais e final, na Lista de PXP.

Dia 35

1. Continuou-se a produção do terceiro DEJ usando a técnica de PXP. Em tal etapa implementou-se um cálculo base para mudança de temas, ainda a ser observado em funcionamento e, então, lapidado;
2. Houve versionamento do DPJ, dividindo-se esse em versões pré e pós-PXP, chamando-as DPJ - versão 1 e DPJ - versão 1, respectivamente;
3. Foram adicionadas as resoluções parciais pós-PXP a suas respectivas perguntas na Lista de PXP.

Dia 36

1. Continuou-se a produção do terceiro DEJ usando a técnica de PXP. Na presente etapa, foi lapidado o cálculo base para mudança de temas, bem como implementou-se os elementos do cenário Nuvens Mágicas, Dia, tendo esse sua aleatoriedade já balanceada.

Dia 37

1. Continuada a produção do terceiro DEJ usando a técnica de PXP, confeccionou-se e iniciou-se a implementação dos assets do cenário Floresta Mágica.

Dia 38

1. Continuou-se a produção do terceiro DEJ usando a técnica de PXP. Na presente etapa, concluiu-se a implementação dos assets de todos os cenários, bem como lapidou-se, uma vez mais, o cálculo base para mudança de temas. Como consequência, os cálculos de aleatoriedade de instanciação e posicionamento de elementos de cenário, cálculo de progressão e movimento de personagem estão em processo de evolução, para melhor balanceamento de jogabilidade.

Dia 39

1. Continuou-se a produção do terceiro DEJ usando a técnica de PXP. Em tal etapa, finalizou-se o polimento dos assets de cenário, restando agora apenas o polimento do cálculo em foco de teste, bem como do cálculo de progressão.

Dia 40

1. Concluiu-se e compartilhou-se o terceiro DEJ usando a técnica de PXP;
2. Foram adicionados, ao DPJ - versão 2, os comentários de PXP contendo as resoluções parciais decorrentes do DEJ 3;
3. Foram adicionadas as resoluções parciais às soluções práticas da Lista de PXP;
4. Foi colocada em discussão uma ideia de mudança de história, e consequentes modificações de jogabilidade, resultante da observação de funcionamento do terceiro DEJ usando a técnica de PXP.

Dia 41

1. Iniciou-se a produção do quarto DEJ usando a técnica de PXP com foco, dessa vez, na busca do cálculo base de aleatoriedade de obstáculos;
2. Iniciou-se ainda a revisão dos obstáculos in-game, ponderando-se possíveis modificações ou adições, a fim de tornar dinâmica e mecânica de jogo mais divertidas;
3. Permaneceu em discussão a ideia de mudança de história e jogabilidade.

Dia 42

1. Iniciou-se a produção das artes do objetos-obstáculo Bomba Mágica e Caixa de Armadilha Mágica, a serem implementados no quarto DEJ usando a técnica de PXP.

Dia 43

1. Foi aprovada a ideia de mudança de história e jogabilidade;
2. Foi colocada latente a produção do DEJ usando a técnica de PXP, a fim de revisão dos elementos do DPJ e suas possíveis adaptações;
3. Foi adicionada a resolução parcial do item 6. História da Lista de PXP;
4. Foi adicionado um comentário e PXP no DPJ - versão 2, com a resolução parcial acerca da mudança de história.

Dia 44

1. Em decorrência da aprovada mudança de história e consequente modificação de aspectos do jogabilidade, foram revisados e complementados com comentários de PXP os itens 6. Conjunto de Características - História; 6.4 Conceito de Jogo; 1. Menu Principal, 2. Jogar e 3. Loja de Utensílios Mágicos do tópico 6.6 Resumo de Fluxo do Jogo; 6.7 Olhar e Sentir; e 6.8.3 Número de NPCs. Tais modificações, em sua maioria, foram referentes apenas a detalhes quanto à mudança de protagonista.

Dia 45

1. Continuou-se a revisão e complemento do conteúdo do DPJ - versão 2, segundo modificações de história e jogabilidade, com os respectivos comentários de PXP

abrangendo-se, nessa etapa, os tópicos 1. Coletáveis, 2. Power-ups e 3. Objetos-obstáculo do item 6.8.4 Número e Tipos de Objetos; e os tópicos 1. Achievements, Conquistas e 2. Missões do item 7.1 Objetivos.

Dia 46

1. Concluiu-se a revisão e complemento do conteúdo textual do DPJ - versão 2 com os respectivos comentários de PXP abrangendo-se, nessa etapa, todos os tópicos restantes, faltando assim apenas os rascunhos de planejamento de level design a serem atualizados;
2. Retificou-se alguns erros gramaticais nos documentos DPJ - versões 1 e 2, incluindo-se ainda, em ambos, pequena frase complementar no item 7.10 Rejogando e Salvando o Jogo;
3. Modificou-se, após reunião, o foco do quarto DEJ usando a técnica de PXP, tornando esse à aplicação da nova mecânica de jogabilidade decorrente da mudança de roteiro, reiniciando-se sua produção e passando-se, assim, ao próximo o cálculo base de aleatoriedade de objetos-obstáculo;
4. Foram adicionados, ao quarto DEJ usando a técnica de PXP, os sete assets de personagens referentes aos gatos da Brigada Arco-íris a serem controlados pelo jogador.

Dia 47

1. Foi implementada, no quarto protótipo DEJ usando a técnica de PXP, a mecânica de jogabilidade da Brigada Arco-íris, restando apenas breve polimento antes do compartilhamento;
2. Foi adicionada ao DPJ - versão 2 uma nova mecânica, ideia decorrente da primeira modificação de jogabilidade, que coloca como maior objetivo o salvamento do chef Annes, através da completude das metas já existentes.

Dia 48

1. Concluiu-se e compartilhou-se o quarto DEJ usando a técnica de PXP, cujo foco se dá na aplicação de mudança de jogabilidade apelidada de Mecânica da Brigada Arco-íris;

2. Adicionou-se o link do quarto DEJ usando a técnica de PXP a seus respectivos tópicos no DPJ - versão 2.

Dia 49

1. Adicionou-se ao DPJ - versão 2 os novos rascunhos de planejamento de level design, como consequência da mudança de jogabilidade pós-PXP. Nesse ponto, todas as modificações decorrentes de tal mudança foram aplicadas ao documento.

Dia 50

1. Foi iniciada a produção do quinto DEJ usando a técnica de PXP, dessa vez retomando-se o foco sobre o cálculo base para a aleatoriedade dos objetos-obstáculo. Nessa etapa, foram produzidos e adicionados os assets que comporão os obstáculos;
2. Poliu-se a descrição do objeto-obstáculo 3.4 Vassoura Mágica, do tópico 3. Objetos-obstáculo do DPJ - versão 2.

Dia 51

1. Continuou-se a produção do quinto DEJ usando a técnica de PXP. Nessa etapa, foram polidos os assets que comporão os objetos-obstáculo.

Dia 52

1. Continuou-se a produção do quinto DEJ usando a técnica de PXP. Em tal etapa, iniciou-se a construção e implementação da lógica de funcionamento, bem como do cálculo base de aleatoriedade dos objetos-obstáculo, desenvolvendo-se, ainda, o comportamento específico do objeto Bomba Mágica.

Dia 53

1. Continuou-se a produção do quinto DEJ usando a técnica de PXP. Nessa etapa, foi iniciada a construção do comportamento do objeto Balde de Água Fria, e continuado o desenvolvimento do cálculo base de aleatoriedade de objetos-obstáculo.

Dia 54

1. Continuou-se a produção do quinto DEJ usando a técnica de PXP. Em tal etapa, concluiu-se a implementação do comportamento do objeto-obstáculo Balde de Água Fria, e iniciou-se o mesmo processo para a Vassoura Mágica. Com tais implementos, vem desenvolvendo-se naturalmente parte da lógica do cálculo base de aleatoriedade de objetos-obstáculo.

Dia 55

1. Continuou-se a produção do quinto DEJ usando a técnica de PXP. Nessa etapa, concluiu-se a implementação do comportamento do objeto-obstáculo Vassoura Mágica, desenvolvendo-se, naturalmente e por conseguinte, a lógica do cálculo base de aleatoriedade de objetos-obstáculo;
2. Corrigiu-se um bug de instanciamento dos elementos de cenário, recorrente nos momentos após o tema Floresta Mágica.

Dia 56

1. Continuou-se a produção do quinto DEJ usando a técnica de PXP. Na presente etapa, poliu-se o comportamento do objeto-obstáculo Vassoura Mágica;
2. Adicionou-se um item de verificação - que se fez necessário em decorrência dos novos comportamentos e reações ingame - na cena do DEJ.

Dia 57

1. Continuou-se a produção do quinto DEJ usando a técnica de PXP. Na presente etapa, adicionou-se o comportamento do objeto-obstáculo Caixa de Armadilha Mágica, restando agora apenas polimentos de comportamento dos objetos-obstáculo, e de seus cálculos de aleatoriedade de efeitos internos ou instâncias gerais.

Dia 58

1. Continuou-se a produção do quinto DEJ usando a técnica de PXP. Nessa etapa, poliu-se o comportamento dos objetos-obstáculo Vassoura Mágica e Caixa de

Armadilha Mágica, principalmente quanto ao cálculo de perda de Moedas Mágicas. Implementados e observados os objetos-obstáculo e seus comportamentos, resta agora a lapidação do foco do protótipo em questão: o cálculo de aleatoriedade dos objetos-obstáculo.

Dia 59

1. Continuou-se a produção do quinto DEJ usando a técnica de PXP. Em tal etapa, concluiu-se a lógica e o cálculo de aleatoriedade dos objetos-obstáculo, alcançando-se, dessa forma, o objetivo do DEJ. Entretanto, com os comportamentos dos obstáculos e elementos de cenário interagindo entre si a olhos vistos, notou-se dificuldades de progressão de jogo por conta de dois elementos importantes: a mecânica de pulo e o design de cenário. Tais elementos serão lapidados numa segunda versão do presente DEJ.

Dia 60

1. Foi compartilhada a primeira versão do quinto DEJ usando a técnica de PXP, a fim de verificação. Em vista das dimensões alcançadas pelo projeto, disponibilizou-se o formato compactado .rar. O mesmo foi feito às demais versões, a fim de padronização dos arquivos;
2. Atualizaram-se os links dos DEJs no DPJ - versão 2;
3. Foi adicionada a resolução parcial do item 3. Aleatoriedade dos Obstáculos à Lista de PXP;
4. Foi adicionado um comentário de PXP no DPJ - versão 2, com a resolução parcial sobre o cálculo de aleatoriedade de objetos-obstáculo;
5. Iniciou-se a produção da segunda versão do quinto DEJ usando a técnica de PXP. Em tal etapa, poliu-se o cálculo e a mecânica de pulo.

Dia 61

1. Continuou-se a produção da segunda versão do quinto DEJ usando a técnica de PXP. Nessa etapa, poliu-se uma vez mais o cálculo e a mecânica de pulo e movimentação, bem como o design de alguns elementos de cenário e os valores do cálculo de aleatoriedade de objetos-obstáculo;
2. Padronizaram-se todos os DEJs compartilhados em formato .zip.

Dia 62

1. Concluiu-se e compartilhou-se via Google Drive a segunda versão do quinto DEJ usando a técnica de PXP.

Dia 63

1. Iniciou-se a produção do sexto DEJ usando a técnica de PXP, foco no cálculo base de aleatoriedade de power-ups. Em tal etapa, implementou-se o comportamento do NPC Xilef, bem como a lógica base de aleatoriedade de instanciação de power-ups;
2. Foram produzidas as artes dos objetos de power-up.

Dia 64

1. Continuou-se a produção do sexto DEJ usando a técnica de PXP. Em tal etapa, lapidou-se e implementou-se as artes dos objetos de power-up, bem como a base de seus cálculos de delay e aleatoriedade, junto ainda aos comportamentos específicos dos efeitos dos objetos Sandálias Mágicas, Ovo Mágico e Poção Mágica.

Dia 65

1. Continuou-se a produção do sexto DEJ usando a técnica de PXP. Nessa etapa, poliu-se a mecânica de posição de instanciação do NPC Xilef, bem como o comportamento de efeito do power-up Ovo Mágico e a arte do objeto de power-up Novelo Mágico, implementando-se ainda seu respectivo comportamento, efeito específico e consequente mecânica. Implementou-se, por fim, parte considerável dos comportamentos, mecânicas e efeitos do power-up Moto do Chef.

Dia 66

1. Concluiu-se e compartilhou-se via Google Drive o sexto DEJ usando a técnica de PXP;
2. Foi adicionada a resolução parcial do item 4. Aleatoriedade dos Power-ups à Lista de PXP;

3. Foram adicionados ao DPJ - versão 2 os comentários de PXP contendo as resoluções parciais decorrentes do Protótipo 6.

Dia 67

1. Poliu-se o texto da resolução parcial do item 4. Aleatoriedade dos Power-ups da Lista de PXP;
2. Retificou-se o número dos DEJs indicados nos comentários pós-PXP dos itens 3.1 Bombas Mágicas e 3.2.3 Explosão Mágica no tópico 3. Objetos-obstáculo;
3. Criou-se o documento Anexo, PXP e a Influência na Fábula de Jogo, com fins a resumir eventuais processos de influência da Programação Exploratória no conceito inicial de RunnerChief;
4. Iniciou-se a produção do sétimo DEJ usando a técnica de PXP, foco na aplicação dos cálculos de atração gravitacional para os centros de gravidade do cenário.

Dia 68

1. Marcou-se em destaque, no Diário de Desenvolvimento, as notações concernentes à influência da Programação Exploratória na Fábula de Jogo e no projeto de jogo;
2. Colocou-se no topo da lista, de forma permanente, os tópicos relacionados aos processos de mudança narrativa e de mecânica de jogo por influência de PXP no grupo RunnerChief do Google Groups;
3. Marcou-se em destaque, na Lista de PXP, os títulos dos tópicos contendo resoluções relacionadas às mudanças por influência da Programação Exploratória;
4. Concluiu-se, por hora, no arquivo Anexo, PXP e a Influência na Fábula de Jogo, a documentação dos processos de influência de PXP nos conceitos iniciais de RunnerChief;
5. Dividiu-se a produção do sétimo DEJ usando a técnica de PXP em duas versões: a primeira, testando de forma isolada a aplicação dos cálculos de atração gravitacional; a segunda, inserindo esses cálculos já funcionais à mecânica, contexto e dinâmica de RunnerChief. Assim, iniciou-se a produção do sétimo DEJ - versãp 1.

Dia 69

1. Concluiu-se a produção e compartilhou-se via Google Drive o sétimo DEJ - versão 1 usando a técnica de PXP;
2. Iniciou-se a produção do sétimo DEJ - versão 2 usando a técnica de PXP, com foco na aplicação dos cálculos já funcionais de gravitação do protótipo anterior à dinâmica de RunnerChief.

Dia 70

1. Concluiu-se a produção e compartilhou-se via Google Drive o sétimo DEJ - versão 1 usando a técnica de PXP;
2. Adicionou-se a resolução parcial decorrente do presente DEJ ao item 1.2 Centros de Gravidade da Lista de PXP;
3. Iniciou-se a adição dos comentários de PXP concernentes ao presente protótipo no DPJ - versão 2, contemplando-se, nessa etapa, o item 5. Cenários do tópico 7.5.1 A Mecânica Explícita, e adicionando-se ainda um novo item, 6. Gravidade, ao mesmo tópico.

Dia 71

1. Concluiu-se a adição dos comentários de PXP decorrentes do sétimo DEJ - versão 1 e 2, contemplando-se, nessa etapa, a construção do novo item 6. Gravidade, Conjunto de Influências e o preenchimento do item 4. Power-ups, Aleatoriedade, sem o devido preenchimento até então;
2. Iniciou-se a produção do oitavo DEJ usando a técnica de PXP, foco no encontro da lógica e aplicação da movimentação de parkour do protagonista. Ante sua primeira etapa de produção, notou-se importante um considerável polimento do design de cenário, já iniciado.

Dia 72

1. Continuou-se a produção das artes para a implementação e reformulação de cenário no oitavo DEJ usando a técnica de PXP. Nessa etapa, encontram-se concluídos os elementos de cenário dos temas Vila Mágica e Floresta Mágica.

Dia 73

1. Continuou-se a produção das artes para a implementação e reformulação de cenário no oitavo DEJ usando a técnica de PXP. Nessa etapa, concluiu-se a confecção dos elementos de cenário do tema Nuvens Mágicas, Dia.

Dia 74

1. Concluiu-se a produção das artes para a reformulação de cenário no oitavo DEJ usando a técnica de PXP, encontrando-se seu processo de implementação em vias de término.

Dia 75

1. Concluiu-se a implementação das artes para reformulação de cenário no oitavo DEJ;

2. Balanceou-se o funcionamento dos elementos “Instanciador de Xilef”, “Instanciador de Bomba Mágica”, “Posição de Respawn do Gato Mágico Vermelho” e, “Movimentação de Gato Mágico”;

3. Iniciou-se a implementação do sistema para realização dos movimentos de parkour.

Dia 76

1. Concluiu-se a produção oitavo DEJ usando a técnica de PXP, compartilhando-se o projeto via Google Drive;

2. Adicionaram-se a resolução parcial do item 1.1 Gravidade à Lista de PXP;

3. Adicionaram-se itens 7. Movimentação Extra, Parkour e 7. Parkour, Lógica de Deslocamento aos tópicos 7.5.1 A Mecânica Explícita e 7.5.2 A Mecânica Implícita, respectivamente, do DPJ - versão 2, contendo os comentários de PXP decorrentes do oitavo DEJ;

4. Iniciou-se a produção do nono DEJ usando a técnica de PXP, foco na implementação de animações para complemento do sistema de movimentação por parkour. Nessa etapa, começou-se a elaboração das artes para animação do protagonista Gato Vermelho.

Dia 77

1. Continuou-se a produção do nono DEJ usando a técnica de PXP. Nessa etapa, houve continuação da elaboração das artes para animação do protagonista Gato Vermelho;
2. Corrigiu-se o texto no item 4 do dia 76 do Diário de Desenvolvimento, a fim de maior clareza quanto à documentação da atividade desenvolvida.

Dia 78

1. Continuou-se a produção do nono DEJ usando a técnica de PXP. Em tal etapa, houve continuação de confecção da principal animação do protagonista Gato Vermelho, que encontra-se já em vias de finalização.

Dia 79

1. Continuou-se a produção do nono DEJ usando a técnica de PXP. Na presente etapa, concluiu-se os ciclos de animação de Corrida e Pulo do protagonista Gato Vermelho, restando por hora apenas o último ciclo de Cambalhota. Tais ciclos, após completos, serão reaproveitados e aplicar-se-ão, dessa forma, aos demais protagonistas.

Dia 80

1. Continuou-se a produção do nono DEJ usando a técnica de PXP. Nessa etapa, concluiu-se o último ciclo de animação do personagem Gato Vermelho. Com os ciclos prontos, passa-se à etapa próxima, de implementação das animações para complemento da movimentação de parkour. Bem-sucedida essa, os mesmos ciclos serão aplicados aos demais protagonistas.

Dia 81

1. Continuou-se a produção do nono DEJ usando a técnica de PXP. Nessa etapa, iniciou-se o processo de implementação das animações para complemento da movimentação de parkour. A esse ponto, faceando-se o funcionamento prático do sistema de animações do motor de jogo, concluiu-se se fazerem necessárias ainda uma quarta animação e duas transições, não essenciais para o foco do DEJ, mas com certeza bastante interessantes a seu polimento. Ainda assim, nota-se nesse

dia que, findando o prazo de produção de RunnerChief, tais transições não serão implementadas. De toda forma, sua ausência não compromete a investigação e resultados essenciais do processo de PXP.

Dia 82

1. Concluiu-se e compartilhou-se via Google Drive o nono DEJ usando a técnica de PXP;
2. Lançou-se importante questão acerca da natureza do DPJ a fim de tornar-se possível sua confecção;
3. Revisou-se e concluiu-se os documentos Anexo, PXP e a Influência na Fábula de Jogo; Diário de Desenvolvimento; Documento de Brainstorm; DPJ - versão 1; DPJ - versão 2; Lista de PXP; e OCCRDD.

Dia 83

1. Foi solucionada a questão acerca do DPJ. Observando-se que o DPJ - versão 2 acabou por cumprir, naturalmente ao longo do processo de PXP, as características desse modelo, apenas adequou-se algumas de suas informações, finalizando-se de fato o documento.

APÊNDICE C – Documentação do Projeto RunnerChief

C.1 Documento de Projeto de Jogo

O Projeto RunnerChief

1. Histórico do projeto

O histórico do projeto encontra-se nos documentos seguintes:

1. Diário de Desenvolvimento;
2. Documento de Brainstorm.

2. Integrantes do projeto

David de Oliveira Lemes

Vitor Ferreira Rodrigues

3. Documentos, pastas, comunicação

3.1 Arquivos do projeto: <<http://bit.ly/1L0VZTw>>

3.2 Grupo de discussão: <<http://bit.ly/1JfVzVr>>

4. Cronograma e ciclos de desenvolvimento

Segundo acordado entre os participantes, o fluxo de desenvolvimento se dará da seguinte maneira:

- 4.1. Lançamento da ideia inicial;
- 4.2. Desenvolvimento e ampliação da ideia inicial através de comunicação via Google Groups e documentação no chamado Documento de Brainstorm;
- 4.3. Confecção do DPJ [Documento de Projeto de Jogo], com base nos itens anterior-

mente desenvolvidos. Foram desenvolvidas duas versões do DPJ;

4.4. Desenvolvimento do DEJ [Documento Executável de Jogo] - para teste da real capacidade de funcionamento de itens-chave do DPJ, por meio da técnica de PXP [Programação Exploratória]. Foram desenvolvidos 9 DEJs;

4.5. Confeção, em paralelo, do Diário de Desenvolvimento, contendo breve documentação do processo de trabalho.

5. Resumo do Projeto

O projeto, chamado de RunnerChief, tem como principal objetivo o auxílio na produção da pesquisa de Doutorado na qual aborda-se o tema PXP [Programação Exploratória], como processo de validação de conceitos e ideias de um jogo por meio de testes concretos em DEJs pontuais, de forma a confirmar ou solicitar alinhamento de tais ideias à capacidade tecnológica e de produção.

RunnerChief, portanto, a princípio, busca colocar em prática as técnicas que vêm sendo pesquisadas, a fim de comprová-las ou melhor direcioná-las.

Em um segundo momento, dependendo do desenvolvimento de RunnerChief e de algumas circunstâncias, há a possibilidade de continuar a produção, com vistas a tornar RunnerChief um jogo comercial, numa vertente de produção de jogos mobile autorais.

6. Resumo do Jogo

6.1 Nome do Jogo

RunnerChief

6.2 Gênero

Endless Runner / Aventura / Plataforma.

6.3 Conjunto de características:

Plataformas: Android / IOS / Windows Phone 8;

Motor de Jogo: Unity Game Engine, versão 4.5.x;

Ambiente de Desenvolvimento: Windows 8, 4GB de RAM, Processador Intel Core 2 Quad, 1TB de HD, Placa de Vídeo GForce 9800 GT;

Linguagem de Programação: C#, dentro do framework Monodevelop, nativo do motor de jogo utilizado;

Estilo Gráfico: 2D (Estilo gráfico 8 bits);

Modos de Jogo: Campanha;

Física: Física 2D;

Número de Níveis: Não existem níveis, apenas progressão contínua da velocidade e desafio do jogo.

História: A história pode ainda ser melhor desenvolvida? Ou por ora cumpre seu papel de apresentação do jogo? Por ora cumpre, mas pode ser alterada após começar o processo de PXP [Programação Exploratória].

Pré-PXP: Annes, Agrag e os Gatos Mágicos

Annes, um chef de cozinha mágico, vive acompanhado de seus incontáveis gatos, também mágicos. Certo dia, Xilef, o mais esperto e mais próximo gato de Annes, avisa-o que o terrível Agrag aprisionou seus muitos gatos em diversos lugares diferentes. Annes então deixa sua mágica cozinha e corre junto a Xilef, utilizando-se ambos de todos os seus recursos fantásticos para salvar os bichanos.

Pós-PXP: No decorrer da produção dos protótipos de PXP, notou-se que a história mantinha-se viável de produção efetiva e ainda cumpria seu papel de apresentação. Entretanto, observações acerca do comportamento concreto da dinâmica de gameplay, fizeram notar-se o seguinte padrão:

O jogo apresentou uma natural facilidade de queda do personagem nos vãos do cenário, decorrendo daí grande quantidade de game overs. Tal questão poderia ser resolvida através de mais profundo balanceamento de gameplay, sobre pontos como movimentação de personagem, ou posicionamento e design dos elementos de cena, por exemplo. Ainda assim, esse mesmo padrão encontrou-se passível de ser utilizado a favor do game design, requisitando, porém, simples, mas importante, mudança de roteiro. Assim, a atual história encontra-se por:

Xilef e a Brigada Arco-íris

Annes, um chef de cozinha mágico, vive acompanhado de seus incontáveis gatos, também mágicos. Xilef, o mais esperto e mais próximo dos muitos gatos mágicos do cozinheiro

encantado, um dia descobre que o chef fora aprisionado pelo terrível Agrag, um feiticeiro que anseia tomar todas as suas receitas. Xilef reúne então uma brigada de sete gatos mágicos, que correrão juntos, utilizando-se de todos os seus recursos fantásticos para salvar o mestre-cuca.

Dada tal mudança, o jogador controlaria agora os sete gatos mágicos, um de cada vez. À “morte” de um, outro seria instantaneamente instanciado, dando continuidade ao gameplay. Tal se repetiria sete vezes, até o efetivo game over. O resultado então seria uma mecânica mais rica, diferente e divertida.

Origem: DEJ 3 - <<http://bit.ly/15u6CO1>>

Aplicação: DEJ 4 - <<http://bit.ly/1CgURHa>>

6.4 Conceito do jogo

O jogo tem como propósito inicial colocar em prática a pesquisa do processo de PXP [Programação Exploratória], concebendo-se assim uma ideia de jogo digital mobile, e testando-se alguns de seus elementos-chave através de protótipos digitais.

Em seguida, dependendo do andar do desenvolvimento, o jogo toma o propósito de ser um jogo mobile comercial autoral, onde pretende-se que o jogador viva uma mágica e dinâmica aventura na pele:

Pré PXP: do Chef Mágico Annes, na busca por salvar seus muitos gatos mágicos das garras do terrível Agrag.

Pós-PXP: de sete gatos mágicos, na busca por salvar o encantado chef Annes das garras do terrível Agrag.

Origem: DEJ 3 - <<http://bit.ly/15u6CO1>>

Aplicação: DEJ 4 - <<http://bit.ly/1CgURHa>>

O jogador, dessa forma, poderia se aventurar de forma simples e em qualquer lugar, visto o caráter mobile e simplista da proposta, aproveitando ainda a diversão do desafio crescente apresentado, buscando superar-se a cada partida, e da sólida e divertida produção artística 8 bits.

6.5 Público-alvo

O jogo dirige-se a um público de jovens e jovens adultos, entre 13 e 22 anos, de

ambos os sexos, que tenham acesso às amplamente disseminadas plataformas mobile e buscam aventurar-se de forma simples em mundos lúdicos mágicos durante alguns de seus momentos de lazer.

6.6 Resumo do fluxo do jogo

Para acessar fluxograma e planejamento de level design, vá ao item 7.3 Fluxo de Jogo

6.6.1 Menu Principal:

O jogador inicia o jogo ante à Tela Inicial, que contém, ao fundo:

Pré PXP: Annes cozinhando magicamente em sua cozinha fantástica e à frente, os botões Jogar, Loja de Utensílios Mágicos, Especialidades do Chef, Cardápio e Sair

Pós PXP: Annes, junto a oito bichanos, cozinhando magicamente em sua cozinha fantástica e à frente, os botões Jogar, Loja de Utensílios Mágicos, Especialidades do Chef, Cardápio e Sair.

Origem: DEJ 3 - <<http://bit.ly/15u6CO1>>

Desse ponto, o jogador fica livre para acessar quaisquer das opções, direcionando cada uma delas a uma parte do jogo. Assim, ao clicar em Jogar:

6.6.2 Jogar:

Cutscene:

Pré PXP: Annes ouve Xilef chamando-o apressadamente, e sai da cozinha correndo.

Pós PXP: Annes é apanhado por uma grande mão de fumaça mágica que invade a cozinha, e Xilef lidera os bichanos a resgatá-lo.

Origem: DEJ 3 - <<http://bit.ly/15u6CO1>>

Gameplay:

O jogador então é levado ao cenário da Vila Mágica, onde já pode interagir dentro da partida:

Pré PXP: no controle de Annes, e deve superar os desafios e alcançar metas, pessoais ou do jogo.

Pós PXP: no controle do primeiro gato mágico, e deve superar os desafios e alcançar metas, pessoais ou do jogo.

Origem: DEJ 3 - <<http://bit.ly/15u6CO1>>

Aplicação: DEJ 4 - <<http://bit.ly/1CgURHa>>

O jogador então progride na partida, e essa continua até que os estados de Pause ou Game Over sejam ativados.

Pause pode ser ativado por meio de um botão no canto da interface e, quando ativo, pára o jogo em meio à partida e exhibe o Menu de Pause, onde pode-se acessar as opções Continuar ou Sair.

Se escolher continuar, a partida continua do ponto onde parou.

Se escolher sair, é direcionado à Tela Inicial, onde tem acesso ao Menu Principal e seus recursos.

Game Over é ativado ao contato com alguma das condições de ativamento espalhadas pelo jogo, no cenário ou nos obstáculos. Quando ativo, a partida termina, e o menu “Jogar Novamente?” aparece. Nesse ponto, o interator pode escolher entre tentar de novo ou voltar ao Menu Principal.

Se tentar de novo, o processo recomeça, sem a cutscene inicial.

Se escolher voltar ao menu principal, é levado à tela inicial, de onde pode acessar novamente as diferentes partes do jogo.

6.6.3 Loja de Utensílios Mágicos:

É a loja ingame onde o jogador tem acesso às melhorias de power-ups disponíveis no jogo. É nela que, contemplados os valores requeridos, pode-se adquirir novos níveis dos power-ups utilizados:

Pré PXP: por Annes.

Pós PXP: pela Brigada Arco-íris.

Origem: DEJ 3 - <<http://bit.ly/15u6CO1>>

Após a aquisição ou não de melhorias, o jogador pode voltar à Tela Inicial, podendo

usufruir ou não delas segundo suas escolhas.

6.6.4 Especialidades do Chef:

É a tela de recordes e achievements. Nela pode-se visualizar e comparar os melhores resultados e todas as conquistas do jogador.

Após acesso, o jogador pode voltar à Tela Inicial para continuar o jogo ou não.

6.6.5 Cardápio:

É a tela de opções do jogo. Nela o jogador pode ajustar os volumes de trilha e efeitos sonoros, escolhendo mesmo se quer ouvi-los ou não.

6.6.6 Sair:

É a opção que desliga o jogo, voltando ao sistema operacional do aparelho do jogador.

6.7 Olhar e Sentir

Uma vez imerso no jogo, o jogador deverá viver e sentir o impulso:

Pré PXP: de Annes para salvar seus gatos, chegar mais longe e superar-se, buscando melhorar sempre, através do treino e da construção de estratégias com os diversos power-ups.

Pós PXP: da Brigada Arco-íris para salvar seu querido chef, chegar mais longe e superar-se, buscando melhorar sempre, através do treino e da construção de estratégias com os diversos power-ups.

Origem: DEJ 3 - <<http://bit.ly/15u6CO1>>

Cumprindo suas metas de evolução, o jogador deverá experimentar certa satisfação de atingi-las, bem como desfrutará os resultados de suas evoluções.

Falhando, poderá superar a frustração e tentar novamente, treinando, planejando e evoluindo até poder alcançar suas metas.

6.8 Escopo do Projeto

RunnerChief está no escopo do desenvolvimento da pesquisa de doutorado PXP - Programação Exploratória, realizada por David de Oliveira Lemes.

A produção de RunnerChief tem como primeiro intuito colocar em prática a teoria abordada, a fim de validá-la ou sugerir alinhamentos em alguns de seus pontos.

Completa sua primeira função, no futuro, há a possibilidade de mudança de escopo, adentrando-se aí um escopo voltado para o entretenimento comercial para plataformas mobile.

6.8.1 Número e tipos de cenários

Por ora, 4 cenários: Vila Mágica, Floresta Mágica, Nuvens Mágicas: Dia, Nuvens Mágicas: Noite.

6.8.2 Número e características dos níveis

Não há níveis em RunnerChief, apenas uma progressão contínua de dificuldade, acompanhada de aparição de cenários e itens de acordo com condições internas de jogo.

6.8.3 Número de NPCs

Pré PXP: Por ora, existem: 8 NPCs. A saber:

Xilef: O gato mágico mais esperto e mais próximo de Annes. O bichano de cor preta ajuda o protagonista em sua jornada, surgindo de vez em vez a flutuar e conceder power-ups diversos.

Gatos Mágicos: São eles os diversos gatos de Annes, aprisionados por Agrag. São vistos no jogo em 7 cores diferentes, em referência às cores do arco-íris, sendo elas: vermelho, laranja, amarelo, verde, azul, índigo e violeta.

Quando salvos, concedem a Annes determinadas quantidades de Moedas de Flipperama Mágicas.

Pós PXP: Por ora, existe 1 NPC. A saber:

Xilef: O gato mágico mais esperto e mais próximo de Annes. O bichano de cor preta ajuda e lidera a Brigada Arco-íris em sua jornada, surgindo de vez em vez a flutuar e conceder power-ups diversos.

Origem: DEJ 3 - <<http://bit.ly/15u6CO1>>

6.8.4 Número e tipos de objetos

Por ora, há 11 objetos no jogo, divididos esses em 3 categorias. São esses:

1. Coletáveis:

Pré PXP:

1.1 Moedas de fliperama Mágicas:

São moedas de fliperamas antigos, dotadas de mágicos poderes. Elas encontram-se espalhadas pelos diferentes cenários do jogo, e devem ser coletadas por Annes de modo a aumentar sua pontuação na partida.

Estão presentes, no jogo, nas seguintes formas e valores:

10 pontos, pequena; 25 pontos, média; 50 pontos, grande; 100 pontos, muito grande.

1.2 Gatos Mágicos:

São os diversos bichanos mágicos de Annes. Presentes em 7 diferentes cores, possuem igual e respectivamente, as seguintes 7 diferentes pontuações:

Vermelho: 50 pontos; Laranja: 75 pontos; Amarelo: 100 pontos; Verde: 125 pontos; Azul: 150 pontos; Índigo: 200 pontos; Violeta: 225 pontos.

Pós PXP:

1.1 Moedas de fliperama Mágicas:

São moedas de fliperamas antigos, dotadas de mágicos poderes. Elas encontram-se espalhadas pelos diferentes cenários do jogo, e devem ser coletadas pela Brigada Arco-íris de modo a aumentar sua pontuação na partida.

Estão presentes, no jogo, nas seguintes formas e valores:

10 pontos, pequena; 25 pontos, média; 50 pontos, grande; 100 pontos, muito grande.

1.2 Novelos Mágicos:

São mágicos artefatos de linhas de lã, deixados por Annes como uma trilha de resgate. Presentes em 7 diferentes cores, possuem igual e respectivamente, as seguintes 7 diferentes pontuações:

Vermelho: 50 pontos; Laranja: 75 pontos; Amarelo: 100 pontos; Verde: 125 pontos; Azul: 150 pontos; Índigo: 200 pontos; Violeta: 225 pontos.

Origem: DEJ 3 - <<http://bit.ly/15u6CO1>>

2. Power-ups:

Pré PXP:

2.1 Poções Energéticas:

Poções mágicas que concedem invulnerabilidade momentânea a Annes. Estão presentes nos seguintes níveis:

2.1.1 Poção Energética Azul: X segundos invulnerável.

2.1.2 Poção Energética Vermelha: X * 1,2 segundos invulnerável.

2.1.3 Poção Energética Rosa: X * 1,4 segundos invulnerável.

2.1.4 Poção energética Branca: X * 2 segundos invulnerável.

2.2 Ovo Mágico:

Misterioso ingrediente mágico que faz o jogo passar em câmera lenta. O efeito é suscetível de interrupção, caso Annes entre em choque com quaisquer dos obstáculos que implicam em Game Over Direto.

2.2.1 Ovo Pequeno: X segundos de câmera lenta.

2.2.2 Ovo Médio: X * 1,5 segundos de câmera lenta.

2.2.3 Ovo Grande: X * 2,5 segundos de câmera lenta.

2.3 Patins Mágicos:

Patins mágicos que fazem Annes flutuar. Assim, a movimentação se torna mais semelhante à de Jetpack Joyride.

2.3.1 Patins Mágicos “Dente-de-leite”: Fazem Annes flutuar.

2.3.2 Patins Mágicos Intermediários: Concedem mais agilidade na movimentação vertical de Annes.

2.3.3 Patins Mágicos “Fada-dos-dentes”: Concedem ainda mais agilidade na movimentação vertical de Annes.

2.4 Moto do Chef:

Moto de Annes. Mágica como todo o resto, tem grande resistência a impactos.

2.4.1 Moto do Chef v1: Imune a X bombas mágicas.

2.4.2 Moto do Chef v2: Imune a $X * 2$ bombas mágicas.

2.5 Nabo Mágico / Legume Mágico:

Misteriosa e rara leguminosa, capaz de conceder grande vitalidade a quem a ingere.

Pode ser adquirida e acumulada por Annes no decorrer de diversas partidas. Quando utilizada, revive o protagonista, concedendo uma nova chance de continuação da atual partida ao jogador.

Pós PXP:

2.1 Poções Energéticas:

Poções mágicas que concedem invulnerabilidade momentânea aos bichanos. Estão presentes nos seguintes níveis:

2.1.1 Poção Energética Azul: X segundos invulnerável.

2.1.2 Poção Energética Vermelha: $X * 1,2$ segundos invulnerável.

2.1.3 Poção Energética Rosa: $X * 1,4$ segundos invulnerável.

2.1.4 Poção energética Branca: $X * 2$ segundos invulnerável.

2.2 Ovo Mágico:

Misterioso ingrediente mágico que faz o jogo passar em câmera lenta. O efeito é suscetível de interrupção caso um bichano entre em choque com quaisquer dos obstáculos que implicam em Game Over Direto.

2.2.1 Ovo Pequeno: X segundos de câmera lenta.

2.2.2 Ovo Médio: $X * 1,5$ segundos de câmera lenta.

2.2.3 Ovo Grande: $X * 2,5$ segundos de câmera lenta.

2.3 Sandálias Mágicas:

Sandálias mágicas que fazem os gatos mágicos flutuarem. Assim, a movimentação se torna mais semelhante à de Jetpack Joyride.

DEJ 3:

2.3.1 Sandálias Mágicas “Dente-de-leite”: Fazem os gatos flutuarem.

2.3.2 Sandálias Mágicas Intermediárias: Concedem mais agilidade na movimentação vertical dos bichanos.

2.3.3 Sandálias Mágicas “Fada-dos-dentes”: Concedem ainda mais agilidade na movimentação vertical dos bichanos. **DEJ 6:**

2.3.1 Sandálias Mágicas “Dente-de-leite”: Concedem X segundos de flutuação.

2.3.2 Sandálias Mágicas Intermediárias: Concedem X * 2 segundos de flutuação.

2.3.3 Sandálias Mágicas “Fada-dos-dentes”: Concedem X * 3 segundos de flutuação.

Origem e Aplicação: DEJ 6 - <<http://bit.ly/1yImatk>> 2.4 Moto do Chef:

Moto de Annes. Mágica como todo o resto, tem grande resistência a impactos.

2.4.1 Moto do Chef v1: Imune a X bombas mágicas.

2.4.2 Moto do Chef v2: Imune a X * 2 bombas mágicas.

2.5 Novelo de Prata:

Misterioso e raro artefato de lã, capaz de conceder grande vitalidade a quem o toca.

Pode ser adquirido e acumulado pela Brigada Arco-íris no decorrer de diversas partidas. Quando utilizado, revive o protagonista, concedendo uma nova chance de continuação da atual partida ao jogador.

Origem: DEJ 3 - <<http://bit.ly/15u6CO1>>

Aplicação: DEJ 6 - <<http://bit.ly/1yImatk>>

3. Objetos-obstáculo:

Pré PXP:**3.1 Bombas Mágicas:**

Bombas preenchidas com magia explosiva. São arremessadas no decorrer do jogo de forma semelhante aos mísseis de Jetpack Joyride, com a diferença de reagirem à física, possuindo assim peso e podendo mesmo quicar no cenário.

Implicam em Game Over Direto.

3.2 Caixas de Armadilha Mágica:

Misteriosas caixas das quais não se pode prever ao certo o que sairá de dentro. Ao contato com Annes, pode gerar as seguintes reações-armadilha:

3.2.1 Perda de Moedas Mágicas.

3.2.2 Escapada de Gato Mágico.

3.2.3 Explosão Mágica. Implica em Game Over Direto.

3.2.4 Apenas um tropeço. Reduz de forma diminuta a velocidade de Annes.

Pós PXP:**3.1 Bombas Mágicas:**

Bombas preenchidas com magia explosiva. São arremessadas no decorrer do jogo de forma semelhante aos mísseis de Jetpack Joyride, com a diferença de reagirem à física, possuindo assim peso e podendo mesmo quicar no cenário.

DEJ 3: Implicam em Game Over Direto.

DEJ 5: Implicam em “morte” de Gato Mágico.

Origem: DEJ 5 - v01 - <<http://bit.ly/1CDw7XE>>

Aplicação: DEJ 5 v02 - <<http://bit.ly/15u6CO1>>

3.2 Caixas de Armadilha Mágica:

Misteriosas caixas das quais não se pode prever ao certo o que sairá de dentro. Ao contato com os gatos da Brigada, pode gerar as seguintes reações-armadilha:

3.2.1 Perda de Moedas Mágicas.

3.2.2 Sucção de energia mágica. Reduz velocidade e capacidade de pulo.

3.2.3 Explosão Mágica.

DEJ 3: Implica em Game Over Direto.

DEJ 5: Implica em “morte” de Gato Mágico.

Origem: DEJ 5 - v01 - <<http://bit.ly/1CDw7XE>>

Aplicação: DEJ 5 v02 - <<http://bit.ly/15u6CO1>>

3.2.4 Água Fria. Faz o bichano mágico pular assustado.

3.3 Balde de Água Fria:

Ponto fraco até mesmo dos bichanos mágicos, é um balde comum de água gelada suscetível de ser derrubado. Ao contato com um gato, fá-lo pular assustado de forma incontável.

3.4 Vassoura Mágica:

Vassoura encantada perigosamente ativa. Lançada por Agrag, pode cair dos céus como um meteoro. Ao contato com os gatos, acarreta perda de determinado montante aleatório de moedas mágicas.

Origem: DEJ 3 - <<http://bit.ly/15u6CO1>>

Aplicação 2: DEJ 5 - v01 - <<http://bit.ly/1CDw7XE>>

Aplicação 2: DEJ 5 v02 - <<http://bit.ly/15u6CO1>>

7. Jogabilidade e Mecânica

7.1 Objetivos

Do ponto de vista de game design, os objetivos devem ser revistos e testados em protótipos digitais.. Entretanto, parece-nos isso só será possível em estado avançado de produção de RunnerChief.

Vamos usar o PXP para validar isso depois. Por ora está ótimo. Depois do processo

de PXP podemos responder: o que de fato mudou?

Pós PXP:

O principal objetivo do jogo é salvar, em apenas uma partida, o maior número de gatos mágicos possível, acumulando dessa forma o maior número de pontos possível, batendo então o recorde anterior. Para tanto, o jogador controla Annes pela fase e seus diferentes mapas, numa corrida intérrima que aumenta de velocidade gradualmente, e deve se manter vivo pela maior distância e tempo possíveis, superando os obstáculos variados que se apresentam.

Como objetivos secundários, poder-se-iam colocar:

Pós PXP:

O principal objetivo inicialmente apresentado ao jogador é o de coletar, em apenas uma partida, o maior número de novelas e moedas de fliperama mágicas possível, acumulando dessa forma o maior número de pontos possível, batendo então o recorde anterior. Entretanto, como resultado natural do primeiro, surge um novo objetivo ainda maior e só revelado após completo pela primeira vez: salvar Annes, tantas vezes quanto possível. Para ambas as metas, o jogador controla a Brigada-Arco-íris, um gato de cada vez, pela fase e seus diferentes mapas, numa corrida intérrima que aumenta de velocidade gradualmente, e deve manter-se vivo pela maior distância e tempo possíveis, superando os obstáculos variados que se apresentam.

Origem: DEJ 3 - <<http://bit.ly/15u6CO1>>

Como objetivos secundários, poder-se-iam colocar:

1. Achievements, conquistas:

Pré PXP:

Desafios maiores, que levam maior tempo para serem completados. Quando terminados, desbloqueiam um novo troféu ao jogador.

São achievements:

Salvar X gatos em uma única partida;

Salvar X * 2 gatos em uma única partida;

Salvar X * 5 gatos em uma única partida;

Coletar um total X de moedas;

Coletar um total X * 2 de moedas;

Coletar um total X * 5 de moedas;

Salvar X gatos totais;

Salvar X * 2 gatos totais;

Salvar X * 5 gatos totais;

Resistir a X bombas totais com a Moto do Chef.

Resistir a X * 2 bombas totais com a Moto do Chef.

Completar todos os achievements anteriores.

Pós PXP:

Desafios maiores, que levam maior tempo para serem completados. Quando terminados, desbloqueiam um novo troféu ao jogador.

São achievements:

Coletar X Novelos Mágicos em uma única partida;

Coletar X * 2 Novelos Mágicos em uma única partida;

Coletar X * 5 Novelos Mágicos em uma única partida;

Coletar um total X de moedas;

Coletar um total X * 2 de moedas;

Coletar um total X * 5 de moedas;

Coletar X Novelos Mágicos totais;

Coletar X * 2 Novelos Mágicos totais;

Coletar X * 5 Novelos Mágicos totais;

Acumular X Novelos de Prata;

Acumular X * 2 Novelos de Prata;

Acumular X * 3 Novelos de Prata;

Resistir a X bombas totais com a Moto do Chef.

Resistir a X * 2 bombas totais com a Moto do Chef.

Salvar Annes 1 vez;

Salvar Annes 4 vezes;

Salvar Annes 7 vezes;

Salvar Annes 7+ vezes;

Correr X metros totais com a Brigada Arco-íris;

Completar todos os achievements anteriores.

Origem: DEJ 3 - <<http://bit.ly/15u6CO1>>

2. Missões:

Pré PXP:

Desafios menores, que podem ser feitos em menor tempo. Quando completos, concedem ao jogador Moedas de Fliperama Mágicas.

São missões:

Não utilizar power-ups na partida;

Terminar a partida com pontuação entre X e Y pontos. Existem 5 variações dessa missão;

Não coletar moedas na partida;

Coletar X moedas usando o patins. Existem 7 variações dessa missão.

Coletar X moedas usando a moto. existem 7 variações dessa missão.

Pós PXP:

Desafios menores, que podem ser feitos em menor tempo. Quando completos, concedem ao jogador Moedas de Fliperama Mágicas.

São missões:

Não utilizar power-ups na partida;

Terminar a partida com pontuação entre X e Y pontos. Existem 5 variações dessa missão;

Não coletar moedas na partida;

Coletar X moedas usando sandálias. Existem 7 variações dessa missão;

Coletar X moedas usando a moto. Existem 7 variações dessa missão;

Correr X metros com o gato de cor Y. Existem 21 variações dessa missão.

Origem: DEJ 3 - <<http://bit.ly/15u6CO1>>

3. Upgrades:

Alcançar os níveis máximos de upgrade dos power-ups.

7.2 Jogabilidade

Toda a jogabilidade se dá através de toques na tela e leitura. Assim interage o jogador, desde a movimentação do personagem, até o reconhecimento de missões e navegação pelos menus.

7.2.1 Progressão do Jogo

A progressão do jogo é contínua, não havendo níveis ou fases.

Assim, o mais próximo que se tem de “fases” seria a mudança dos temas de cenário, que acarretam algumas mudanças no mapa e por tanto na disposição dos desafios.

7.2.2 Estrutura das Missões / Desafios*

*O item é suscetível de ampliação segundo o item 7.1 Objetivos.

Segundo os tópicos do item 7.1 Objetivos, segue a descrição das estruturas:

1. Desafio Principal:

Pré PXP:

Acumular, em uma única partida, o maior número de pontos possível, salvando o maior número possível de Gatos Mágicos e coletando a maior quantidade de moedas de fliperama mágicas possível. Para tanto, deve-se continuar correndo pela tela pelos maiores tempo e distância possíveis, superando os obstáculos diversos a se apresentarem, através do treino do próprio jogador, e da coleta e evolução dos power-ups.

Pós PXP:

O principal desafio apresentado, a princípio, ao jogador, é o de acumular, em uma única partida, o maior número de pontos possível, coletando o maior número de Novelos Mágicos e moedas de fliperama mágicas possível. Entretanto, o desafio maior, e só revelado quando alcançado pela primeira vez, é o de salvar Annes, tantas vezes quanto for possível. Para tanto, deve-se continuar correndo pela tela pelos maiores tempo e distância possíveis, superando os obstáculos diversos a se apresentarem, através do treino do próprio jogador, e da coleta e evolução dos power-ups.

Origem: DEJ 3 - <<http://bit.ly/15u6CO1>>

2. Achievements, conquistas:

Pré PXP:

2.1 Salvar X gatos em uma partida: Deve-se libertar um número X de gatos mágicos das jaulas enfeitadas de Agrag, espalhadas pela tela.

2.2 Salvar $X * 2$ gatos em uma partida: Deve-se libertar um número $X * 2$ de gatos mágicos das jaulas enfeitiçadas de Agrag, espalhadas pela tela.

2.3 Salvar $X * 5$ gatos em uma partida: Deve-se libertar um número $X * 5$ de gatos mágicos das jaulas enfeitiçadas de Agrag, espalhadas pela tela.

2.4 Coletar um total de X moedas: Deve-se coletar um número total X das moedas de fliperama mágicas espalhadas pelo mapa, no decorrer de diversas partidas jogadas.

2.5 Coletar um total de $X * 2$ moedas: Deve-se coletar um número total $X * 2$ das moedas de fliperama mágicas espalhadas pelo mapa, no decorrer de diversas partidas jogadas.

2.6 Coletar um total de $X * 5$ moedas: Deve-se coletar um número total $X * 5$ das moedas de fliperama mágicas espalhadas pelo mapa, no decorrer de diversas partidas jogadas.

2.7 Salvar X gatos totais: Deve-se libertar um número X total de gatos mágicos das jaulas enfeitiçadas de Agrag, espalhadas pelo mapa, através do decorrer de diversas partidas jogadas.

2.8 Salvar $X * 2$ gatos totais: Deve-se libertar um número $X * 2$ total de gatos mágicos das jaulas enfeitiçadas de Agrag, espalhadas pelo mapa, através do decorrer de diversas partidas jogadas.

2.9 Salvar $X * 5$ gatos totais: Deve-se libertar um número $X * 5$ total de gatos mágicos das jaulas enfeitiçadas de Agrag, espalhadas pelo mapa, através do decorrer de diversas partidas jogadas.

2.10 Resistir a X bombas totais com a Moto do Chef: Deve-se ser atingido por um número X total de Bombas Mágicas quando utilizando a Moto do Chef, no decorrer de diversas partidas jogadas.

2.11 Resistir a $X * 2$ bombas totais com a Moto do Chef: Deve-se ser atingido por um número $X * 2$ total de Bombas Mágicas quando utilizando a Moto do Chef, no decorrer de diversas partidas jogadas.

2.12 Gatos e Maçãs: Deve-se completar todos os achievements, a fim de salvar os gatinhos extras e desbloqueiar o Troféu-maçã-dourada.

Pós PXP:

2.1 Coletar X novos mágicos em uma partida: Deve-se coletar um número X de

novelos mágicos deixados por Annes, espalhados pela tela.

2.2 Coletar $X * 2$ novelos mágicos em uma partida: Deve-se coletar um número $X * 2$ de novelos mágicos deixados por Annes, espalhados pela tela.

2.3 Coletar $X * 5$ novelos mágicos em uma partida: Deve-se coletar um número $X * 5$ de novelos mágicos deixados por Annes, espalhados pela tela.

2.4 Coletar um total de X moedas: Deve-se coletar um número total X das moedas de fliperama mágicas espalhadas pelo mapa, no decorrer de diversas partidas jogadas.

2.5 Coletar um total de $X * 2$ moedas: Deve-se coletar um número total $X * 2$ das moedas de fliperama mágicas espalhadas pelo mapa, no decorrer de diversas partidas jogadas.

2.6 Coletar um total de $X * 5$ moedas: Deve-se coletar um número total $X * 5$ das moedas de fliperama mágicas espalhadas pelo mapa, no decorrer de diversas partidas jogadas.

2.7 Coletar X novelos totais: Deve-se coletar um número X total de novelos mágicos deixados por Annes, espalhados pelo mapa, através do decorrer de diversas partidas jogadas.

2.8 Coletar $X * 2$ novelos totais: Deve-se coletar um número $X * 2$ total de novelos mágicos deixados por Annes, espalhados pelo mapa, através do decorrer de diversas partidas jogadas.

2.9 Coletar $X * 5$ novelos totais: Deve-se coletar um número $X * 5$ total de novelos mágicos deixados por Annes, espalhados pelo mapa, através do decorrer de diversas partidas jogadas.

2.10 Acumular X novelos de prata: Deve-se acumular, ao decorrer de diversas partidas jogadas, um número máximo de X novelos de prata antes de utilizá-los.

2.11 Acumular $X * 2$ novelos de prata: Deve-se acumular, ao decorrer de diversas partidas jogadas, um número máximo de $X * 2$ novelos de prata antes de utilizá-los.

2.12 Acumular $X * 3$ novelos de prata: Deve-se acumular, ao decorrer de diversas partidas jogadas, um número máximo de $X * 3$ novelos de prata antes de utilizá-los.

2.13 Resistir a X bombas totais com a Moto do Chef: Deve-se ser atingido por um número X total de Bombas Mágicas quando utilizando a Moto do Chef, no decorrer de diversas partidas jogadas.

2.14 Resistir a $X * 2$ bombas totais com a Moto do Chef: Deve-se ser atingido por um número $X * 2$ total de Bombas Mágicas quando utilizando a Moto do Chef, no decorrer de diversas partidas jogadas.

2.15 Correr X metros totais com a Brigada Arco-íris: Deve-se correr, no decorrer das diversas partidas jogadas, um número X total de metros no controle dos diversos gatos da Brigada-arco-íris.

2.16 Novelo Dourado: Deve-se completar todos os achievements, a fim de salvar Annes uma vez mais e desbloqueiar o Troféu Novelo Dourado.

Origem: DEJ 3 - <<http://bit.ly/15u6CO1>>

3. Missões:

Pré PXP:

3.1 Não utilizar power-ups na partida: Não acionar ou fazer uso de quaisquer itens da categoria power-ups, concedidos pelo gato Xilef, durante a partida.

3.2 Terminar a partida com pontuação entre X e Y pontos: Deve-se terminar a partida, induzindo-se o estado de Game Over no momento em que a pontuação atingir um valor entre X e Y. - Essa missão contém 5 variáveis, modificando-se os valores de X e Y em cada uma.

3.3 Não coletar moedas na partida: Não coletar nenhuma das moedas de fliperama mágicas da tela durante a partida.

3.4 Coletar X moedas usando o patins: Coletar determinado número X total de moedas de fliperama mágicas enquanto em uso do item de power-up patins. - Essa missão contém 7 variáveis, modificando-se o valor de X em cada uma.

3.5 Coletar X moedas usando a moto: Coletar determinado número X total de moedas de fliperama mágicas enquanto em uso do item de power-up moto do chef. - Essa missão contém 7 variáveis, modificando-se o valor de X em cada uma.

Pós PXP:

3.1 Não utilizar power-ups na partida: Não acionar ou fazer uso de quaisquer itens da categoria power-ups, concedidos pelo gato Xilef, durante a partida.

3.2 Terminar a partida com pontuação entre X e Y pontos: Deve-se terminar a partida

induzindo-se o estado de Game Over no momento em que a pontuação atingir um valor entre X e Y. - Essa missão contém 5 variáveis, modificando-se os valores de X e Y em cada uma.

3.3 Não coletar moedas na partida: Não coletar nenhuma das moedas de fliperama mágicas da tela durante a partida.

3.4 Coletar X moedas usando sandálias: Coletar determinado número X total de moedas de fliperama mágicas enquanto em uso do item de power-up sandálias mágicas. - Essa missão contém 7 variáveis, modificando-se o valor de X em cada uma.

3.5 Coletar X moedas usando a moto: Coletar determinado número X total de moedas de fliperama mágicas enquanto em uso do item de power-up moto do chef. - Essa missão contém 7 variáveis, modificando-se o valor de X em cada uma.

3.6 Correr X metros com o gato de cor Y: Correr um número X total de metros com determinado bichano da Brigada-Arco-íris em uma única partida. - Essa missão contém 21 variáveis, modificando-se X entre três valores diferentes, e Y entre os sete diferentes gatos mágicos.

Origem: DEJ 3 - <<http://bit.ly/15u6CO1>>

7.2.3 Estruturas dos Puzzles

O único puzzle presente em RunnerChief até o momento é o uso de itens. Ainda assim, tal categoria de puzzle é bastante simplificada pela forma de como tal uso pode ou não ser feito. Se não vejamos:

O acesso aos itens em meio à partida só se dá através do gato Xilef, que, vez ou outra, segundo regras de aleatoriedade do jogo, surge flutuando pela tela e concede algum item surpresa ao jogador. O item só faz efeito quando o jogador o coleta.

A principal forma de escolha quanto aos itens, então, é pegá-los ou não quando lançados por Xilef, construindo-se assim uma certa estratégia de jogo nesse aspecto, como por exemplo utilizar a movimentação de um patins mágico, ou deixar de coletá-lo e manter a movimentação padrão.

Assim, o jogador não escolhe qual item terá em mãos, e a escolha de quando usar um item é também reduzida aos momentos de aparição de Xilef, e de coleta do jogador.

7.3 Fluxo do Jogo

Abaixo o fluxograma do fluxo lógico de RunnerChief, segundo especificações do item 6.6 Resumo do Fluxo de Jogo.

Na imagem:

Linhas significam “avanço”; pontilhados significam “volta”.

Quadros de contorno forte: “telas”; quadros de contorno fraco: “menus”.



Figura 5 – Fluxo do jogo RunnerChief

7.4 Planilha de fluxo de telas

1. Abaixo o fluxograma do fluxo de telas geral de RunnerChief. Intimamente ligado ao fluxo lógico, possui apenas pequenas mudanças.

Na imagem:

Linhas significam “avanço”; pontilhados significam “volta”.

Quadros de contorno forte: “telas”; quadros de contorno fraco: “menus”.

2. Ainda abaixo, os conceitos base de fluxo de Level Design, um para cada tema de



Figura 6 – Fluxo de telas do jogo RunnerChief

cenário.

Pré PXP:



Figura 7 – Pré PXP: Vila Mágica

Pós PXP:



Figura 8 – Pré PXP: Floresta Mágica



Figura 9 – Pré PXP: Nuvens Mágicas, Dia

Origem: DEJ 3 - <http://bit.ly/15u6CO1>

Aplicação Parcial: DEJ 4 - <http://bit.ly/1CgURHa>

7.5 Mecânicas

O jogo consiste na acumulação de pontos através da superação de obstáculos e coleta de itens, numa corrida progressiva por uma tela em estilo plataforma 2D.

Assim, o jogador deve utilizar-se de seus reflexos visomotores, junto a planejamentos simples e rápidos, para seguir vivo e adiante através dos desafios e velocidade crescentes do jogo.



Figura 10 – Pré PXP: Nuvens Mágicas, Noite



Figura 11 – Pós PXP: Vila Mágica

A acumulação de pontos indica a autossuperação do jogador, enquanto a de moedas mágicas proporciona a aquisição de melhorias nos itens power-ups que podem ser utilizados na partida.

7.5.1 A mecânica explícita

1. Pontuação:

A pontuação se dá através:

Pré PXP: do salvamento dos gatos mágicos, da coleta de moedas de fliperama mágicas e da distância percorrida. Seu estado atual, bem como suas atualizações, ficam constan-



Figura 12 – Pós PXP: Floresta Mágica



Figura 13 – Pós PXP: Nuvens Mágicas, Dia

temente visíveis ao jogador, através de uma HUD de pontuação na interface do jogo.

Pós PXP: da coleta de novelos mágicos, da coleta de moedas de fliperama mágicas e da distância percorrida. Seu estado atual, bem como suas atualizações ficam constantemente visíveis ao jogador, através de uma HUD de pontuação na interface do jogo.

Origem: DEJ 3 - <<http://bit.ly/15u6CO1>>

2. A Corrida:

A corrida é constante e de velocidade progressiva, sendo claramente sentida pelo jogador no decorrer da partida. Seu papel é aumentar o desafio do jogo, até o momento onde



Figura 14 – Pós PXP: Nuvens Mágicas, Noite

o jogador entrará em estado de Game Over, e deverá recomeçar se quiser bater seus recordes.

3. Obstáculos:

Os obstáculos têm como meta, junto à velocidade progressiva, desafiar os reflexos do jogador. Eles surgem em variadas formas, como em vãos entre telhados ou bombas mágicas, e podem tanto desorganizar o ritmo de corrida do jogador, como levá-lo ao estado de game over.

Os efeitos dos obstáculos são sempre acionados ao contato com o protagonista.

4. Power-ups:

Os power-ups, por outro lado, têm a função de tentar facilitar a jornada do protagonista, concedendo-lhe impulsos, proteções e ligeiras modificações de jogabilidade, a fim da melhor superação dos obstáculos e, por conseguinte, maior desenvolvimento na partida e no jogo.

Seus efeitos se dão ao contato com o protagonista, e sua utilização dentro da partida só se dá através da coleta, após sua concessão pelo NPC Xilef, o gato. A escolha, portanto, do item a ser utilizado, restringe-se apenas a coletá-lo ou não, pois sua disponibilidade é gerada por regras implícitas.

Os power-ups são ainda suscetíveis de melhorias, através da aquisição progressiva dessas na loja ingame. Quanto mais avançados, maior a tendência de facilitar o percurso do jogador.

5. Cenários:

A mudança dos temas de cenário é igualmente visível ao jogador.

Na mecânica, tais mudanças implicam apenas ligeiras diferenças na disposição das plataformas, requerendo determinada atenção do jogador:

Pré PXP: quanto aos pulos a serem realizados.

Pós PXP: quanto aos pulos a serem realizados, à exceção entretanto do tema Nuvens Mágicas, Noite, que dispõe dos chamados Planetóides, excededores de influência gravitacional sobre o protagonista, que incrementa a mecânica usual.

6. Gravidade:

Ocorre como força constante no jogo, influenciando em power-ups, obstáculos e os próprios personagens jogáveis. Está ainda presente como força secundária - mas igualmente importante - quando se faz presente o elemento de cenário Planetóide, do tema Nuvens Mágicas, Noite, que exerce poder de atração considerável sobre o protagonista, modificando sua dinâmica de movimentação pelo cenário.

Aplicação: DEJ 7 - v02 - <<http://bit.ly/1yP3OII>>

7. Movimentação Extra, Parkour:

Junto à movimentação padrão e pontuais modificações na dinâmica gravitacional do jogo, podem ainda os personagens jogáveis realizar pequena série de movimentos extras, permitindo-os a escalada de paredes, ganho de impulso ou completude de pulos demasiadamente longos entre plataformas, por exemplo. Essa classe de movimentos é chamada, em RunnerChief, movimentação de parkour.

Aplicação: DEJ 8 - <<http://bit.ly/1BIBs3f>>

8. Distância:

É incrementada ao passo que o jogador progride na partida, sendo tal incrementação dependente da velocidade do jogo e sua progressão. Quanto mais tempo na partida, maior a distância total percorrida.

A distância percorrida na partida é, como a pontuação, atualizada e vista em tempo real, através de uma HUD na interface do jogo.

9. Missões:

As missões, em RunnerChief, são metas em geral de mais simples, ou ainda, mais rápida completude. Ao serem completas, concedem ao jogador determinados montantes de moedas de fliperama mágicas.

As missões presentes no jogo estão descritas no item 3. Missões, do tópico 7.2.2 Estrutura das Missões / Desafios.

10. Achievements, Conquistas

São, em RunnerChief, as metas mais trabalhosas ou, de mais lenta completude. Quando completas, desbloqueiam diferentes troféus ao jogador.

Atenção devida à particularidade do achievement:

Pré PXP: Gatos e Maçãs, o único que, adicionalmente, libera e salva certa quantidade de gatos mágicos ao jogador quando completo.

Pós PXP: Novelo Dourado, o único que, adicionalmente, libera e salva Annes mais uma vez para o jogador quando completo.

Origem: DEJ 3 - <<http://bit.ly/15u6CO1>>

Os achievements, conquistas, estão detalhados no item 2. Achievements, conquistas, do tópico 7.2.2 Estrutura das Missões / Desafios.

11. Salvar e Rejogar:

O processo de save game se dá automaticamente, toda vez que alguma ação significativa é realizada no jogo. - Para informações complementares, vá ao item 7.5.2 A mecânica implícita.

O processo de rejogo, por sua vez, se dá manualmente, à escolha do jogador e através dos menus ingame. Pelo menu de partida, toda vez que uma partida termina; ou pelo menu principal, toda vez que o jogo é iniciado no aparelho.

7.5.2 A mecânica implícita

Pré PXP:

1. Pontuação, O Cálculo Interno:

O cálculo interno de pontuação é feito com base nas metas principais do jogo, que são:

1. Salvar o maior número de Gatos Mágicos possível;
2. Acumular o maior número de pontos, coletando as Moedas de Fliperama Mágicas, possível;
3. Alcançar a maior distância possível.

Dessa forma, cada Gato Mágico possui um valor de pontuação, bem como cada Moeda de Fliperama Mágica.

A fórmula então seria: $Pt = Gm + Pmfm + Dmp$

Onde:

“Pt” = Pontuação Total;

“Gm” = Gatos Mágicos (com seus diferentes valores);

“Pmfm” = Pontuação de Moedas de Fliperama Mágicas (também com seus diversos valores).

“Dmp” = Distância Máxima Percorrida.

Pós PXP:

1. Pontuação, O Cálculo Interno:

O cálculo interno de pontuação é feito com base nas metas principais do jogo, que são:

1. Coletar o maior número de Novelos Mágicos possível;
2. Acumular o maior número de pontos, coletando as Moedas de Fliperama Mágicas, possível;
3. Alcançar a maior distância possível.

Dessa forma, cada Novelo Mágico possui um valor de pontuação, bem como cada Moeda de Fliperama Mágica.

A fórmula então seria: $Pt = Nm + Pmfm + Dmp$

Onde:

“Pt” = Pontuação Total;

“Nm” = Novelos Mágicos (com seus diferentes valores);

“Pmfm” = Pontuação de Moedas de Fliperama Mágicas (também com seus diversos valores).

“Dmp” = Distância Máxima Percorrida.

Origem: DEJ 3 - <<http://bit.ly/15u6CO1>>

2. A Corrida, Cálculo de Progressão:

Compreende-se até o momento que, somente desta forma se poderá testar uma base para a fórmula de progressão de velocidade de jogo, ao menos num primeiro momento.

Pré PXP:

A progressão de velocidade da corrida de Annes, em RunnerChief, se dá pela fórmula X, não conhecendo-se ainda o valor de X.

Pós PXP:

DEJ 2: Após tal DEJ ser construído, estabeleceram-se determinados valores de velocidade e incremento, junto a uma condição de tempo, em segundos, para tal acréscimo.

Assim, as fórmulas e condições seriam:

Se $(T < T+5)$

$Vp = Vp + In$

Onde:

“T” = Tempo em segundos;

“Vp” = Velocidade de Progressão;

“In” = Incremento;

Origem e Aplicação: DEJ 2 - <<http://bit.ly/1Eu9Op3>>

DEJ 3: Com os elementos de jogo mais concretamente delineados através da prototipagem, pôde se observar de forma um tanto mais clara sua dinâmica. O incremento de velocidade de progressão tornou-se então menor, a fim de balanceamento de gameplay. Sua fórmula, entretanto, não se modificou.

Aplicação e Origem: DEJ 3 - <<http://bit.ly/15u6CO1>>**3. Obstáculos, Aleatoriedade:****Pré PXP:**

O cálculo de aleatoriedade dos obstáculos é X.

Os obstáculos e suas funcionalidades estão descritos no item 3. Objetos-obstáculo, do tópico 6.8.4 Número e Tipos de Objetos.

Pós PXP:

A construção dos DEJs tornou visível o comportamento e a interação dos obstáculos junto à dinâmica geral do jogo e, por conseguinte, demonstrou a necessidade de um “tempo seguro” ao jogador, onde não há obstáculos ativos. Determinados obstáculos revelaram-se também, mais difíceis do que outros, resultando assim na separação, por determinados percentuais, de suas possibilidades de ocorrência. Adicionou-se ainda um percentual Y de não-ocorrência de obstáculos. Por fim, colocou-se como requisito de instanciação um determinado tempo de delay, variável entre X e Y, concluindo o balanceando.

As fórmulas e condições então, seriam:

1. Se (sd == verdadeiro)

D = Randomizar(X, Y)

2. Se (C >= D)

Oa = Randomizar(X, Y)

Csd == verdadeiro

Onde:

“Csd” = Chave de Sorteio de Delay;

“D” = Delay;

“C” = Contador de Tempo;

“Oa” = Obstáculo Ativo;

Os obstáculos e suas funcionalidades estão descritos no item 3. Objetos-obstáculo, do tópico 6.8.4 Número e Tipos de Objetos.

Origem e Aplicação: DEJ 5 - <<http://bit.ly/1CDw7XE>>

4. Power-ups, Aleatoriedade:

O cálculo base de aleatoriedade dos power-ups deve ser testado no motor de jogo.

Pré PXP:

O cálculo de aleatoriedade dos power-ups concedidos pelo gato Xilef é X.

Assim como o item anterior, suas funcionalidades estão descritas no item 2. Power-ups, do tópico 6.8.4 Número e Tipos de Objetos.

Pós PXP:

Após prototipagem, tendo em vista ainda o modelo de mecânica planejada em que o NPC Xilef é quem concede os power-ups, dividiu-se o processo de cálculo de aleatoriedade em dois procedimentos menores. O primeiro, um cálculo de definição do tempo de delay para a aparição do NPC, terminante em menores valores antes da primeira aparição, e em maiores após essa; o segundo, um cálculo de randomicidade, dependente do surgimento de Xilef para seu efetivo funcionamento e resultante em determinado valor entre X e Y que corresponde, por sua vez, segundo certas diretrizes condicionais, a um dos power-ups possíveis.

Assim, os processos, fórmulas e condições seriam:

1. Se (Pv == verdadeiro)

D = Randomizar(A, B)

Se não

D = Randomizar(X, Y)

Se(C >= D)

NPC == verdadeiro

2. Se(NPC== verdadeiro)

Ns = Randomizar(X, Y)

Caso(Ns <25%)

Pw1 == verdadeiro

Caso(Ns >25% && Ns <= 50%)

Pw2 == verdadeiro

Caso(Ns >50% && Ns <= 75%)

Pw3 == verdadeiro

Caso(Ns >75% && Ns <= 90%)

Pw4 == verdadeiro

Caso(Ns >90% && <= 100%)

Pw5 == verdadeiro

Onde:

“Pv” = Primeira Vez;

“D” = Delay;

“A” e “B” = Valores inteiros mínimo e máximo, respectivamente.

“X” e “Y” = Valores inteiros mínimo e máximo, respectivamente.

“C” = Contador de tempo, em segundos.

“NPC” = Xilef.

“Ns” = Número Sorteado.

“PwX” = Power-up número X.

Assim como o item anterior, suas funcionalidades estão descritas no item 2. Power-ups, do tópico 6.8.4 Número e Tipos de Objetos.

Origem e Aplicação: DEJ 6 - <<http://bit.ly/1yImatk>>

5. Mudança de Cenários:

O tempo de mudança e aleatoriedade dos temas de cenário, bem como os diferentes posicionamentos dos telhados, devem ser testados no motor de jogo, a fim do encontro das fórmulas base para tal, ao menos num primeiro momento.

Pré PXP:

As fórmulas para os diferentes posicionamentos dos elementos de cenário são X, Y e Z.

A fórmula para o tempo de mudança de cenários é X.

A fórmula para a aleatoriedade dos cenários é X.

Pós PXP:

DEJ 2:

Posicionamento de Elementos de Cenário: A fórmula para o posicionamento dos elementos de cenário revelou-se, após tal DEJ, dependente de uma espécie de regra de tempo de delay entre a instanciação de um elemento e outro. Sua base, porém, ainda mesmo para definir-se o tempo de delay, encontrou-se bastante ligada à velocidade de progressão do jogo, estando ambos interrelacionados. Ainda assim, tal ligação, em termos de programação, não deu-se de forma direta, mas sim por intermédio de um terceiro elemento: o binômio condicional posicionamento dos elementos de cenário X número de elementos ativos.

Assim, as fórmulas e condições seriam:

Se (Pec <= X && Neca <4)

Nec == verdadeiro

Onde:

“Pec” = Posição de Elementos de Cenário;

“Neca” = Número de Elementos de Cenário Ativos;

“Nec” = Novo Elemento de Cenário;

“X” = Valor de posição no eixo X;

Estabelecendo-se, dessa forma, uma posição-chave para instanciação de novo elemento de cenário, quanto maior a velocidade de progressão do jogo, mais rápido alcançar-se-á tal posição, realizando-se assim o instanciamento, mantendo-se o tempo de delay e espaçamento entre elementos do cenário sempre constante e proporcional.

Origem e Aplicação: DEJ 2: <<http://bit.ly/1Eu9Op3>>

DEJ 3:

Posicionamento de Elementos de Cenário: Através de prototipagem um tanto mais completa, observou-se a presença de dinâmicas específicas para cada tema de cenário. Assim, sem modificação do cálculo base, adicionou-se apenas pequena fórmula condicional precedente, junto a duas mudanças de valor na fórmula principal, tornando assim o cálculo mais versátil e adaptável às requisições de cada tema.

As fórmulas então seriam:

1. Se(Csp == verdadeiro)

Pc = Randomizar(X, Y)

2. Se(Pec <= Pc && Neca <= A)

Nec == Verdadeiro

Onde:

“Csp” = Chave de Sorteio de Posição;

“Pc” = Posição Chave;

“X” e “Y” = Valor flutuante mínimo e máximo, respectivamente;

“Pec” = Posição de Elemento de Cenário;

“Neca” = Número de Elementos de Cenário Ativos;

“A” = Valor inteiro variável de acordo com o tema atual;

“Nec” = Novo Elemento de Cenário;

Tempo de Mudança de Cenário: Vinculada ao cálculo de aleatoriedade de mudança de cenários, tal fórmula randomiza determinado valor de tempo entre dados valores mínimo e máximo. O valor resultante configura efetivamente o tempo de mudança de cenário.

A fórmula, então, seria:

Se(Cst == verdadeiro)

Tmc = Randomizar(X, Y)

Onde:

“Cst” = Chave de Sorteio de Tempo

“Tmc” = Tempo de Mudança de Cenário;

“X” e “Y” = Valores inteiros mínimo e máximo, respectivamente.

Aleatoriedade de Mudança de Cenário: Tal fórmula se faz dependente dos resultados do cálculo de tempo de mudança de cenário. A seu turno, realiza um processo de randomização, e seus diferentes resultados correspondem aos diversos temas de cenário.

A fórmula então seria:

Se(C >= Tmc)

Nt = Randomizar(X, Y)

Cst = verdadeiro

Onde:

“C” = Contador de tempo, em segundos;

“Tmc” = Tempo de Mudança de Cenário;

“Nt” = Novo Tema;

“X” e “Y” = Valores inteiros mínimo e máximo, respectivamente;

“Cst” = Chave de Sorteio de Tema;

Origem e Aplicação: DEJ 3 - <<http://bit.ly/15u6CO1>>

6. Gravidade, Conjunto de Influências:

A gravidade se dá, em RunnerChief, nas formas constante e pontual. A primeira, funcional de forma natural e nativa por meio da biblioteca de física bidimensional do motor de jogo, segundo o valor de 9,81 para força de atração, sempre para baixo. A segunda, proveniente do elemento específico de cenário Planetóide, ativada ao contato com seu elemento “Atmosfera”, e funcional através da aplicação em código da fórmula base de força de atração da lei de gravitação universal, $F = G \cdot ((m1.m2) / d^2)$, agindo junto a um Vetor 2D correspondente à direção dessa atração, gerando um vetor final de forças também bidimensional a ser aplicado ao componente Corpo Rígido (Rigidbody 2D) do personagem jogável.

Assim, as fórmulas e procedimentos da segunda forma de gravidade são:

$$1. Ds = (Pps, Ppl)$$

$$F = G * ((Mps * Mpl) / Ds^2)$$

$$Dag = -(Pps - Ppl)$$

$$Ig = Dag * F$$

$$2. Se(Cca == verdadeiro)$$

Adicionar força Ig em Ps.Rigidbody2D

$$Ps.eY = -Dag$$

Onde:

“Ds” = Distância;

“Pps” e “Ppl” = Posições de personagem e planetóide, respectivamente.

“F” = Força de Atração;

“G” = Gravidade;

“Mps” e “Mpl” = Massas de personagem e planetóide, respectivamente;

“Dag” = Direção de Atração Gravitacional;

“Ig” = Influência Gravitacional;

“Cca” = Contato com Atmosfera;

“Ps.Rigidbody2D” = Componente de Corpo Rígido do Personagem;

“Ps.eY” = Eixo Y do Personagem;

Origem e Aplicação: DEJ 7 - v02 - <<http://bit.ly/1xBBrHP>>

7. Parkour, Lógica de Deslocamento:

O parkour, em RunnerChief, vem somar-se à movimentação padrão de corrida e pulo. Essa modalidade de deslocamento permite principalmente três formas de movimento extra: correr pelas paredes; agarrar-se e tomar impulso para completar uma escalada; e utilizar desníveis do cenário para tomar impulso. A realização desses deslocamentos se dá através de um único vetor específico de forças, adaptável segundo os contextos apresentados pelo cenário e adicionado aos personagens jogáveis sempre que detectada a possibilidade de sua aplicação. A detecção ocorre através do contato com colisores estrategicamente posicionados pelo cenário, enquanto a adaptação das forças aplicadas se dá naturalmente por conta dos diferentes posicionamentos e tamanhos dos colisores, que acarretam maiores ou menores extensões de pontos de contato.

Assim as fórmulas e procedimentos da segunda forma de gravidade são:

Enquanto(AP == verdadeiro)

Adicionar força Dp em Ps.Rigidbody2D

Onde:

“P” = Distância;

“AP” = Aplicabilidade de Parkour;

“Dp” = Deslocamento de Parkour;

“Ps.Rigidbody2D” = Componente de Corpo Rígido do Peronsagem;

Aplicação: DEJ 8 - <<http://bit.ly/1BIBs3f>>

7.6 Física do jogo

Há física 2D arcade no jogo.

7.7 Ações

No jogo, deve-se basicamente olhar, clicar e ler.

RunnerChief é, principalmente, um jogo de reflexos e estratégias rápidas. Assim, o jogador principalmente:

Olha tudo o que encontra-se na tela - feedbacks visuais, textos, ícones, movimentações e outros - a fim de reconhecer o que se passa e reagir da melhor forma possível.

Clica durante toda a interação, seja para adquirir itens, navegar pelos menus ou movimentar a personagem.

Lê os diversos botões e descrições presentes no jogo, a fim de movimentar-se através e agir nele, reconhecendo funcionalidades e traçando diferentes estratégias para melhor desempenho.

7.8 Economia

A economia do jogo funciona através da acumulação de recursos e aquisição de itens disponíveis na loja ingame.

O jogador, portanto, acumula recurso monetário durante as partidas através da coleta das moedas de fliperama mágicas espalhadas no mapa do jogo e, atingindo determinados valores, pode adquirir itens ou melhorias diversas para seus power-ups, devendo, pois, administrar tais recursos a fim de adquirir os itens que melhor auxiliem na evolução de sua estratégia.

7.9 Opções do jogo

Por ora, as únicas opções de jogo são de configuração de volume de trilha e efeitos sonoros.

7.10 Rejogando e Salvando o jogo

Uma vez iniciado, o jogo salva seu estado automaticamente, em pontos chave durante a navegação. Assim, o jogador só poderá iniciar um novo estado, do “zero”, ao reinstalar o jogo.

O rejogo pode dar-se sempre ao término de uma partida, onde pergunta-se ao jogador se ele deseja ou não tentar novamente, ou ainda pelo encerramento e reabertura do jogo. Dessa forma, o jogador e o estado do jogo progridem através dos muitos rejogos, das diversas partidas terminadas e recomeçadas.

8. Projeto Artístico

Por ora, há apenas referências artísticas dos futuros elementos do jogo. Com a evolução da produção efetiva, adicionar-se-á as artes produzidas e mais definitivas.

8.1 Cortes de Cena:

Os únicos cortes de cena até o momento se dão ao início do jogo na Tela Inicial, e no começo da partida quando do menu principal.

Pré PXP:

1. No primeiro, Annes é visto cozinhando magicamente, tranquilo e alegre em sua cozinha mágica, numa animação procedural, que poderá ser repedita indefinidas vezes até que o jogador saia da tela inicial.

2. No segundo, Xilef chama por Annes apressadamente, e o Chef Mágico corre para auxiliá-lo em vista das ocorrências descritas pelo gato mágico.

Pós PXP:

1. No primeiro, Annes é visto cozinhando magicamente, tranquilo e alegre em sua cozinha encantada junto a sete gatos mágicos e Xilef, numa animação procedural, que poderá ser repedita indefinidas vezes até que o jogador saia da tela inicial.

2. No segundo, uma grande mão de fumaça mágica invade a cozinha encantada e leva Annes. Xilef então lidera a Brigada Arco-íris ao resgate, saindo pelo buraco deixado na parede do lugar.

Origem: DEJ 3 - <<http://bit.ly/15u6CO1>>

9. Interface

9.1 Sistema Visual

9.1.1 Visão Geral e Evolução da Interface

Por ora, não há evolução de interface, pois seu processo de produção não foi ainda iniciado.

9.1.2 HUD (Head-Up Display)

Até o momento, o jogo contém as seguintes HUDs:

Pré PXP:

Pontuação, Gatos Mágicos Salvos, Moedas Mágicas Coletadas, Distância Percorrida.

Pós PXP:

Pontuação, Novelos Mágicos Coletados, Moedas Mágicas Coletadas, Distância Percorrida.

Origem: DEJ 3 - <<http://bit.ly/15u6CO1>>

9.1.3 Menus

O jogo, até o momento, contém os seguintes menus:

Menu Principal, Loja Ingame, Menu de Pause, Menu de Tentar Novamente.

9.1.4 Câmera

Câmera Ortogonal em 2D.

9.2 Sistema de Controle

Todo o controle do jogo se dá através de toques na tela, desde a navegação nos menus e telas, até a interação com itens e movimentação da personagem.

9.3 Sistema de Áudio

Um sistema de áudio estéreo, 3D.

9.3.1 Músicas

Por ora, não há músicas escaladas para RunnerChief.

9.3.2 Efeitos sonoros

Por ora, não existem sons específicos escalados para os efeitos sonoros de RunnerChief. Entretanto, estão planejados efeitos como:

Passos, colisão, explosão, coleta de moedas,

Pré PXP: salvamento de Gatos Mágicos, coleta de power-ups, aquisição de melhoria de power-ups, motor de moto, efeito remetente a arco-íris, e ainda outros a serem adicionados.

Pós PXP: coleta de Novelos Mágicos, coleta de power-ups, aquisição de melhoria de power-ups, motor de moto, efeito remetente a arco-íris, e ainda outros a serem adicionados.

Origem: DEJ 3 - <<http://bit.ly/15u6CO1>>

9.4 Sistema de Ajuda

No momento, o único sistema de ajuda do jogo é uma simples e discreta animação no canto da tela ao início da partida, onde vê-se: “dedo tocando a tela + personagem pulando”.

9.5 Inteligência Artificial

Não existe IA.

9.6 IA de NPCs

Não existe IA. Há apenas regras e cálculos de aleatoriedade para os power-ups concedidos por Xilef.