

Victor Freitas Caraméz

**DESENVOLVIMENTO DE UM PROTÓTIPO DE JOGO 2D:
ANIMAÇÃO, GAME DESIGN E SONORIZAÇÃO**

Projeto de Conclusão de Curso submetido(a) ao Programa de Graduação da Universidade Federal de Santa Catarina para a obtenção do Grau de Bacharel em Animação.
Orientadora: Prof^ª. Dr^ª. Mônica Stein

Florianópolis

2019

Victor Freitas Caraméz

Desenvolvimento de um protótipo de jogo 2D: Animação, Game Design e Sonorização

Este Trabalho de Conclusão de Curso (TCC) foi julgado adequado para obtenção do Título de Bacharel em Animação e aprovado em sua forma final pelo Curso de Animação da Universidade Federal de Santa Catarina.

Florianópolis, 26 de Novembro de 2019.

Prof. Flávio Andaló, Dr. Coordenador do Curso de Animação UFSC

Banca Examinadora:

Prof^ª. Monica Stein, Dr^ª. (UFSC)



Documento assinado digitalmente
Monica Stein
Data: 08/02/2021 10:38:12-0300
CPF: 886.707.249-87

Prof. Flávio Andaló, Dr. (UFSC)

Prof. Gabriel de Souza Prim, Ms. (UFSC)

Este trabalho é dedicado a todos que compartilharam da experiência de criação deste projeto comigo. Valeu.

AGRADECIMENTOS

Agradeço primeiramente à Bianca, pessoa que amo partilhar a vida. Com você tenho me sentido mais vivo de verdade. Obrigado pelo carinho, a paciência e a capacidade de trazer paz todos os dias na correria dos semestres.

Também agradeço a Augusto e Poffo, que em conjunto tornaram este projeto possível. Sem vocês, literalmente, esse projeto não seria possível.

Aos colegas e amigos que sempre estiveram ali por mim nos momentos difíceis e por dividirmos as maravilhosas sessões de RPG nos finais de semana que conseguíamos marcar alguma coisa.

Aos companheiros de banda, que definitivamente são profissionais e proporcionaram muita diversão durante a produção deste projeto.

Quero agradecer também aos familiares, que me incentivaram nos momentos que eu precisava, meus priminhos principalmente, me alegrando quando podíamos passar tempo juntos.

E finalmente agradecer à professora Mônica, que ajudou a desenvolver este trabalho, pela paciência na orientação e incentivo que tornaram possível o desenvolvimento deste trabalho.

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Caramez, Victor Freitas
DESENVOLVIMENTO DE UM PROTÓTIPO DE JOGO 2D : ANIMAÇÃO,
GAME DESIGN E SONORIZAÇÃO / Victor Freitas Caramez ;
orientador, Monica Stein, 2019.
53 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Centro de
Comunicação e Expressão, Graduação em Animação, Florianópolis,
2019.

Inclui referências.

1. Animação. 2. Jogos Digitais. 3. Animação. 4. Som. 5.
Game Design. I. Stein, Monica. II. Universidade Federal de
Santa Catarina. Graduação em Animação. III. Título.

RESUMO

O presente trabalho tem como proposta explicitar o processo de criação, bem como execução de animações em conjunto com game design, além da sonorização para o desenvolvimento de um protótipo de jogo 2D, para a plataforma de PC, no estilo de jogos multijogador local. O projeto possui o objetivo de mostrar a necessidade de se pensar em game design enquanto animador, para a produção de um jogo. A metodologia escolhida para o desenvolvimento do projeto foi a Double Diamond, mostrando-se efetiva e eficiente com relação ao estabelecimento de escopo e organização do fluxo de trabalho no grupo. A relevância do material é a evidente importância, no mercado de jogos, do animador pensar não só na área de animação, mas no processo de criação de games como um todo.

Palavras-chave: Animação. Som. Jogos digitais. Game Design. Multiplayer Local.

ABSTRACT

This work aims to explicit the creation, as well as execution process of animations together with game design, alongside sound for the development of a 2D game prototype, for PC, in a local multiplayer style. The project intends to show the need to think about game design as an animator to develop a game. The methodology chosen here was Double Diamond, shown to be effective and efficient in relation to the establishment of scope and workflow organization in the group. This work's relevancy comes from the fact that, in the digital games' market, the animator needs to not only think about animation, but also in the whole game developing process.

Keywords: Animation. Sound. Digital games. Game Design. Local Multiplayer.

LISTA DE FIGURAS

Figura 1 - Representação gráfica da metodologia Double Diamond	15
Figura 2 – Metodologia Double Diamond aplicada no projeto	16
Figura 3 - Frame da animação Sinfonia Amazônica	18
Figura 4 - Arte do jogo TowerFall Ascension feita pelo estúdio Mini Boss	19
Figura 5 - Core Gameplay Loop de Super Mario Bros.	23
Figura 6 - Core Gameplay Loop de Clash of Clans	24
Figura 7 – Gameplay loop TowerFall Ascension	25
Figura 8 - Jogo Super Smash Bros	27
Figura 9 - Jogo TowerFall Ascension	28
Figura 10 - Jogo Wand Wars	28
Figura 11 - Jogo Nidhogg	29
Figura 12 - Jogo Speedrunners	29
Figura 13 - Core gameplay loop do jogo	31
Figura 14 - Interface Godot	31
Figura 15 - Protótipo inicial do jogo	32
Figura 16 - Protótipo com alguns efeitos e sprites implementados	33
Figura 17 - Interface do PyxelEdit	34
Figura 18 - Jogos Samurai Gunn, Wand Wars e Towerfall Ascension	35
Figura 19 - Estética synthwave	36
Figura 20 – Lista de animações e efeitos sonoros produzidos	37
Figura 21 - Frames do idle de um dos personagens	38
Figura 22 - Frames do walk dos personagens com smears	39
Figura 23 - Frames do walk de um dos personagens	39
Figura 24 - Frames do crawl	40
Figura 25 - Frames de transição	40
Figura 26 - Frames da animação de charge	41
Figura 27 - Frames do dash	41
Figura 28 - Animações de colisão	42
Figura 29 - Frames da animação de morte de um personagem	43
Figura 30 - Animações do rabo genéricas	43
Figura 31 - Animações de renascimento	44
Figura 32 - Frames das animações dos itens	44
Figura 33 - Frames das animações de efeitos	45
Figura 34 - Interface FL Studio	45
Figura 35 - Interface Bfxr	46
Figura 36 - F-Zero, Hotline Miami e Far Cry 3: Blood Dragon	47
Figura 37 - Timeline da produção da música	48
Figura 38 - Timelines de algumas faixas	49
Figura 39 – Screenshots do protótipo finalizado	50

SUMÁRIO

SUMÁRIO	11
1.INTRODUÇÃO	12
1.1 OBJETIVOS	12
1.1 Objetivo geral	12
1.2 Objetivos específicos	12
1.2 JUSTIFICATIVA	13
1.3 METODOLOGIA	13
1.4 DELIMITAÇÕES	16
2.DESENVOLVIMENTO TEÓRICO	17
2.1 ANIMAÇÃO	17
2.1.1 Primórdios da animação	17
2.1.2 Os princípios de animação	18
2.1.3 Animação nos jogos	19
2.2 GAME DESIGN	21
2.2.1 O que é game design?	21
2.2.2 Core gameplay loop e mecânicas	22
2.3 SONORIZAÇÃO	24
2.3.1 Sonorização em seus contextos históricos	24
2.3.2 Sonorização nos jogos	24
3.DESENVOLVIMENTO	26
3.1 GAME DESIGN	26
3.1.1 Pesquisa de referências e proposta	26
3.1.2 Mecânicas e desenvolvimento do protótipo	31
3.2 ARTE	33
3.2.1 Estilo gráfico e referências	33
3.2.2 Processo de animação	37
3.2.2.1 Personagens e itens	38
3.2.2.1.1 Idle	38
3.2.2.1.2 Walk	38
3.2.2.1.3 Crawl	40
3.2.2.1.4 Transições de estado	40
3.2.2.1.5 Charge	41
3.2.2.1.6 Dash	41
3.2.2.1.7 Colisões	42
3.2.2.1.8 Morte	43

3.2.2.1.9 Rabo	43
3.2.2.1.10 Itens	44
3.2.2.2 Efeitos	45
3.3 SONORIZAÇÃO	45
3.3.1 Escolha do programa e pesquisa de referências	45
3.3.2 Trilha Sonora	44
3.3.2.1 Faixa 1	47
3.3.2.2 Faixas 2 até 6	48
3.3.3 Efeitos Sonoros	49
3.4 PROTÓTIPO FINALIZADO	50
4.CONSIDERAÇÕES FINAIS	51
5.REFERÊNCIAS	53

1.INTRODUÇÃO

A indústria de jogos cresceu exponencialmente desde seus primórdios nos anos 70. E com esse crescimento, as demandas e oportunidades de trabalho subiram em conjunto. Uma das necessidades para a produção de um jogo é a animação, sendo a grande parte da experiência visual de um jogo.

As animações nos jogos avançaram cada vez mais, considerando a crescente necessidade dos jogadores por qualidade. Jogos que buscam o fotorrealismo são cada vez mais presentes na indústria de jogos AAA, utilizando-se das tecnologias que avançam dia após dia. Já no caso dos jogos *indie*, este nível de qualidade não pode ser atingido, devido a diversos fatores. Seja a equipe reduzida, ou até mesmo as carências de programas, os jogos *indie* tendem a ser mais simples, tanto em escopo mecânico quanto visual.

Na tentativa de alcançar o sucesso, os jogos *indie* tentam possuir propriedades mais sutis, buscando soluções alternativas às grandes produtoras. Levando estes fatores em consideração, tais *games*, na busca por atenção de um maior público, por vezes tentam obter uma qualidade de animação acima da média como um dos pontos de venda.

O projeto aqui desenvolvido busca explorar a integração da animação com o *game design*, desenvolvendo as animações para um protótipo de jogo digital 2D, bem como elaborar um *game design* ágil e simples, que dê suporte para as animações que serão desenvolvidas.

O estudo de animação em conjunto com *game design* é importante para a formação de um bom profissional da área de jogos. Cada vez mais o profissional necessita entender de todos os processos presentes no desenvolvimento de um jogo, considerando que muitos estúdios na área possuem uma equipe reduzida.

A metodologia utilizada no trabalho foi a Double Diamond, sendo uma metodologia ágil e funcional para uma equipe pequena. Ela dá um caminho a ser seguido no sentido de organização dos momentos de desenvolvimento e os momentos de pesquisa, guiando o processo como um todo até um produto final.

Os cenários do projeto desenvolvido, por sua vez, foram produzidos por Augusto Kerber, portanto não sendo explorados no trabalho aqui descrito.

O trabalho foi dividido em quatro capítulos, que por sua parte possuem subdivisões. No capítulo de desenvolvimento teórico, o texto foi dividido entre *game design* e animação, dando um embasamento tanto histórico quanto teórico. O capítulo do desenvolvimento é separado entre animação, *game design* e sonorização, mas agora mostrando os processos e os resultados do projeto. O capítulo de considerações finais, por sua parte, mostra o projeto finalizado, bem como as etapas que ocorreram como planejado ou não.

1.1 OBJETIVOS

1.1.1 Objetivo geral

- Produzir a animação e o áudio de um protótipo de jogo 2D.

1.1.2 Objetivos específicos

- Discutir fatores que auxiliam as animações desenvolvidas através do *game design*;

- Criar o *game design* do jogo especificando as animações a serem desenvolvidas no projeto pelo animador;

1.2 JUSTIFICATIVA

Apesar das ferramentas e princípios de animação presentes em outros meios de comunicação aplicarem-se para as animações de jogos, os processos e técnicas se mostram diferentes. O animador deve pensar em todas as interações possíveis que o jogador pode ter com o objeto sendo animado antes de começar o processo em si, além das implicações de timing e afins que podem ocorrer. Os jogos que se utilizam de um *game design* em conjunto com a animação acabam adicionando à experiência dos jogadores, como Mariel Cartwright demonstra em sua palestra de 2014 “*Making Fluid and Powerful Animation for Skullgirls*”.

Outro fator auxiliar numa melhor experiência do jogador é o som. Na palestra “*Juice it or lose it*” de Petri Purho e Martin Jonasson, ambos explicam como a sonorização de um jogo pode auxiliar nas sensações tanto de impacto quanto de movimento dos jogadores em um determinado ambiente.

O projeto desenvolvido aqui busca aplicar o conteúdo explicitado ao longo dos capítulos do texto, produzindo tanto as animações estipuladas pelo *game design*, quanto os sons do jogo.

1.3 METODOLOGIA

A metodologia escolhida pelo autor para ser aplicada no projeto é a metodologia *Double Diamond*, desenvolvida pelo IDEO e utilizada pelo *Design Council*. Esta metodologia consiste em 4 etapas:

Discover

A primeira etapa do processo busca ampliar e expandir ao máximo as ideias e o conhecimento relacionados ao projeto. O mínimo possível de cortes é realizado, visando quantidade, não qualidade.

Nesta etapa, as ideias iniciais para o jogo foram propostas e discutidas entre o grupo, com uma noção geral das mecânicas que existirão no projeto desenvolvido.

Define

Esta etapa busca filtrar todo o conteúdo produzido na etapa anterior, reduzindo os caminhos a serem seguidos. Neste momento, busca-se qualidade, no lugar de quantidade, com o intuito de limitar o escopo e opções somente àquilo que é viável. Ao fim desta etapa, define-se o problema em questão.

Este momento teve a criação da arte conceitual do jogo, uma maior definição das mecânicas que aparecerão no jogo e um desenvolvimento inicial das animações, pensando mais no tempo pela qual as mesmas irão acontecer.

Develop

O terceiro passo do método, expande em cima dos resultados obtidos na etapa anterior e da definição do problema. Aqui é onde, em geral, ocorre a prototipação, as tentativas de novas ideias e experiências para solucionar problemas.

Aqui aconteceu a finalização do protótipo inicial do jogo com sprites implementados e mecânicas mais bem definidas. O produto final ainda não está completo, porém a maior parte da parte de criação “bruta” acontece aqui.

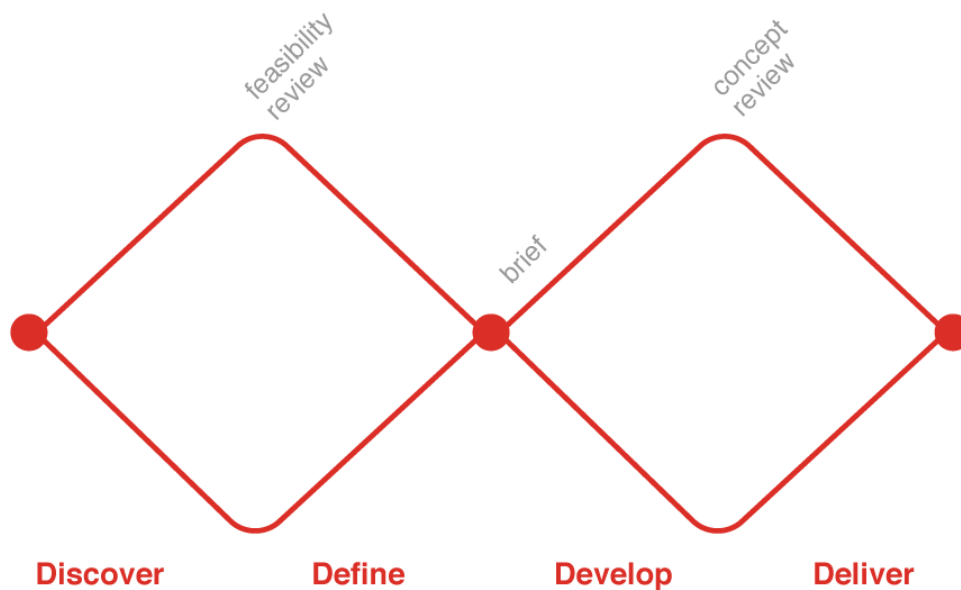
Delivery

Por fim, o segundo diamante é fechado com o refinamento da solução escolhida na fase anterior, marcando a entrega da concepção da ideia, onde o produto será finalizado.

Finalmente, o protótipo é finalizado, com as mecânicas finais já definidas e implementadas, bem como as animações e cenários.

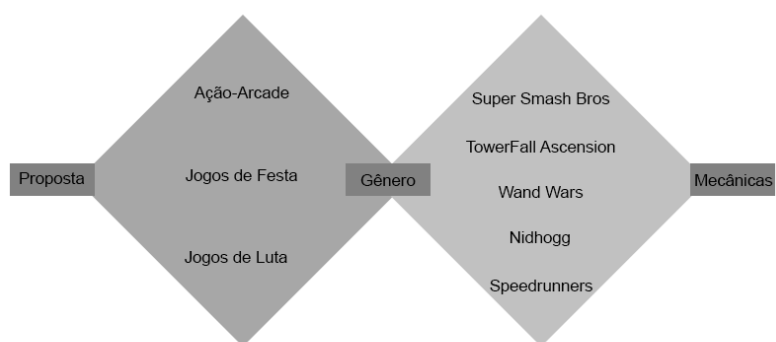
A metodologia *Double Diamond* foi escolhida por se mostrar efetiva em evidenciar problemas ao longo do processo de produção, deixando potenciais ajustes mais fáceis de identificar e aplicar, sem uma perda substancial de tempo extra.

Figura 1 - Representação gráfica da metodologia *Double Diamond*



Fonte: <https://developers.google.com/web/fundamentals/design-and-ux/ux-basics/?hl=pt-br>

Figura 2 – Metodologia Double Diamond aplicada no projeto



Fonte: Desenvolvida pelo autor

1.4 DELIMITAÇÕES

O jogo desenvolvido neste trabalho é 2D, com estética *pixel art*, para PC. As mecânicas básicas do personagem, e todas as interações entre os jogadores, bem como os espaços por onde estes vão estar presentes são os pontos produzidos.

O game design desenvolvido serve somente para a obtenção de um produto de animação coerente às suas especificidades, servindo somente como suporte para o material visual do jogo produzido no projeto.

Outra parte importante que terá uma exploração menor será a da criação dos conceitos para personagens e cenários, além de todas as técnicas e processos envolvidos. A maior parte da conceitualização de personagens e afins, além da criação de cenários foi realizada por Augusto Kerber, portanto, não será descrita detalhadamente aqui.

Portanto, a parte de programação não será abordada neste projeto, salienta-se que a programação foi feita por um terceiro integrante do grupo que desenvolve o game objeto deste PCC. Este não integra o quadro de alunos da UFSC, e a programação, portanto, não será apresentada/estudada aqui.

2 DESENVOLVIMENTO TEÓRICO

Ao longo deste capítulo, o embasamento sobre os conteúdos a serem desenvolvidos será apresentado.

2.1 ANIMAÇÃO

No capítulo a seguir, tanto a base histórica da animação, bem como sua aplicação nos jogos será explorada.

2.1.1 Primórdios da animação

A animação pode datar seus primeiros passos até às primeiras imagens registradas nas paredes das cavernas antigas, onde o ser humano buscava mostrar cenas de caça, e representações de animais, sugerindo movimento. Vários instrumentos foram desenvolvidos desde essa época para criar impressões de movimento. Um dos experimentos, valeu-se de uma sequência de vinte e quatro câmaras escuras que captavam progressivamente a passagem de um cavalo. Esta experiência levou a várias outras pessoas começarem a tentar inventar seus próprios dispositivos para captura de imagens sequenciais, até que chegemos nas tecnologias de hoje.

Richard Williams (2016, pg 15) diz que o antecessor do que conhecemos como desenho animado foi o filme criado por Thomas Edison e James Stuart Blackton, de 1906, chamado “Fases cômicas de faces engraçadas”. A animação era primitiva, porém abriu as portas para as futuras gerações. No Brasil, a história da animação já é mais recente, tendo na década de 50 o primeiro longa-metragem de animação produzido no país, chamado “Sinfonia Amazônica”.

Figura 3 – Frame da animação Sinfonia Amazônica



Fonte: <https://revistaintertelas.com/2018/10/26/cinemateca-capitolio-celebra-dia-mundial-da-preservacao-audiovisual-com-sessoes-especiais/#more-2828>

Atualmente as empresas brasileiras de desenvolvimento de jogos já participam mais ativamente num nível global no mercado. Empresas como *Behold Studios* ou *Mini Boss* têm em seus catálogos produtos que ganharam reconhecimento mundial.

Figura 4 – Arte do jogo TowerFall Ascension feita pelo estúdio Mini Boss



Fonte: <https://blog.studiominiboss.com/towerfall>

2.1.2 Os princípios de animação

Os doze princípios de animação foram introduzidos pelos animadores Ollie Johnston e Frank Thomas, depois dos anos trabalhando como animadores chefe nos estúdios *Disney*. Estes princípios buscam demonstrar pontos que, caso aplicados, virão a melhorar a qualidade da animação sendo produzida. Estes princípios são ensinados em muitas escolas de animação, devido a sua onipresença nas produções da indústria. Tammy Coron traz:

Esses princípios formam a base de todo trabalho de animação e são relevantes para um número de campos diferentes. O uso mais óbvio é para animar um personagem, mas essas regras também são guias indispensáveis em outras áreas, por exemplo, se quiser introduzir movimento em sua interface com um pouco de animação CSS. (CORON, 2019, tradução nossa)

Estes princípios são importantes para a produção de uma animação de qualidade, porém não são uma regra definitiva e imutável. Eles podem ser utilizados de maneiras diferentes em situações diferentes, como será explorado em capítulos posteriores. Tendo isto em vista, seguem os princípios:

- *Squash and stretch*: *Squash and stretch* são as deformações que algo sofre, de acordo com o movimento. O objeto ou personagem precisam manter a massa geral quando mudarem de forma.

- Antecipação: Toda ação precisa de alguma antecipação, um movimento precedendo tal ação, que mostre ao espectador o que vai acontecer em seguida.

- *Ease in e ease out*: A aceleração e desaceleração que definem a velocidade inicial e final de cada um dos movimentos do personagem, dando mais fluidez.

- Timing: A velocidade da ação do objeto ou personagem, dando a percepção de peso da mesma.

-Arcos: Movimentos em forma de arco, com o intuito de deixar a ação mais orgânica.

-*Straight ahead e pose-to-pose*: Este princípio mostra as duas formas principais que animadores usam para criar uma cena. A animação *pose-to-pose* é quando são desenhados os principais movimentos de uma personagem, e só então, são feitos os movimentos

intermediários. Já no modelo *straight ahead*, o artista anima o objeto quadro a quadro, podendo criar novos movimentos à medida que desenha.

-*Follow through e overlapping action*: Quando os objetos pararem depois de estarem em movimento, diferentes partes do objeto serão interrompidas em taxas diferentes. Da mesma forma, nem tudo em um objeto se moverá na mesma proporção.

-Ação secundária: Ações secundárias que apoiam ou enfatizam a ação principal que ocorre em uma cena.

-Exagero: Extrapolação da ação, com o objetivo de deixar a cena mais impactante, não se importando com parâmetros realistas.

-Appeal: A clareza do que está sendo representado, a fim de deixar a leitura da ação mais clara.

-*Staging*: O propósito deste princípio é direcionar a atenção da audiência, e deixar claro qual é a ação de maior importância na cena.

-Solid drawing: É manter a forma e volume do objeto ou personagem, independente do ângulo, dando a impressão de tridimensionalidade.

2.1.3 Animação nos jogos

Nas animações para filmes ou mídias afins, o animador só é responsável por animar o que está na frente da câmera. Isso permite o animador a cortar muitos caminhos, sabendo que a cena só será vista de um ângulo. A audiência não tem controle sobre nenhuma das ações tomadas na tela.

Animações em jogos, por sua vez, são um dos meios de comunicação do jogador com o jogo. São a resposta da intenção do jogador, seja essa intenção uma boa história, personagens cativantes, um combate fluido, ou até mesmo influenciar outros jogadores de uma maneira positiva ou negativa. Estas animações possuem o fator da interatividade do jogador com o conteúdo durante a exibição dos *frames*.

Pense em algo simples como um soco em um jogo de luta. O quão satisfatório esse soco parece enquanto o jogador aperta o botão de ataque é principalmente devido à qualidade do conteúdo de animação. Ele pode informar o jogador de informações de *gameplay design* como quão aberto ele está para retaliação depois de apertar para socar, e quão forte é o soco. Ajuda a contar ao jogador quem o personagem é, quão forte ele é, e quanta convicção emocional está por trás do soco. E ajuda com o fluxo direcional de como o olhar do jogador está sendo levado pela tela. Este simples soco pode comunicar arcs e um fluxo que pode ser confortável ou desconfortável. Ainda assim, é meramente uma animação de soco. Animação de *gameplay* é ativa, comunicação de cá para lá. Enquanto animações para filmes são uma experiência passiva. Com jogos, o jogador tem este *input* para dizer ao jogo o que ele espera, e como o jogo responde é uma medida de experiência. Com filmes, o espectador tem expectativas, e ainda assim não tem um meio de interação. Esperam que seja uma via de mão única onde o filme alimenta o espectador. Jogos precisam comunicar tudo que um filme precisa comunicar. Uma experiência que seja ressonante com o

espectador/jogador é de suma importância para ambos. Mas animadores de *gameplay* têm a complicação adicionada das interações do jogador, e como essas interações influenciam o resto do quebra-cabeça. Diversas vezes em jogos, uma resposta propriamente dita entra em conflito consigo mesma. Por exemplo, criar um soco robusto, com bastante peso se beneficia de ter uma grande antecipação e tempo de assentamento, como você vê nos filmes. Mas, para ele ser responsivo numa situação de combate, uma antecipação extremamente curta e tempo de assentamento onde o jogador pode interromper é ideal. Achar soluções criativas para estes conflitos é algo único das animações de *gameplay*, algo que eu acho extremamente desafiador e interessante. (LICO, Richard, 2017, tradução nossa)

Tendo isto em vista, alguns dos princípios de animação explanados anteriormente precisam, por vezes, ser aplicados de maneiras diferentes das estipuladas nas suas definições. Jonathan Cooper traz:

Antecipação é um tópico controverso nos videogames, com o designer frequentemente pedindo o mínimo possível (de antecipação) e animadores forçando para o máximo de *frames* possível. Pouca antecipação e o movimento desejado, como um soco ou balanço de espada, terá pouco peso (uma componente chave para o feedback do jogador, não somente estético). Antecipação demais e o movimento irá parecer não responsivo, removendo agência do jogador e reduzindo o sentimento de controlar o avatar diretamente. (COOPER, 2019, p. 01, tradução nossa)

As interações do jogador com as animações não são o único motivo de pensar na conexão do *game design* com a animação. Cooper itera também que a antecipação para ações de personagens não controlados pelo jogador é intencionalmente mais longa, de modo que o jogador consiga ler a antecipação, e planejar suas ações de acordo.

É possível ver no jogo *Monster Hunter World* (2017) como o cuidado com as animações de antecipação é fundamental para a experiência do jogo. A possibilidade de ler as ações futuras dos inimigos, traz a sensação de responsabilidade do jogador quando o mesmo é acertado por um ataque, reduzindo a frustração se comparado a um golpe com uma antecipação mínima. Este jogo também conta com uma variedade na velocidade da antecipação dos ataques do próprio jogador, devido à escolha do equipamento do mesmo. Todos estes fatores mostram como a animação pode auxiliar o *game design*.

Um dos animadores do jogo *Destiny* (2014), Richard Lico, fala em uma entrevista para o site *80lv* que a interatividade do jogador influencia o modo em os animadores devem pensar em suas animações. Considerando o gênero do jogo em questão, a velocidade e responsividade dos controles é importante não só para um entendimento do que acontece na tela, mas também no balanceamento do jogo.

Contraste é chave! Saber o tom e ritmo de todas as animações que você está fazendo torna adicionar ênfase pelo contraste fácil. Por exemplo, um soco em *Destiny* começa com uma ação rápida, segura depois do impacto, e volta na ação rápida para o retorno. Este contraste conta a história da ação, já que a pausa após o impacto explica o que aconteceu no soco que a precedeu. Mas o contraste nesse soco não é tão bom do que o que você encontra quando solta um *super* em *Destiny*. Eu faria esse contraste mais extremo para ajudar a vender a grandeza da ação. Fazer com que pareça maior que um soco com

um maior risco pela recompensa. Mesmo dentro de um clipe, eu deixo onde adicionar contraste e ênfase baseado em onde eu quero chamar a atenção do jogador. Nós estamos fazendo um jogo sobre um rato na Polyarc. Ela vai por vezes apontar a objetos que queremos que o jogador note. Então para chamar atenção, eu começo a animação de apontar com um *pop* abrupto, assentado então no ponto. Este contraste faz com que seja fácil para o jogador perceber o rato, e então seguir sua direção para o objeto que precisamos que você veja. (LICO, Richard, 2017, tradução nossa)

Vê-se então uma necessidade de pensar na animação para jogos de maneira diferenciada. Todas as interrupções possíveis devido às interações do jogador devem ser pensadas no momento da produção das animações.

2.2 GAME DESIGN

Aqui, a exploração se dará nos aspectos de *game design* estudados.

2.2.1 O que é *game design*?

Jogos possuem vários aspectos, sendo um deles o seu *game design*. *Game design* é a tomada de decisões que compõem um jogo. A criação de objetivos, regras e desafios enquadram-se no campo do mesmo. Jesse Schell (2008) traz:

Game design é o ato de decidir o que um jogo deve ser. É isso. Na superfície, isso soa muito simples. ‘Você quer dizer que você pode fazer o *design* de um jogo fazendo uma só decisão?’ Não. Para decidir o que um jogo é, você deve fazer centenas, geralmente milhares de decisões. ‘Eu não preciso de algum equipamento especial para fazer o *design* de um jogo?’ Não. Já que *game design* é simplesmente a tomada de decisões, você pode fazer o *design* de um jogo na sua cabeça. Geralmente, no entanto, você vai querer escrever estas decisões, pois nossas memórias são fracas, e é fácil perder alguma coisa importante se não anotar essas coisas. Ademais, se quiser que outras pessoas ajudem a fazer decisões, ou a construir o jogo, você precisa comunicar estas decisões para eles de algum jeito, e escrever é um bom modo de fazer isso. (SCHELL, 2001, tradução nossa)

Entretanto, um bom *game design* não é tão facilmente definido. Por vezes, escolhas que seriam consideradas *game design* ruim são parte do que torna o jogo especial. Stanislav Costiuc (2018) descreve:

Por exemplo, não botar *checkpoints* automáticos frequentes é um mau *game design*? Se você falhar em algum desafio particularmente difícil, ou somente precisa sair do jogo, nós não queremos botá-lo muito pra trás, isso seria muita punição. Logo, *checkpoints* não frequentes são ruins... Mas também são bons, como no *Alien: Isolation*, onde seu sistema de *save points* manual põe seus *save spots* bem distantes uns dos outros. Bem, nós sabemos que jogos sem algum cancelamento de animação são ruins. Se você começar a socar um inimigo e vir algum outro atacando, você precisa poder apertar um botão e cancelar seu ataque para contra-atacar com sucesso. Isso faz sentido certo? Você não sabia que o outro cara ia atacar, você precisa poder adaptar. A falta de cancelamento de animação é ruim... Mas também é bom, como em *Dark Souls* onde o comprometimento com suas ações é o que adiciona tanta profundidade a seu sistema de combate.

Hm... Bem, você sabe, *Alien: Isolation* é um jogo de terror e *Dark Souls* é, bem, *Dark Souls*. Faz sentido esses jogos subverterem as normas estabelecidas e ainda estar tudo bem, certo? Oh, eu sei, vamos olhar para o *progress tracker*. Jogos podem tomar uma grande quantidade de tempo, então é simplesmente bom *game design* mostrar quanto do jogo você já viu, de alguma maneira e se tem alguma coisa que você ainda não viu. Não permitir rastrear seu progresso é ruim...

Mas também é bom, como em *The Stanley Parable*, onde o ponto do jogo é descobrir todos os finais diferentes sem saber quantos faltam exatamente. Espere, então se todas as normas de bom *design* também possuem exemplos de bom *design* que fazem exatamente o oposto... O que é então bom *game design*? (COSTIUC, Stanislav, 2018, tradução nossa)

Cada vez mais estudos sobre *game design* aparecem, tendo em vista a importância do assunto com relação ao crescente interesse no desenvolvimento de jogos. O assunto, também, muda constantemente, como a maior parte dos aspectos provenientes da profissão, adaptando-se às novas experiências e tecnologias.

2.2.2 Core gameplay loop e mecânicas

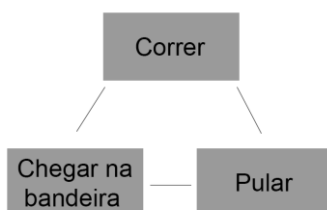
Kevin Wolstenholme (2017) define o *gameplay loop* como:

O core loop é essencialmente o batimento cardíaco do seu jogo. É uma série ou cadeia de ações que repete várias vezes sendo o fluxo primário da experiência dos seus jogadores. (WOLSTENHOLME, 2017, tradução nossa)

Portanto, o core gameplay loop pode ser definido como a essência de um jogo, a parte principal do que o jogador vai fazer enquanto joga, todos seus obstáculos e elementos que criam a experiência do *player*.

Os primeiros jogos digitais já possuíam essa esquematização da experiência, e isso continua até hoje. Por exemplo em *Super Mario Bros.* (1985) temos o seguinte *loop* de *gameplay*:

Figura 5 – Core Gameplay Loop de *Super Mario Bros.*



Fonte: Desenvolvido pelo autor

A simplicidade no *loop* é essencial para a retenção do jogador. Uma grande quantidade de mecânicas não vai necessariamente tornar um jogo melhor ou mais divertido. Jogos digitais modernos, principalmente na área dos *mobiles*, precisam conquistar o jogador nos primeiros minutos de *gameplay*, então simplicidade é um dos fatores mais importantes no momento de definição do *core gameplay loop* de um jogo. *Clash of Clans* (2012), um jogo aclamado, traz

um *gameplay loop* simples, mas que permite uma grande customização por parte do jogador, atraindo muito do público de jogos *mobile* pela acessibilidade e profundidade alcançável. Nathan Lovato (2017) traz:

“Quanto mais caminho o jogador tem que percorrer pelo *loop*, maior deverá ser a qualidade do mesmo. Em dispositivos móveis ou na internet, se as mecânicas principais não são intuitivas, os controles não são polidos, você vai perder seu público dentro de alguns minutos. As pessoas só vão lhe dar uma única chance de entretê-las.” (LOVATO, Nathan, 2017, tradução nossa)

Figura 6 – Core Gameplay Loop de Clash of Clans

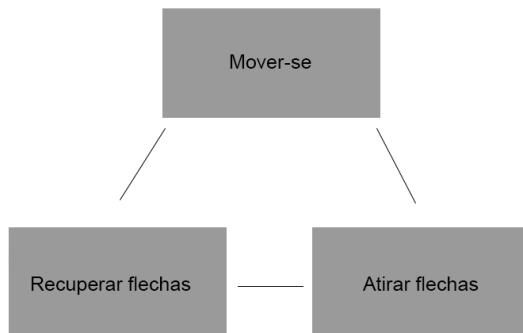


Fonte: <https://risinghighacademy.com/what-is-a-core-loop-in-a-mobile-game/>

As mecânicas por sua vez são os elementos que fazem parte do *gameplay loop*. Após definidas pelo *game design*, a execução das mecânicas necessita ser coerente com suas definições, tendo em vista as implicações que o design explicita.

Jogos de *multiplayer* local em geral trazem alguma mecânica principal bem evidenciada, além de contar com vários jogadores interagindo ao mesmo tempo, tornando cada partida diferente, o que se torna um elemento de atração de jogadores, continuando divertido mesmo após várias sessões de jogo. *Towerfall Ascension* (2013), por exemplo, traz um *loop* bem reduzido baseado em mover-se, atirar flechas e recuperar flechas. A pouca variedade de ações, porém bem executadas e o fator *multiplayer* tornam o jogo pouco repetitivo.

Figura 7 – Gameplay Loop TowerFall Ascension



Fonte: *Desenvolvido* *pele* *autor*

2.3 SONORIZAÇÃO

Assim como nos capítulos anteriores, uma base sobre sonorização será vista aqui.

2.3.1 Sonorização em seus contextos históricos

Os sons são usados de várias maneiras, muito especialmente para comunicação por meio da fala ou, por exemplo, música. A percepção do som também pode ser usada para adquirir informações sobre o ambiente em propriedades como características espaciais e presença de outros animais ou objetos.

Mais de 2000 anos antes do cinema, o teatro clássico grego usava alguns recursos interessantes, como colocar o coro fora do palco, às vezes na parte superior do teatro e rodeando as arquibancadas. Dessa forma, colaborava para melhor mergulhar o espectador e “envolvê-lo” no duplo sentido na palavra, na ação da peça.

Ao longo da história do ser humano, as aplicações do som expandiram-se, trazendo à tona experiências novas. Uma destas novas sensações foi o som no cinema, que inicialmente era tocado ao vivo em conjunto com o filme reproduzido.

Em relação ao cinema, a primeira experiência do uso de sons que não vêm unicamente das caixas que estão por trás da tela, foi o filme “Fantasia” da companhia Disney em 1941, utilizando um sistema desenvolvido pela companhia e chamado “Fantasound”. Conforme as tecnologias avançaram, desenvolveram-se novas técnicas de exposição de som.

2.3.2 Sonorização nos jogos

O som é parte fundamental de qualquer jogo. Em games com temática de guerra, de tiro em primeira ou terceira pessoa, ouvir o que se passa no ambiente é muito importante. Seja um inimigo rastejando por perto ou então uma bomba lançada de longe, escutar o que está ao seu redor é determinante.

O jogo *Call of Duty*, vencedor do prêmio de “Melhor Jogo de Ação em Primeira Pessoa” do Video Game Awards de 2003 é provido de um trabalho sonoro condizente com a realidade do jogo. Contendo efeitos de explosões, tiros, balas ricocheteando, entre outros, o som se mistura com os gritos e ordens do líder do esquadrão de modo que passa ao jogador a sensação de estar

imerso em um campo de batalha. A fragilidade da parte sonora, contudo, refere-se à dublagem de algumas das personagens, que ao não serem adequadamente trabalhadas prejudicam a questão imersiva em alguns casos específicos.

3 DESENVOLVIMENTO

A etapa de desenvolvimento deste projeto foi dividida entre *game design*, arte e sonorização, bem como suas próprias subdivisões.

3.1 GAME DESIGN

Aqui exploraremos o desenvolvimento do *game design* no jogo produzido.

3.1.1 Pesquisa de referências e proposta

A primeira proposta de *game design* para o jogo desenvolvido baseava-se em um *gameplay* de velocidade, dando prioridade a animações de impacto e velocidade, de forma que se pudesse usar bastante do princípio de *squash and stretch*. Outro fator importante definido era de que o jogo seria *multiplayer*. O objetivo era explorar um gênero de jogos que os membros da equipe se sentiam mais confortáveis em desenvolver algo, devido ao interesse mútuo e experiências prévias. De maneira ideal, o jogo teria partidas rápidas, de até quatro jogadores, retendo o jogador pelo tempo que ele quisesse.

A primeira etapa, então, da expansão da metodologia proposta se iniciou, mostrando alguns subgêneros baseados nas definições de Scott Rogers, no livro *Level Up*. O jogo proposto acaba por se encaixar tanto no gênero de ação, quanto no gênero de jogos de festa, sendo um jogo destinado a vários jogadores simultâneos, baseado em um modelo competitivo de interação.

Algumas definições propostas por Scott na aba de jogos de ação não se encaixam com a proposta desenvolvida, devido à necessidade de velocidade e interação entre jogadores do projeto. Jogos *beat 'em up*, *stealth*, *action-adventure* e *fighting* englobam outros aspectos não incorporados no jogo desenvolvido aqui, dentre eles a necessidade de uma história a longo termo, a ênfase em evitar a detecção dos inimigos, entre outros. Os jogos de luta mais se assemelham ao jogo proposto, porém necessitam de uma profundidade maior na jogabilidade, seja nos controles ou na quantidade de combinações de ataques possíveis.

Os subgêneros de jogos de ação, descritos por Rogers, enquadram somente jogos de um único jogador, porém algumas das definições aplicam-se ao jogo desenvolvido. Jogos *action-arcade*, e *platformers* possuem, entre outras características, movimentação por um cenário, por vezes desafiador, e jogabilidade rápida, bem como placares e tempo de jogabilidade curto. Alguns exemplos de jogos que se enquadram nessas predeterminações são *Dig Dug*, *Diner Dash* e *Super Mario World*.

Os jogos de festa, apesar de não possuírem uma definição muito detalhada no livro de Rogers, também se encaixam bem na temática desejada, sendo definidos como jogos especificamente projetados para múltiplos jogadores e baseados na jogabilidade competitiva. Jogos que exemplificam este tipo de jogo, segundo Rogers, são *Mario Party* e *Buzz!*.

Os jogos de *arcade*, segundo Scott, são jogos com ênfase e *gameplay* de reflexos, pontuação e jogabilidade rápida, mostrando-se enquadrar bem nas definições propostas. Passa-se então para o próximo passo da expansão do desenvolvimento do jogo. Rogers diz que os gêneros de jogos aparecem em todos os tamanhos e formas, que não se deve ter medo de misturá-los, logo, o jogo a ser produzido terá elementos do gênero de jogos de festa, assim como de *arcade*, além de contar com alguns elementos de outros gêneros descritos anteriormente, em menor escala.

Jogos provenientes da era dos *arcades*, nos anos 80, tinham um foco no uso contínuo de uma única mecânica, onde o jogador precisaria utilizá-la em contextos diferentes e cada vez mais desafiadores, de forma que a experiência continuasse divertida e levasse mais tempo até tornar-se morosa. O objetivo principal de muitos títulos da época era arrecadar dinheiro, em função do valor monetário acoplado às fichas necessárias para jogar. Títulos como *Dig Dug* e *Diner Dash*, citados pelo autor, mostram um *core gameplay loop* simplificado, que se mantém divertidos mesmo após inúmeras partidas. Pensando nisso então, começou-se a fase de pesquisa de referências de títulos que pudessem agregar com mecânicas no jogo a ser feito.

Seguindo na lógica dos jogos com um *gameplay loop* simples, em 1999, foi lançado o jogo *Super Smash Bros*, publicado pela Nintendo. O título misturava mecânicas de jogos de luta com jogos de plataforma e jogos de festa. O loop de *gameplay* era simples, movimentar-se pelo cenário e eliminar os oponentes, que eram controlados por outros jogadores.

Figura 8 – Jogo Super Smash Bros



Fonte: <https://www.nintendo.pt/Jogos/Nintendo-64/Super-Smash-Bros--269756.html>

A partir da inspiração pela simplicidade elegante de *Super Smash Bros*, a segunda etapa do processo da metodologia iniciou-se. Neste momento, buscou-se testar mecânicas e *gameplay loops* que poderiam ser agregados ao projeto a ser produzido, tendo em mente os conceitos de jogabilidade rápida e mecânicas simplificadas. O título que se destacou, num primeiro momento, foi *TowerFall Ascension*, de Matt Thorson. O jogo é uma ramificação dos conceitos de *Super Smash Bros*. Cada jogador controla um personagem, que pode dar um *dash* e atirar uma flecha. O jogador tem um limite de flechas a atirar, e, caso o jogador erre as flechas, ele pode recuperá-las onde elas caírem. Também, ao longo de cada partida, aparecem itens aleatoriamente pelo mapa, que podem deixar o jogador mais forte. O jogo apresenta uma mecânica de jogo de plataforma, assim com *Smash*, além do fator *multiplayer*.

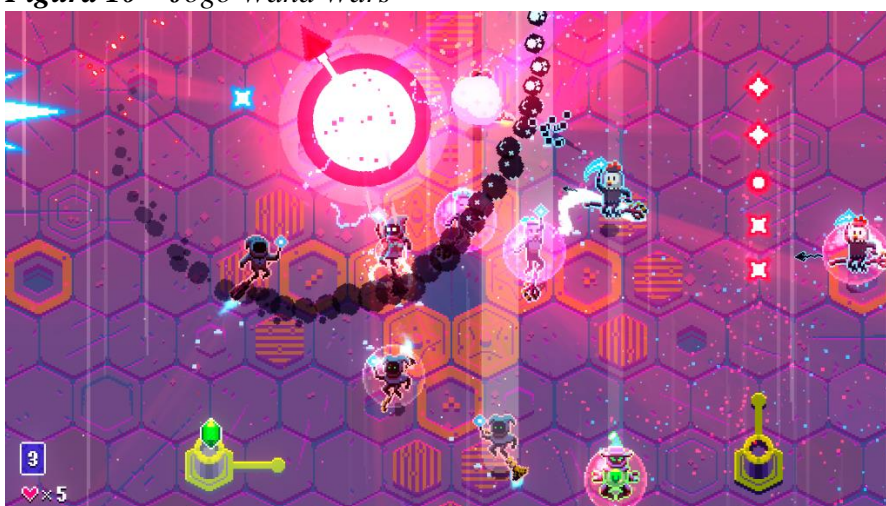
Figura 9 – Jogo TowerFall Ascension



Fonte: https://store.steampowered.com/app/251470/TowerFall_Ascension/

Outro título que mostrou relevância na pesquisa foi Wand Wars, de MoonRadish. O game consiste de uma mecânica simples, onde os jogadores tem que rebater uma bola entre si, podendo tentar desviar da mesma, numa espécie de partida de queimada. O modo de interação com a bola, permite que o jogador a segure e selecione a direção na qual quer mandá-la, aumentando a velocidade da mesma toda vez que isso acontece. O intuito do jogo é, como em TowerFall, alcançar um placar maior do que os outros jogadores, seja sobrevivendo por mais tempo, ou eliminando mais jogadores.

Figura 10 – Jogo Wand Wars



Fonte: <https://moonradish.itch.io/wand-wars>

Um jogo que também se mostrou pertinente foi Nidhogg, da desenvolvedora Messhof, de Los Angeles. Outro jogo que também tem uma mecânica simples, porém intuitiva além de um gameplay fácil e rápido. O estilo gráfico é minimalista, com os personagens tendo somente uma cor, sem sombreamento ou qualquer outro tipo de definição além da silhueta. No jogo, dois jogadores disputam um duelo de esgrima, podem ajustar a altura de suas espadas. O

objetivo é eliminar o inimigo e tentar passar por algumas salas até chegar na base do outro jogador. Com cada morte do personagem dando uma margem para avanço do sobrevivente. O loop de gameplay do jogo é simples, assim como nos outros, e dá bastante espaço ao jogador para criar momentos memoráveis, não sendo repetitivo.

Figura 11 – Jogo Nidhogg



Fonte: <https://store.steampowered.com/app/94400/Nidhogg/>

Em 2016, foi lançado o jogo *Speedrunners*, da desenvolvedora DoubleDutch Games. O título tem como princípio a velocidade. Quatro jogadores disputam uma corrida em um ambiente de plataforma, podendo deslizar pelas paredes ou atirar ganchos, que funcionam como impulsos e auxílios de movimento para os jogadores. Espalhados ao longo dos cenários ficam caixas de itens, que podem dar algumas vantagens para quem alcançá-las primeiro. Os itens variam de mísseis para fazer com que algum outro jogador tropece, até um gancho que, quando acerta algum inimigo, faz com que os dois jogadores troquem de lugar, alterando o ritmo da partida. Caso os jogadores sobrevivam por muito tempo, as bordas da tela começam a diminuir, tornando cada colocação abaixo da primeira mais perigosa. As partidas são rápidas, assim como os movimentos dos personagens, que possuem cada um uma estética bem definida, dando bastante personalidade para cada um deles.

Figura 12 – Jogo Speedrunners



Fonte: <https://store.steampowered.com/app/207140/SpeedRunners/>

Os títulos mostraram-se coerentes com a proposta do jogo desenvolvido, devido aos seus estilos gráficos, *gameplays* ágeis e simples, animações fluídas e que adicionam ao peso das ações ocorrendo na tela, grande utilização de efeitos, tornando a experiência visual tanto

impressionante quanto estimulante, e *game design* inteligente, tendo uma grande repetibilidade com controles simples e poucas mecânicas.

Assim, os elementos de *gameplay* que se mostraram significativos para o projeto foram:

- *Gameplay* em plataforma com cenário limitado de *TowerFall Ascension*, assim como o sistema de flechas limitadas;

- *Seleção de direção* de *Wand Wars*, bem como a mecânica da “batata quente”;

- Combate simplificado de *Nidhogg*;

- Itens espalhados pelos mapas de *Speedrunners*;

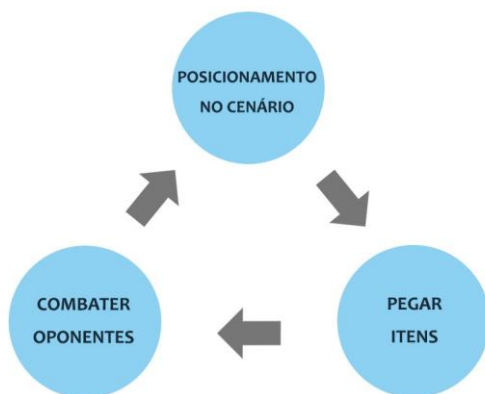
- Aspecto *multiplayer*;

A mecânica da bola sendo jogada de um jogador para outro se mostrou ser de uma complexidade alta, podendo comprometer outros aspectos do desenvolvimento do jogo. Por este motivo, optou-se por removê-la do jogo, substituindo-a por uma mecânica de *dash*.

O *core gameplay loop* consistiria do jogador eliminar os outros jogadores com o *dash*, e alcançar o maior placar possível, dentro de um cenário limitado. Os jogadores, ao levarem um primeiro *hit*, se tornariam vulneráveis para aí então levar um ataque letal, aumentando o placar de quem matou, e, indo para a próxima partida, até um placar máximo de dez vitórias, onde um vencedor é definido.

Os jogadores podem se movimentar pelas paredes do cenário, onde itens podem aparecer, dando algum tipo de bônus ao jogador, seja um escudo que dá uma proteção para um *hit* ou mais velocidade ao jogador. Os itens aparecem de forma aleatória, e os players precisam dar um *dash* no item para conseguir alcançá-lo.

Figura 13 – Core *gameplay loop* do jogo



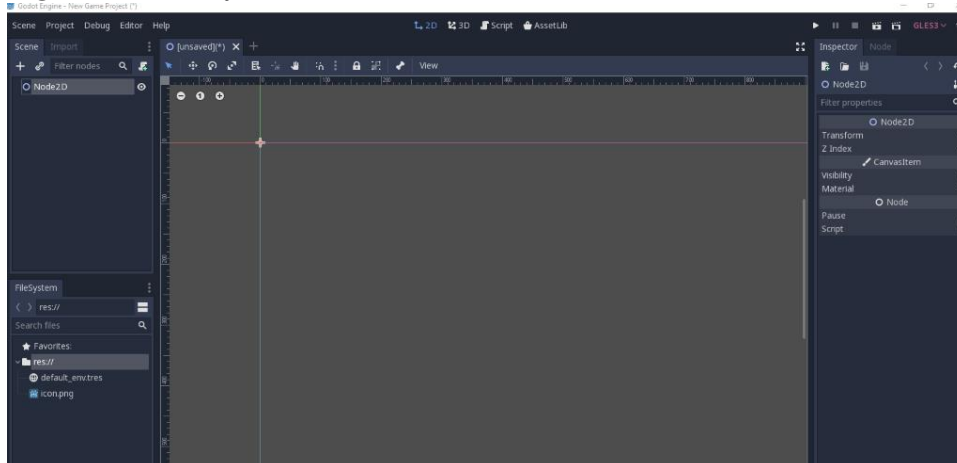
Fonte: Desenvolvido pelo autor

3.1.2 Mecânicas e desenvolvimento do protótipo

A *engine Godot* é *open source*, e vem se expandindo no mercado. Não existem muitos jogos desenvolvidos neste programa até o momento, justamente pelo fato de que é um motor relativamente novo. É uma *engine* que vem sendo elogiada por diversos desenvolvedores

independentes, dando mais possibilidades de customização que a *engine Game Maker*. A linguagem de programação utilizada no programa é mais simples que a da *Unity*, sendo então a escolha do programador para o desenvolvimento do protótipo do jogo do trabalho.

Figura 14 - Interface Godot



Fonte: Screenshot tirado pelo autor

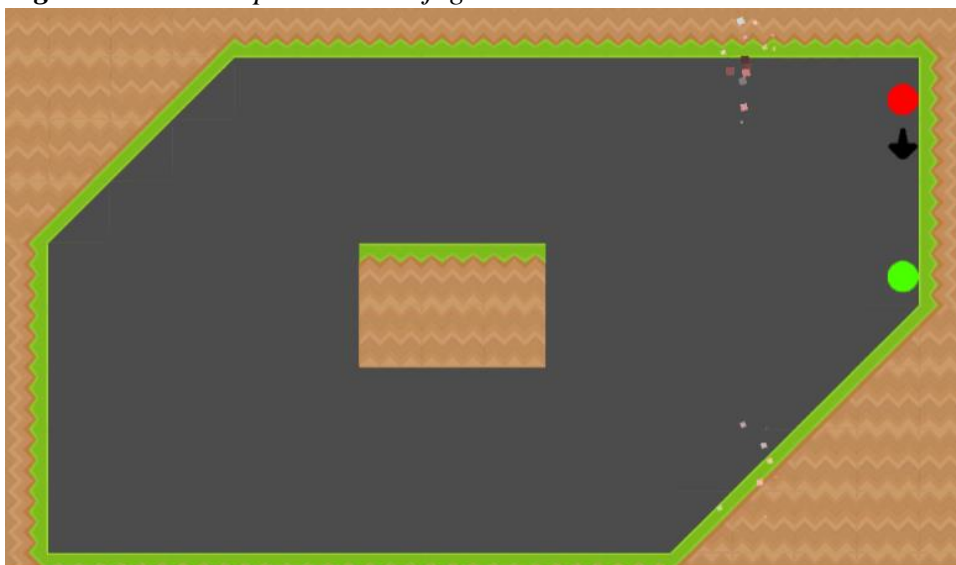
Após a escolha do programa para desenvolvimento do projeto, iniciou-se a criação do primeiro protótipo. O primeiro passo foi definir as mecânicas básicas do jogo. Baseado nos exemplos anteriores, o jogador teria a possibilidade de movimentar-se, tanto pelas paredes quanto no chão, e um *dash*, que também pode ser utilizado para movimentação. Cada jogador possui três usos de *dash* disponíveis, que após esgotados, demoram alguns segundos para voltar.

Os jogadores têm como objetivo eliminar os outros jogadores, então as interações possíveis entre cada um incluem colisões dos *dashes*, que não resultam em eliminação, colisão do *dash* de algum jogador em outro, onde o jogador que não estava no *dash* fica vulnerável então para uma eliminação, e colisão de *dash* com um jogador vulnerável, assim eliminando o mesmo.

A progressão ao longo do jogo ocorre a partir das eliminações dos jogadores, passando para uma próxima partida a partir do momento onde somente um jogador estiver vivo. O objetivo dos jogadores é tanto sobreviver o maior tempo possível quanto eliminar a maior quantidade de jogadores possível, utilizando-se do *dash* tanto para fugir quanto atacar.

A prototipagem inicial incluiu todas as mecânicas básicas para o funcionamento do jogo, como o *dash* e a movimentação, não mostrando nenhum aspecto gráfico. Desta forma, foram usados somente blocos e formas básicas para as representações dos personagens e do cenário, tendo um primeiro estabelecimento de velocidade tanto dos *dashes* quanto da movimentação base dos personagens.

Figura 15 - Protótipo inicial do jogo

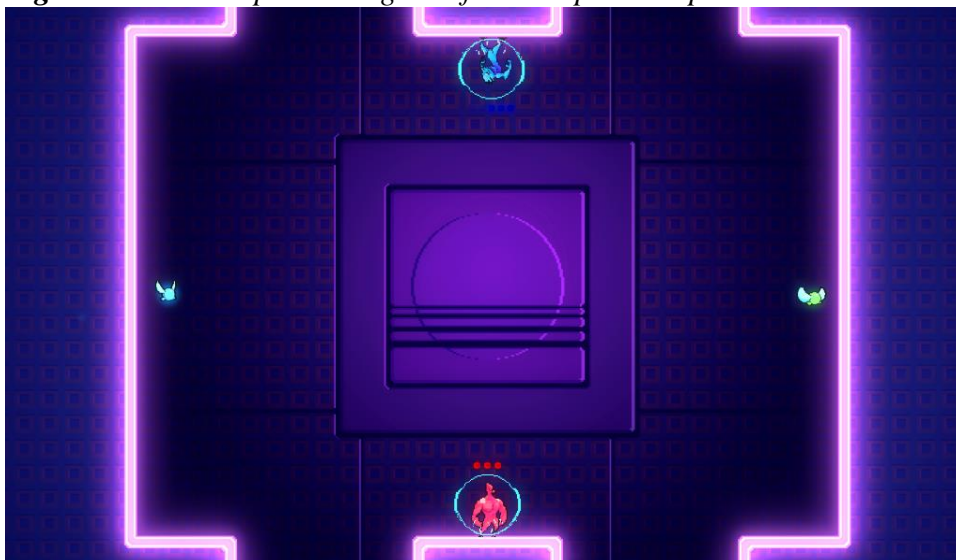


Fonte: Screenshot feito pelo autor

O tamanho escolhido para os *sprites* dos personagens, que são as animações implementadas no jogo, foi de 64 *pixels* de altura por 64 *pixels* de largura. Estes *sprites* têm o seu tamanho então duplicado enquanto dentro do jogo, para uma melhor visualização na tela. Este tamanho foi escolhido pelo fato de ser pequeno o suficiente para enaltecer o estilo *pixel art*, sendo também possível de ser produzido no tempo estipulado.

Com as mecânicas definidas, passou-se para a implementação das artes do jogo, cujo processo será explicitado no capítulo a seguir.

Figura 16 - Protótipo com alguns efeitos e sprites implementados



Fonte: Screenshot feito pelo autor

3.2 ARTE

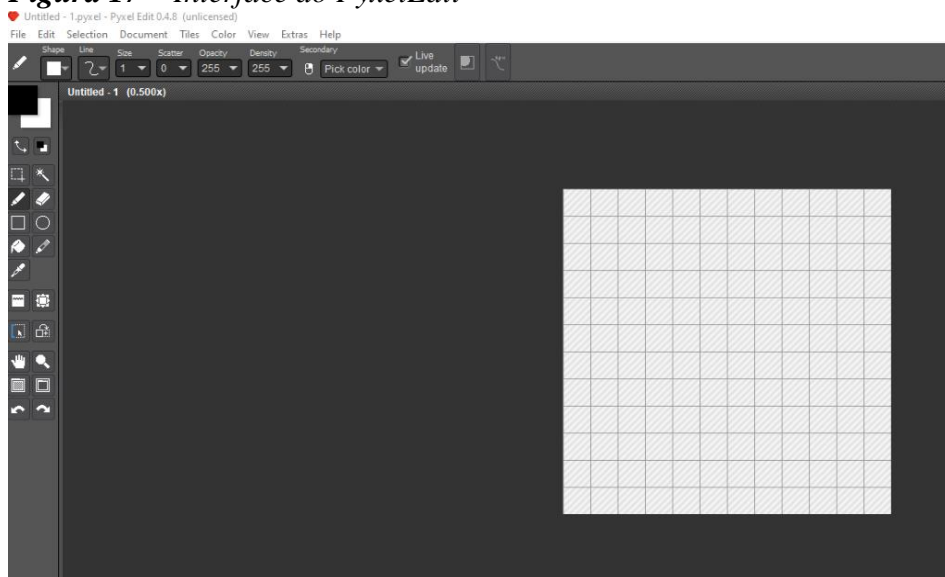
Nos capítulos a seguir, as animações desenvolvidas terão os seus respectivos processos explicitados.

3.2.1 Estilo gráfico e referências do jogo

Como explanado anteriormente, o estilo gráfico escolhido foi o de *pixel art*, que é uma forma de arte digital na qual as imagens são criadas ou editadas tendo como elemento básico os *pixels*. Elementos gráficos provenientes de sistemas computacionais antigos ou relativamente limitados, como consoles de videogames e telefones celulares são considerados *pixel art*. Inicialmente este tipo de arte precisava ser feito através de códigos, porém hoje temos tecnologias mais avançadas, que permitem um controle artístico maior sobre o conteúdo realizado, sem a necessidade do conhecimento de programação requerido anteriormente.

Alguns dos programas que podem ser utilizados para a criação das animações neste estilo são *Pyxel Edit* e *Aseprite*. Ambos possuem uma interface limpa e ágil, porém o programa escolhido foi *PyxelEdit*, devido à chave de produto já adquirida previamente por um dos membros do grupo. O programa possui funções similares a outros programas de edição de imagens, separando seu *canvas* em *tiles*, permitindo a criação de quadros de animação lado a lado.

Figura 17 – Interface do PyxelEdit



Fonte: Screenshot tirado pelo autor

A estética *pixel art* foi escolhida tanto pela familiaridade dos integrantes do grupo quanto pelo seu uso recorrente nos títulos do gênero definido. A estética permite uma prototipação rápida, além de remeter à era dos jogos mais antigos, que é outra parte da estética escolhida. O intuito do game design é de que o jogo fosse ágil e veloz, logo a animação deveria seguir o mesmo caminho.

Foram analisadas as animações dos títulos *Towerfall Ascension*, *Samurai Gunn* e *Wand Wars* para a produção do projeto. Todos os títulos fazem parte do gênero proposto anteriormente, e obtiveram uma estética que o grupo achou condizente com o jogo que estava sendo desenvolvido.

Figura 18 – Jogos Samurai Gunn, Wand Wars e Towerfall Ascension



Fonte: Montagem feita pelo autor

Os títulos descritos, em especial *TowerFall Ascension*, trazem uma estética semelhante aos jogos antigos dos *arcades*, onde o próprio Matt Thorson, criador de *TowerFall*, cita em uma entrevista para IGF:

A intenção original da arte era poder fazê-la sozinho. Eu não sou um artista visual muito habilidoso, então escolhi limitações que me permitiriam realizar um estilo consistente e legível: originalmente o jogo tinha uma resolução de 320x240 e a paleta de cores do NES. Quando Pedro tomou as rédeas do pixel art e refez tudo, nós largamos a paleta, mas mantivemos a resolução pequena. O estilo do Pedro é bem gótico e sombrio – o mundo do jogo e música acabaram tomando várias dicas disso. Eu amo pixel art como uma estética, eu amo como ela torna as coisas mais abstratas e deixa os jogadores preencherem os vazios. Tem um aspecto nostálgico para muitas pessoas, mas nossa intenção nunca foi maximizar isto. (THORSON, 2014, tradução nossa)

Outro ponto estético importante que foi decidido para o projeto foi o estilo *synthwave*. É um estilo também conhecido como retrofuturismo, e remete também à época dos *arcades*, utilizando o espaço imagético do que se imaginava ser o futuro nos anos 80. O *synthwave* agrega elementos do *cyberpunk* e *vaporwave*, utilizando-se de cores contrastantes e a sensação de velocidade.

Figura 19 - Estética synthwave

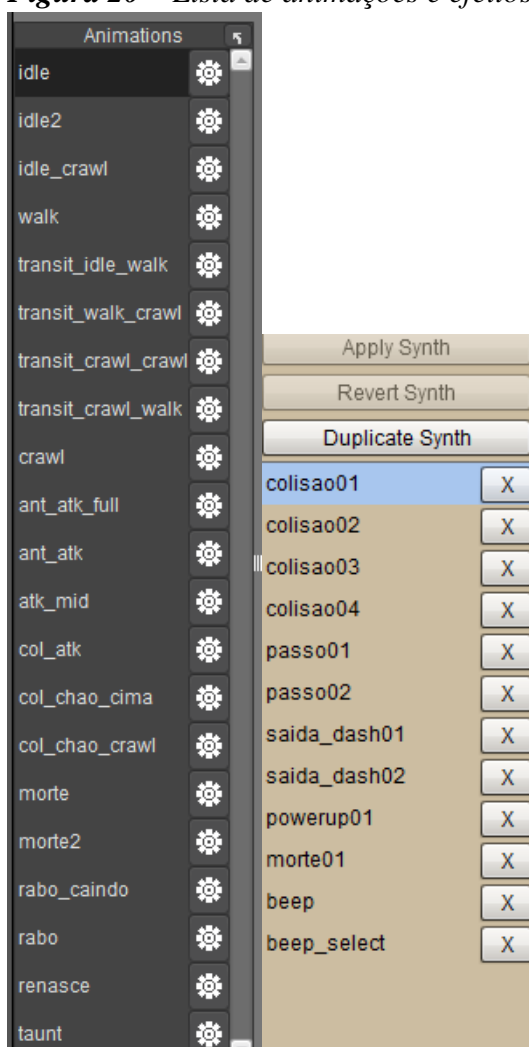


Fonte: Montagem desenvolvida pelo autor

3.2.2 Processo de animação

Este capítulo será dividido entre as animações de personagens e animações de efeitos, bem como os processos e delimitações envolvidos na produção de ambos de acordo com o *game design* proposto. De maneira a explicitar melhor os movimentos produzidos, as figuras mostradas ao longo do capítulo mostrarão somente o *outline* das animações.

Figura 20 – Lista de animações e efeitos sonoros produzidos



Fonte: Screenshot tirado pelo autor

3.2.2.1 Personagens e itens

Após decidido o estilo gráfico do projeto, as animações base começaram a ser produzidas conforme as solicitações do *game design*. Como uma maneira de organizar o processo de produção, uma lista de animações necessárias para a base do personagem ficar completa foi feita. As animações representam os estados nos quais os personagens vão estar. Assim como os personagens, os itens e *power-ups* também foram animados, com seus estados próprios. As animações são:

3.2.2.1.1 Idle

As animações de *idle* são animações onde o personagem está em um estado de repouso, em contato com o chão, enquanto não toma nenhuma ação. Tais animações em geral não possuem uma tentativa de ação grande, ou mesmo muito movimento. No caso do jogo sendo produzido, os personagens possuem um *idle* principal, e um *idle* secundário, que acontece no começo da partida, demonstrando um pouco da personalidade do personagem.

O processo de criação destes loops de animação é simples, em geral mostrando o personagem respirando, ou tendo algum movimento simples e sutil. O intuito das animações é manter o personagem em movimento enquanto não tomando ações, então no caso do projeto desenvolvido fizemos o personagem respirando.

Figura 21 – Frames do *idle* de um dos personagens



Fonte: Desenvolvido pelo autor

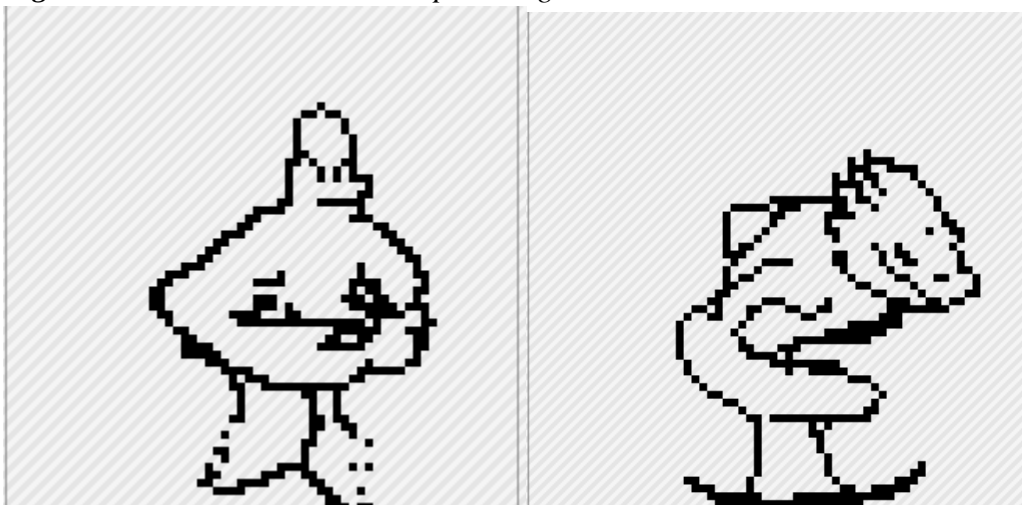
As animações de *idle* são feitas em *loop*, bem como as animações de *walk*, e *crawl*. As animações foram divididas basicamente em oito *frames*, sendo destes três principais e cinco intermediários. Os frames principais mostram o ápice tanto da movimentação de expansão da respiração quanto da retração, e também o *frame* inicial base. Os *frames* intermediários interpolam os *frames* principais, dando preferência pelos mesmos, dando um *ease* maior tanto de entrada quanto de saída dos movimentos, deixando o *frame* do meio como um momento de maior velocidade.

3.2.2.1.2 Walk

Walks são as animações de movimentação básica do personagem. Diversos jogos utilizam *walks* em velocidades diferentes, seja pela capacidade de modular a velocidade do jogador por sensibilidade de controle, ou algum tipo de modificador por botão, como um botão de correr. O projeto deste trabalho possui dois tipos de *walks*, com o básico sendo uma animação do personagem correndo e a outra animação sendo explanada no próximo item.

Optou-se por utilizar do recurso de *smears* nesta animação, auxiliando na visualização da velocidade do personagem. *Smears* são borrões de movimento utilizando em momentos de alta velocidade do objeto sendo movido, no caso do personagem, as pernas.

Figura 22 – Frames do walk dos personagens com smears



Fonte: Desenvolvido pelo autor

Na animação de walk foi utilizado o “método da pose baixa” (2016) de Richard Williams proposto em seu Manual de Animação. No método, primeiramente são feitas as poses mais baixas da corrida do personagem, para em seguida as poses de subida serem realizadas. Após este processo, são produzidos os frames de interpolação.

De maneira geral, os ciclos de caminhada dos personagens são diferentes entre si, apesar de seguirem o mesmo método de produção. Alguns têm mais *followthrough* nos braços, outros são mais duros na movimentação, tentando buscar uma diferenciação de personalidade entre os personagens.

Figura 23 – Frames do walk de um dos personagens



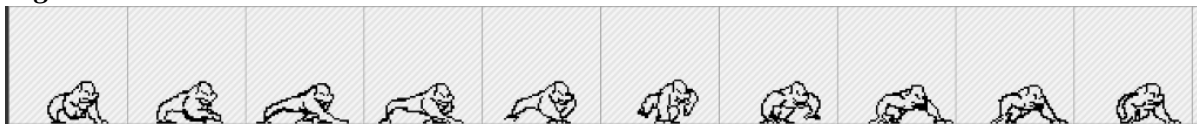
Fonte: Desenvolvido pelo autor

3.2.2.1.3 Crawl

Semelhante à animação de *walk*, o *crawl* é um conjunto de frames que representa uma movimentação básica do personagem, porém em um momento onde o personagem está com todas as extremidades de seus membros encostando no chão.

Esta animação é utilizada quando o personagem anda pelas paredes ou teto dos cenários, pois a opinião do grupo foi de que utilizar um *walk* normal para as movimentações não convencionais, que desafiariam a gravidade, causaria um estranhamento nos jogadores.

Figura 24 – Frames do *crawl*



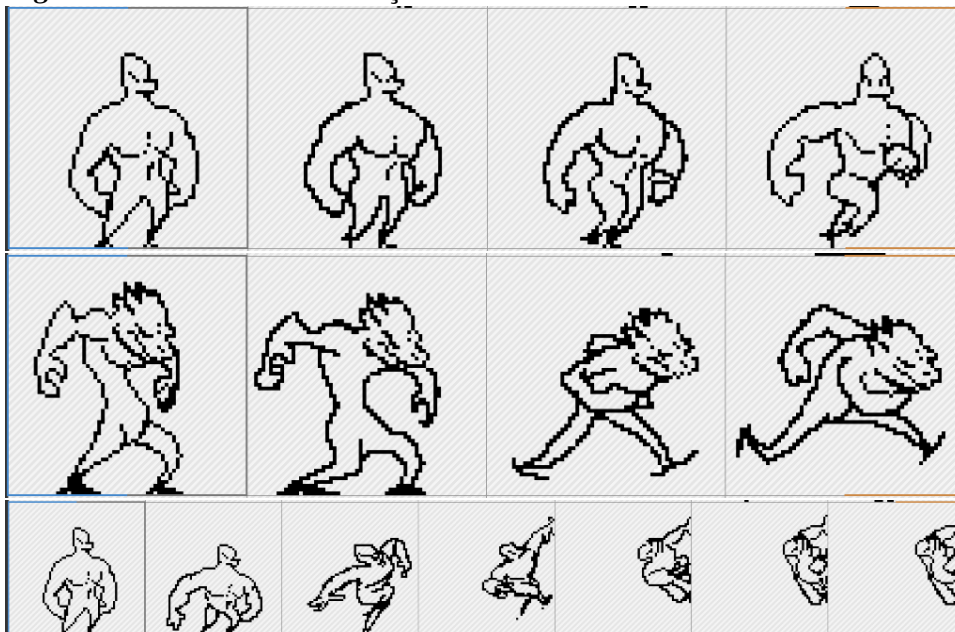
Fonte: Desenvolvido pelo autor

3.2.2.1.4 Transições de estado

As animações de transição de estado servem para conectar algumas animações, como a troca de estado entre *idle* e *walk*, ou entre *walk* e *crawl*, entre outras. Em geral, estas animações devem ser rápidas, caso contrário o jogo pode ficar com uma cadência lenta, prejudicando as ideias discutidas pelo *game design*.

Este tipo de animação tem um número de frames limitados, tendo em vista suas implicações caso contrário. Tendo isto em vista, algumas concessões foram realizadas no sentido de implementação dos princípios. As antecipações e *eases* são rápidos, quase inexistentes, não deixando muito tempo para interpretação do jogador.

Figura 25 – Frames de transição



Fonte: Desenvolvido pelo autor

Para as animações mostradas nas imagens acima, os frames iniciais não são considerados no momento de importação para o jogo em si, visto que já estarão presentes nas animações as quais precederam.

3.2.2.1.5 Charge

A animação de carregamento do *dash* demonstra o momento de antecipação ao ataque do personagem. Duas animações foram produzidas para este estado, uma mostrando o processo de carregamento e outra para quando o personagem já carregou o máximo de energia para um ataque.

As animações, durante os momentos de gameplay, podem não acontecer completamente, devido à capacidade do jogador de liberar o ataque rapidamente, portanto a animação acaba também por ser mais simplificada, tentando somente mandar uma mensagem para os outros jogadores de que alguém está preparando um *dash*.

Figura 26 – Frames da animação de charge



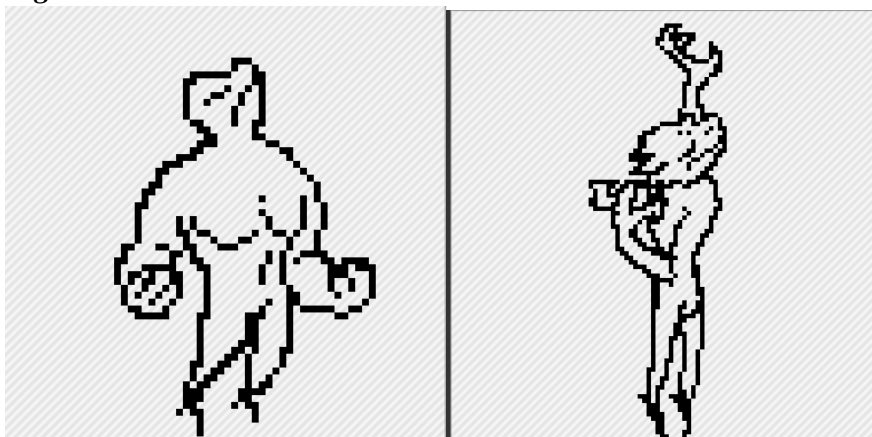
Fonte: Desenvolvido pelo autor

3.2.2.1.6 Dash

Já o *dash*, o momento de ataque do personagem, conta com uma animação de um frame. Enquanto o personagem se move pelo ar no *dash*, foi escolhida a utilização de uma animação de um frame pela velocidade na qual o personagem se encontra. Não há necessidade de mostrar mais informação, além de existir um efeito de *tracing* e *frames* que vão desaparecendo ao longo do caminho do personagem.

A pose então escolhida para os personagens também busca mostrar um pouco da personalidade de cada, devido à falta de uma animação propriamente dita. Cada personagem deve ter o máximo de informação no mínimo de tempo possível, logo cada personagem teve uma pose diferente escolhida.

Figura 27 – Frames do dash



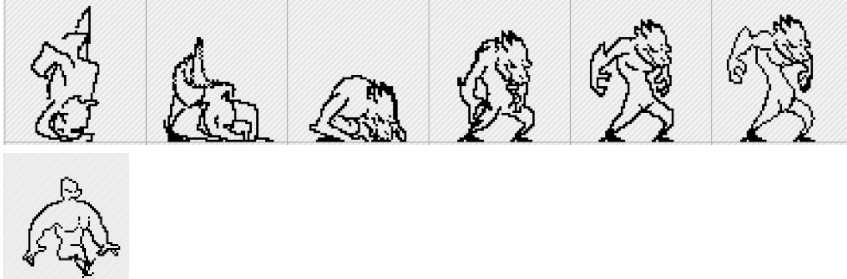
Fonte: Desenvolvido pelo autor

3.2.2.1.7 Colisões

Diversas animações de colisão foram produzidas, devido ao número de possíveis colisões que um personagem pode ter. As colisões também são animações rápidas, que tentam não diminuir o ritmo da ação acontecendo, já que o jogador fica imóvel durante tais momentos.

As colisões são tanto do personagem com o cenário, variando entre seus possíveis estados como de colisão entre os próprios personagens. A colisão entre os personagens também é uma animação de um frame, pelos mesmos motivos da animação anterior, já que acontecem devido às colisões entre dashes.

Figura 28 - Animações de colisão



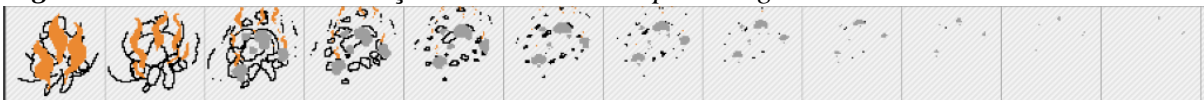
Fonte: Desenvolvido pelo autor

3.2.2.1.8 Morte

As animações de eliminação dos personagens foram pensadas inicialmente com base nos elementos que cada personagem ia representar, porém, conforme a ideia dos personagens elementais foi sendo posta de lado, as animações tomaram um rumo próprio, buscando explicitar o uso de animação de efeitos.

As animações de morte mostram o corpo do personagem sendo eliminado, deixando para trás somente o rabo, que fica então vulnerável para uma real eliminação do personagem. As animações, portanto, não são rápidas como algumas anteriores, pois neste momento pode-se utilizar de uma indicação maior para os jogadores de que algum jogador está vulnerável para uma eliminação.

Figura 29 – Frames da animação de morte de um personagem



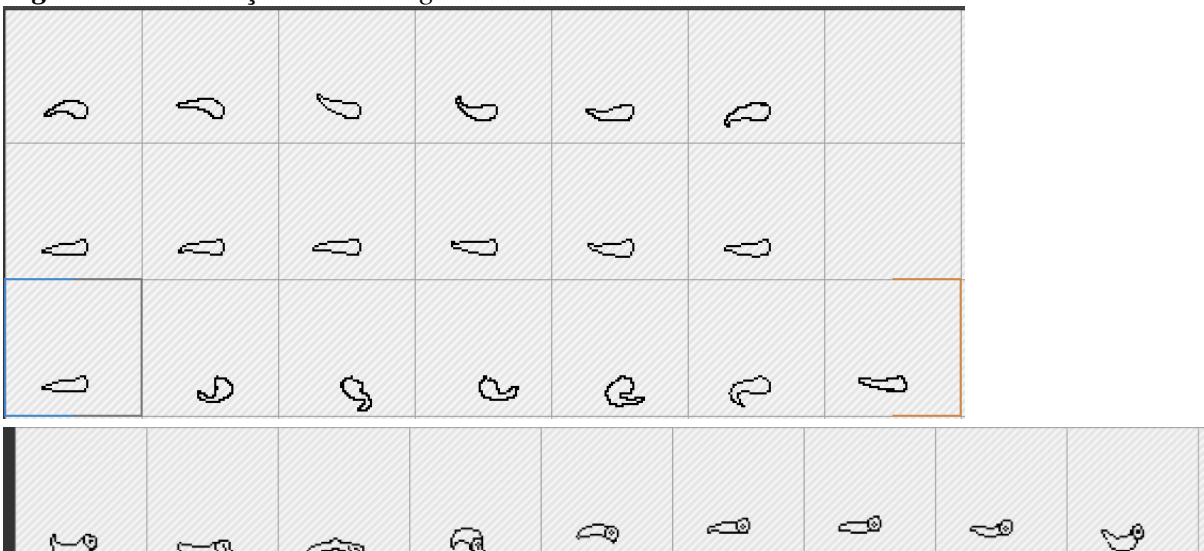
Fonte: Desenvolvido pelo autor

3.2.2.1.9 Rabo

O rabo dos personagens possui somente uma ação base, que é a de movimentação, porém, pode encontrar-se em alguns estados diferentes, que são de queda, movimentação, renascer e morte. Os estados diferentes possuem animações diferentes, que variam de loops até animações com fim determinado.

De maneira geral as animações também são simples, usando principalmente de movimentos curtos e rápidos. As animações utilizadas aqui têm o intuito de ser utilizadas por todos os personagens de maneira genérica, logo optou-se por movimentos mais neutros, sem muita definição de personalidade.

Figura 30 - Animações do rabo genéricas

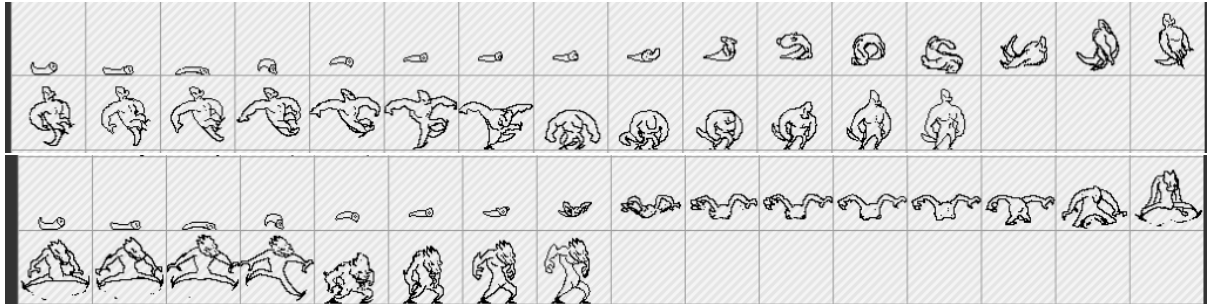


Fonte: Desenvolvido pelo autor

As animações de renascer do rabo são as animações que foram personalizadas para cada personagem, devido ao estado final da animação. Estas animações buscaram ter um *ease* maior

no final, assentando melhor a imagem de cada personagem ao final do processo. Os momentos de renascimento do personagem mostrarão o *sprite* piscando, como maneira de evidenciar um momento de invencibilidade do jogador.

Figura 31 - Animações de renascimento



Fonte: Desenvolvido pelo autor

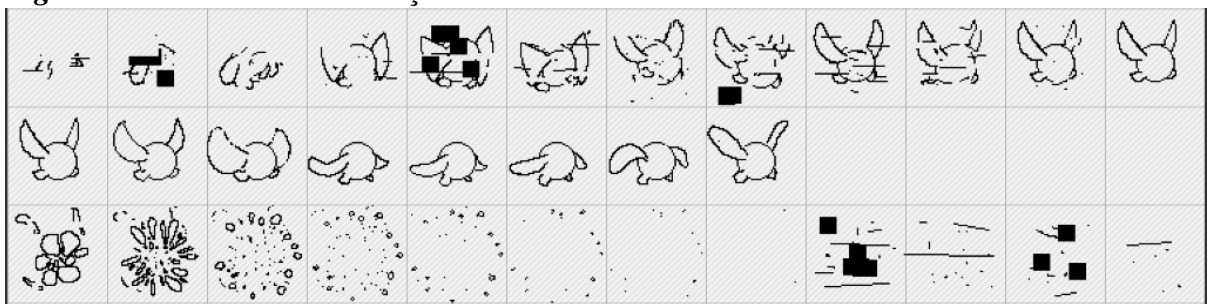
3.2.2.1.10 Itens

Os itens possuem sprites que são customizáveis baseados nas cores do cenário no qual estão presentes. Três estados possíveis para os itens são possíveis, sendo eles o de movimentação, de *spawn* e de morte.

A animação de movimentação é em *loop*, mostrando o item voando. A movimentação das asas tem um *ease* mais forçado para ambas as extremidades do movimento, com os *frames* intermediários sendo de maior velocidade.

As outras duas animações não são em *loop*, devido à troca de estado ao fim das mesmas. A utilização de efeitos de *glitch* ocorre nas animações, como forma de mostrar a estética *synthwave* do jogo, unificando o estilo.

Figura 32 – Frames das animações dos itens



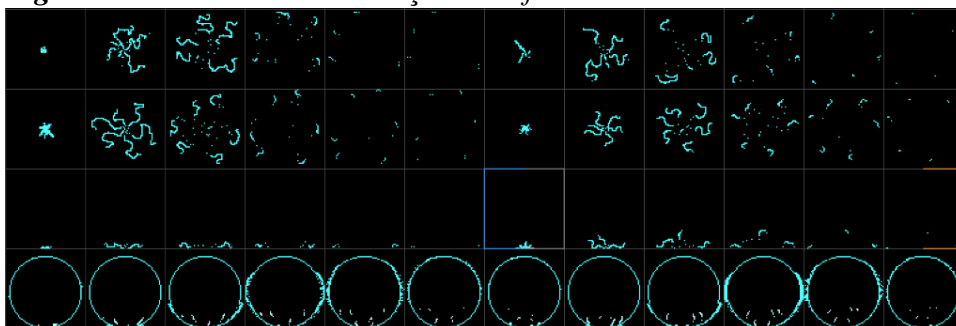
Fonte: Desenvolvido pelo autor

3.2.2.2 Efeitos

As animações de efeitos foram produzidas como uma forma de acentuar as colisões possíveis do personagem, dando um impacto maior nas mesmas. Tendo em vista a estética futurista, foi escolhido para as colisões impactos de eletricidade, dando a impressão do personagem estar se movendo em um ambiente cibernético.

Estas animações, de maneira geral, possuem movimentos rápidos inicialmente, com uma grande desaceleração no decorrer de seu trajeto. Essa fórmula básica é aplicada em todas as animações de efeito mostradas a seguir.

Figura 33 – Frames das animações de efeitos



Fonte: Desenvolvido pelo autor

3.3 SONORIZAÇÃO

3.3.1 Escolha dos programas e pesquisa de referências

O programa utilizado na criação das músicas é o *FL Studio*, sendo um programa frequentemente usado na área de criação de batidas e músicas. O programa possui muitas ferramentas que facilitam a criação de tons e ritmos, além de uma quantidade ampla de tutoriais espalhados pela internet.

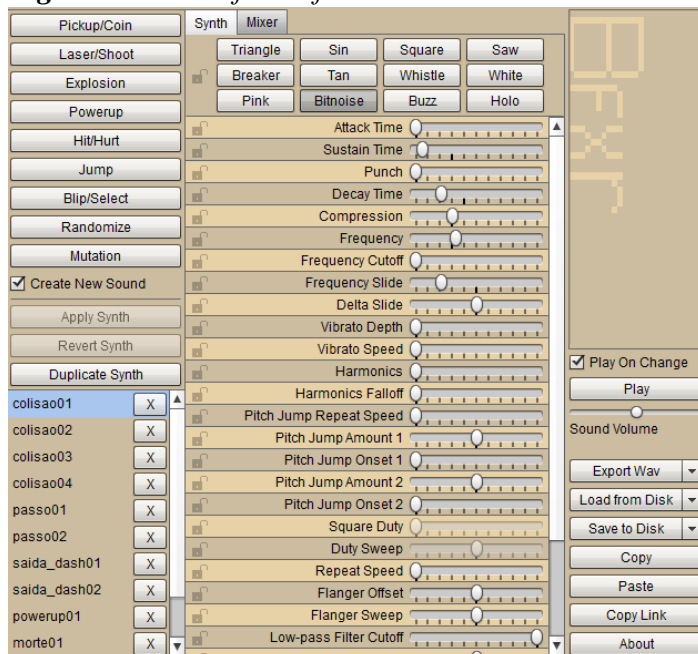
Figura 34 – Interface FL Studio



Fonte: Screenshot tirado pelo autor

O outro programa usado é o *Bfxr*, um programa leve e simples para a criação de efeitos sonoros. O programa conta com uma gama de parâmetros editáveis para a criação de diversos sons diferentes que remetem à época dos jogos de *SNES*. Além da leveza, o software é simples e intuitivo, permitindo uma prototipação de sons rápida.

Figura 35 – Interface Bfxr



Fonte: Screenshot tirado pelo autor

Assim como na parte visual, os sons do jogo têm como base a estética *synthwave*. O estilo tem como base o uso de sintetizadores antigos, e uso de reverberação na grande maioria dos instrumentos. Em geral, as músicas são instrumentais, e apesar de emularem instrumentos analógicos dos anos 80, algumas técnicas de produção modernas são utilizadas amplamente nas composições. Alguns exemplos relevantes de jogos que se utilizam deste tipo de sonoridade são *F-Zero*, *Hotline Miami* e *Far Cry 3: Blood Dragon*.

Figura 36 - F-Zero, Hotline Miami e Far Cry 3: Blood Dragon



Fonte: Montagem desenvolvida pelo autor

3.3.2 Trilha Sonora

As faixas para a trilha sonora foram desenvolvidas no programa *FL Studio*, sendo o programa tanto para criação como edição das mesmas.

Inicialmente, três músicas foram propostas, baseadas nos cenários os quais foram desenvolvidos para o protótipo. Conforme o projeto foi sendo produzido, viu-se que haveria um ganho na atmosfera do jogador caso mais músicas fossem produzidas, tendo, finalmente, seis músicas produzidas.

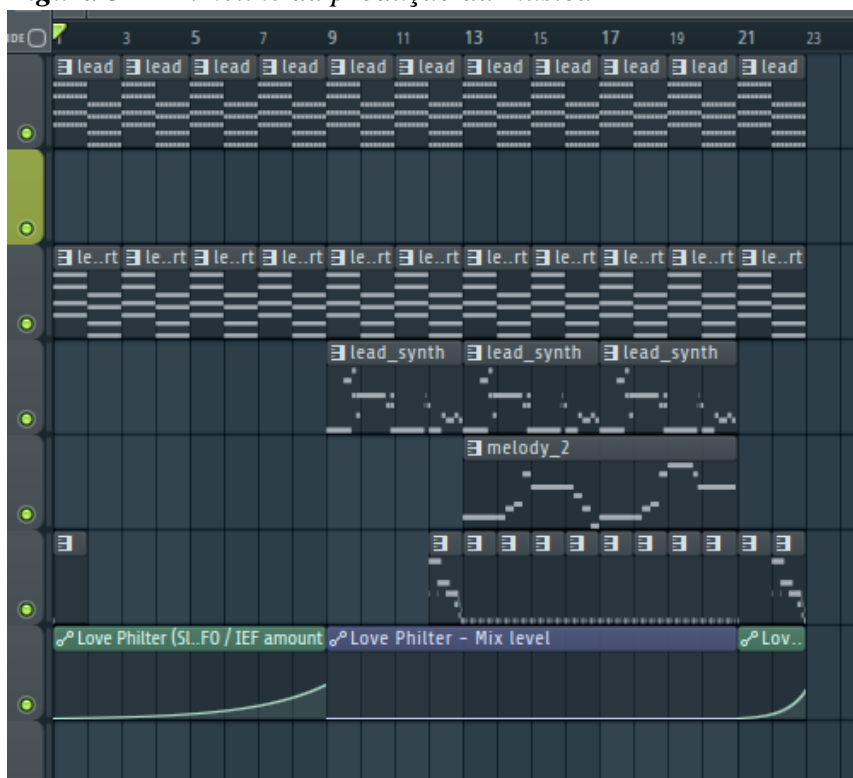
As músicas, assim como as animações, funcionam em *loops*, devido à natureza randômica das partidas de jogador contra jogador. Por vezes, as partidas levavam mais tempo que o proposto na música, logo, a natureza repetível das faixas foi essencial.

3.3.2.1 Faixa 1

A primeira música composta para o projeto tem como objetivo ser a faixa do menu principal do jogo. Levando isto em consideração, tal música deve explicitar a temática do jogo, bem como a velocidade no qual o mesmo é jogado.

Os sintetizadores escolhidos para a confecção dessa música possuem os mesmos atributos gerais entre si. Todos simulam teclados prevalentes nos anos 80, com uma melodia que tenta trazer a ideia tanto de velocidade como de agressividade. As batidas tem uma função similar, apesar de não tomar tanta parte da melodia em si, mas ditam um ritmo acelerado da música, deixando o jogador com uma ideia geral do que o jogo busca ser.

Figura 37 – Timeline da produção da música



Fonte: Screenshot tirado pelo autor

Os números na parte superior da imagem, indicam o compasso no qual a batida está localizada. Cada um destes retângulos com algum tipo de identificação em suas respectivas partes superiores indica uma batida diferente, seja uma melodia ou simplesmente marcação de tempo com a bateria. Cada uma destas batidas pode ser posta na linha do tempo do programa, formando a melodia da música em questão.

3.3.2.2 Faixas 2 até 6

As outras músicas foram utilizadas como fundo para as partidas em si. Considerando isto, elas possuem características similares compartilhadas por todas. A sensação de velocidade foi o fator de união maior entre as faixas, cada uma contendo algum tipo de estereótipo dos anos 80, típico do estilo proposto.

Similar à faixa principal, a escolha de instrumentos, apesar de mais variada, possui a mesma base. Instrumentos que simulam sons dos anos 80, buscando reforçar a ideia do estilo proposto.

Figura 38 – Timelines de algumas faixas



Fonte: Montagem desenvolvida pelo autor

3.3.3 Efeitos Sonoros

Os efeitos sonoros produzidos no programa *Bfxr* são simples, porém efetivos tanto em passar a ideia dos movimentos nos quais estão sendo utilizados, bem como em emular o estilo de efeito sonoro utilizado pelos consoles dos anos 80 e começo dos anos 90.

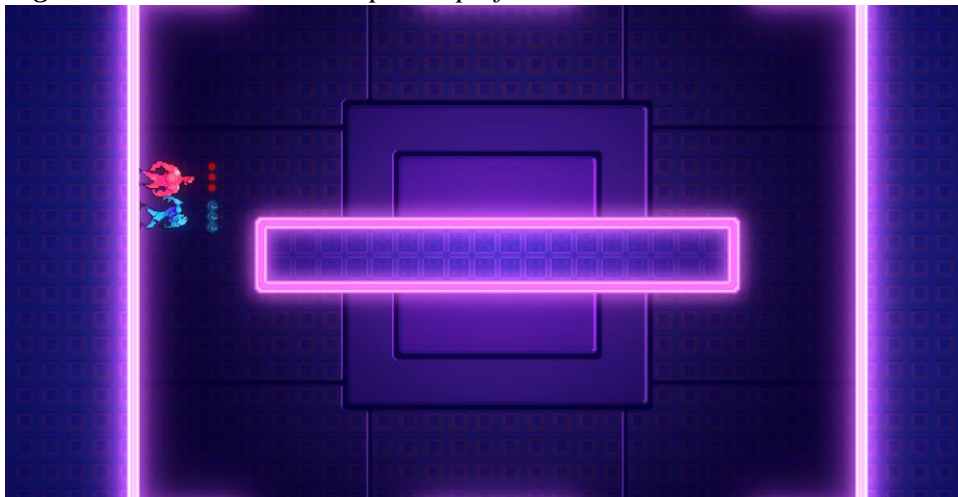
Tais sons são baseados tanto nas animações dos personagens quanto nas interações possíveis que o jogador pode ter com o jogo. Os sons podem ser tanto de mecânicas do jogo, como de um personagem chocando com outro ou interações de interface, como o som da seta de seleção mudando de opção.

A produção dos sons no programa é simples, tendo como suporte algumas bases já feitas pelos desenvolvedores do programa. Inicialmente se escolhe o tipo de som a ser feito, e então ajustes são realizados em cima deste som. Seja mudando o formato de onda do som, ou ajustando o *pitch*, a customização possível pelo programa é gigantesca.

3.4 PROTOTIPO FINALIZADO

Os cenários presentes no protótipo finalizado foram desenvolvidos por Augusto Kerber.

Figura 39 – Screenshots do protótipo finalizado



Fonte: Sreenshots tirados pelo autor

4 CONSIDERAÇÕES FINAIS

O desenvolvimento do presente estudo possibilitou uma análise de como funciona o processo de desenvolvimento de um protótipo de jogo digital 2D. Além disso, permitiu ao grupo obter dados mais consistentes, além de experiência sobre as etapas do processo, quais as partes que necessitam de mais atenção e até um entendimento sobre o grau de conhecimento que o profissional deve ter ao começar um projeto deste escopo.

Pensar no *game design* em conjunto com a animação auxilia o animador a preparar tanto a lista de animações a serem feitas, como as próprias animações, considerando todos os fatores possíveis de interação dos jogadores. Cada vez mais, os profissionais da área dos jogos precisam entender o funcionamento de todas as outras áreas envolvidas na produção. O jogo proposto foi finalizado, dentro das limitações estabelecidas pelo grupo, portanto pode-se concluir que o procedimento aqui sugerido é eficiente tanto para oferecer uma forma de organização na produção, quanto na sua aplicabilidade dentro de um ambiente com equipe reduzida e recursos limitados.

Algumas observações relativas ao projeto devem ser feitas, no entanto. A etapa do projeto que mais levou tempo foi o de *develop*. A equipe teve dificuldade em delimitar a temática a ser exposta no jogo, algo que deveria ter sido realizado antes. A definição da temática deveria ter sido resolvida na etapa de *refine*, permitindo uma prototipação mais rápida, tendo em vista as voltas para arrumar cenários e a estética dos personagens que ocorreram.

A falta de experiência prévia dos participantes do projeto também foi um grande fator impedor de progresso, sendo o motivo dos empecilhos demonstrados no parágrafo anterior. Apesar do entusiasmo inicial com relação à produção de um jogo digital como projeto de conclusão do curso, percebemos que subestimamos a quantidade de preparação necessária para uma produção de qualidade, considerando o tempo que tínhamos a nossa disposição. A experiência serviu para também ajudar a remediar este problema em projetos futuros a serem desenvolvidos potencialmente pela mesma equipe.

Outro fator que atrapalhou a velocidade de desenvolvimento do projeto foi a distância entre os integrantes do projeto e a falta de tempo para a programação. O programador que trabalhou no jogo vive em outra cidade, então a comunicação ficava limitada. O tempo disponível para trabalhar no projeto também foi limitado, devido a compromissos do mesmo, tornando o progresso do jogo mais lento.

A partir do trabalho aqui concluído e dos resultados obtidos, cabe também ressaltar os fatores que deram certo no projeto. Após os contratempos na etapa de *develop*, a segunda metade deste processo ocorreu da maneira esperada, sem nenhum contratempo adicional. As animações propostas foram concluídas, assim como o refinamento das mecânicas propostas inicialmente. O protótipo fechou dentro do escopo estipulado anteriormente pelos membros da equipe, ratificando a eficiência da metodologia escolhida para guiar o processo do projeto de conclusão.

Considerando que o procedimento em si sugerido aqui não demonstrou nenhum tipo de impedimento, uma possibilidade de continuação do jogo existe. Pode-se estender o desenvolvimento do projeto para a adição de mais personagens jogáveis, mais cenários, e mais *power-ups*, podendo futuramente ser lançado no mercado.

REFERÊNCIAS

CARTWRIGHT, Mariel. **Making Fluid and Powerful Animations For Skullgirls**. 2016. (21m05s) Disponível em:

<<https://www.youtube.com/watch?v=Mw0h9WmBlsw>> Acesso em: 10 out. 2019.

COOPER, Jonathan. **Game Anim: Video Game Animation Explained: A Complete Guide to Video Game Animation**. Estado, CRC Press, 2019.

CORON, Tammy. **Understand Disney's 12 principles of animation**. Estado, 17 de junho de 2019. Disponível em:

<<https://www.creativebloq.com/advice/understand-the-12-principles-of-animation>> Acesso em 10 out. 2019.

COSTIUC, Stanislav. **Farlands - What Is Good Game Design?** Estado, 25 de janeiro de 2018. Disponível em:

<https://www.gamasutra.com/blogs/StaniSlavCostiuc/20180125/313717/Farlands_What_Is_Good_Game_Design.php> Acesso em 10 out. 2019.

JONASSON, Martin. **Juice it or lose it**. 2012 (15m37s) Disponível em:

<<https://www.youtube.com/watch?v=Fy0aCDmgnxg>>

LICO, Richard. **Animation in Games: Function and Automation**. Estado, 25 de abril de 2017. Disponível em:

<<https://80.lv/articles/animation-in-games-function-and-automation/>> Acesso em 10 out. 2019.

LOVATO, Nathan. **How To Perfect Your Game's Core Loop**. Estado, 13 de julho de 2017. Disponível em:

<<https://gameanalytics.com/blog/how-to-perfect-your-games-core-loop.html>> Acesso em 10 out. 2019.

ROGERS, Scott. **Level Up: UM GUIA PARA O DESIGN DE GRANDES JOGOS**. São Paulo: Editora Edgard Blücher Ltda., 2013.

SCHELL, Jesse. **The Art of Game Design: A Book of Lenses**. Estado, CRC Press, 2008.

THOMAS, Frank; JOHNSTON, Ollie. **Disney Animation: The Illusion of Life**. Cidade, Disney Editions, 1995.

THORSON, Matt. **Road to the IGF: Matt Thorson's Towerfall: Ascension**. Estado, 24 de fevereiro de 2014. Disponível em:

<https://www.gamasutra.com/view/news/211431/Road_to_the_IGF_Matt_Thorsons_Towerfall_Ascension.php> Acesso em: 10 out. 2019.

WILLIAMS, Richard. **Manual de Animação**. São Paulo: Editora Senac São Paulo, 2016.

WOLSTEHOLME, Kevin. **What is a Core Loop in a Mobile Game?** Estado, 19 de julho de 2017. Disponível em:

<<https://risinghighacademy.com/what-is-a-core-loop-in-a-mobile-game/>> Acesso em 10 out. 2019.

