

UNIVERSIDADE FEDERAL DE VIÇOSA

Felipe Silva Rossi

FORGOTTEN FACTORY

Design e implementação de um jogo de plataforma/stealth

Viçosa, Minas Gerais
2019

Felipe Silva Rossi

FORGOTTEN FACTORY

Design e implementação de um jogo de plataforma/stealth

Trabalho de Conclusão de Curso apresentado à
Universidade Federal de Viçosa, como parte das
exigências do Programa de Graduação em
Ciência da Computação, para obtenção do título
de Bacharel

Orientador: Marcos Henrique Fonseca Ribeiro

Viçosa, Minas Gerais
2019

Resumo

Palavras-chaves: TCC, Game Design, Godot, UFV.

Design e Implementação de um jogo de plataforma / stealth para PC, realizado com a Godot engine, software open-source para desenvolvimento de games.

Abstract

Keywords: TCC, Game Design, Godot, UFV.

Design and implementation of a Platforming/Stealth game for PC , built with the Godot engine, an open-source game development software.

Sumário

Resumo	3
Abstract	4
1 - Introdução	6
2 - Referencial Teórico	7
2.1 - Tecnologias utilizadas	7
2.2 - Produtos Relacionados	7
2.3 - Conceitos Importantes de Design	7
3 - Forgotten Factory	8
3.1 - Estrutura do Design	8
3.1.1 - Design da GUI	8
3.1.2 - Design da Fase	8
3.1.2.1 - Sistema de Alarme	8
3.1.2.2 - Espinhos	9
3.1.2.3 - Câmera	9
3.1.2.4 - Cabine	9
3.1.2.5 - Barras	10
3.1.2.6 - Sirene	10
3.1.3 - Design dos Inimigos	10
3.1.3.1 - Inimigo 1	11
3.1.3.2 - Inimigo 2	11
3.1.3.3 - Inimigo 3	11
3.1.3.4 - Inimigo 4	12
3.1.3.5 - Chefe	12
3.1.4 - Design do personagem jogável	13
4 - Protótipo Atual	14
5 - Considerações Finais	17
Referências	18

1 - Introdução

Os jogos de plataforma/ação da franquia megaman e seus vários spin-offs, ofereceram, por bastante tempo, um tipo de experiência com jogabilidade única que cativou uma base fiel de fãs, com expoentes de vendas como Megaman 2 - 1.51 milhões de cópias , Megaman 3 - 1.08 milhões de cópias e Megaman X - 1.16 milhões de cópias[1].

Entretanto, com o passar dos anos, as vendas estagnaram e a empresa responsável, Capcom, silenciou-se por muito tempo quanto a continuacões futuras, mantendo a franquia em hiato desde 2010, salvo recentes compilados de jogos antigos e um jogo novo.

O nicho de fãs fiéis, entretanto, ainda existe e está sedento por jogos similares, como evidenciado pelo valor impressionante arrecadado, \$3,845,170, pela campanha de financiamento coletivo de um sucessor espiritual desenvolvido por uma das figuras chave por trás da criação da franquia[2].

Com o nicho disponível então, propõe-se o desenvolvimento de um jogo que possa saciar o desejo de fãs do gênero, além de contar com elementos de stealth para garantir personalidade própria. Para que a implementação se adeque ao escopo de produção de uma equipe de uma pessoa só, trabalhando durante um ano, será construído um “corte vertical” do jogo, consistindo de uma fase com todas as mecânicas relacionadas à movimentação do personagem, sua interação com inimigos e cenário, subsistemas de stealth e um Chefe como obstáculo final. A idéia é apresentar os conceitos que trazem o diferencial ao jogo, ao mesmo tempo que permita o desenvolvimento em tempo hábil.

A fim de poupar tempo de criação de sistemas para base do jogo, como renderização, input, colisão, dentre outros, optou-se por utilizar como base a Godot Engine, software open-source para criação de games. Deste modo, mais tempo pode ser dedicado à criação de mecânicas, produção de conteúdo e balanceamento geral do jogo, elementos mais importantes para a identidade do mesmo.

2 - Referencial Teórico

2.1 - Tecnologias utilizadas

A principal tecnologia utilizada será a Godot engine, motor para desenvolvimento de jogos open-source com uma linguagem de script própria com sintaxe similar a python. Atualmente na versão 3.x, disponibiliza ferramentas para organizar e facilitar a implementação dos diversos sistemas necessários para o jogo, lidando desde coisas básicas, como renderização, controle de áudio e leitura de inputs, até coisas mais específicas, como física e colisões 2D . A documentação da mesma pode ser encontrada em[3].

2.2 - Produtos Relacionados

Como dito anteriormente, a proposta é implementar um produto nos moldes da série de plataforma/ação Megaman, portanto, a análise do design dos jogos da série é de extrema importância. Foco maior será dado à série de jogos “Megaman Zero” para o portátil Game Boy Advance, devido à maior similaridade dos jogos em relação à visão criativa do produto a ser feito. Para os elementos de stealth produtos relevantes são jogos de plataforma/stealth como “Way of the Ninja” e a série “Assassin’s Creed Chronicles”, ambos disponíveis para PC.

2.3 - Conceitos Importantes de Design

O maior desafio em relação ao design é o balanço entre os conceitos de jogo de plataforma com ação rápida, inerente aos jogos da série Megaman, e os elementos de stealth que muitas vezes equivalem-se a uma aproximação deliberada, lenta. Se não bem pensados ambos os elementos do design podem contradizer-se, criando um jogo que não é envolvente ou divertido.

3 - Forgotten Factory

3.1 - Estrutura do Design

O jogador controla o personagem principal que interage com o mundo do jogo, composto pelo cenário 2D e inimigos com habilidades e comportamentos diferentes. O objetivo do jogo é avançar para a próxima fase, ultrapassando obstáculos do cenário e distraindo ou derrotando inimigos.

3.1.1 - Design da GUI

A camada de Graphic User Interface deve ser bem simples, exibindo apenas o essencial de informação para o jogador, ocupando o menor espaço da tela possível sem prejudicar a visibilidade da mesma. Existem apenas as barras de vida do personagem, sempre visível, e a da chefe, que aparece ao se aproximar da mesma.

3.1.2 - Design da Fase

A fase consiste em sua maior parte de blocos de tamanho mínimo pré-definido (Tiles) que o jogador tem que navegar. Além disso, existem alguns elementos interativos, que reagem a outros sistemas de gameplay, como o sistema de alarme, descrito a seguir.

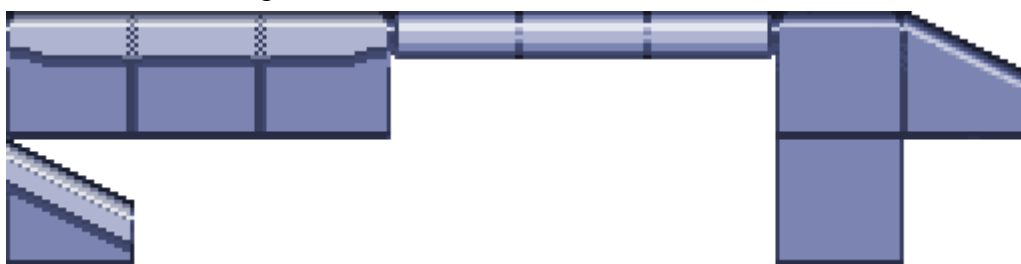


Figura 1: Tiles do cenário

3.1.2.1 - Sistema de Alarme

Ao ficar no campo de visão de certos elementos da fase ou inimigos por um certo período de tempo, o jogador dispara um alarme, que causa diversos efeitos adversos, dificultando o jogo. Para se desativar o sistema de alarme, é necessário que o jogador desabilite um elemento específico da fase, a sirene, atingindo-o com ataques.

3.1.2.2 - Espinhos

Elemento da fase que ao ser tocado mata o jogador instantaneamente. Tem o objetivo de limitar o espaço pelo qual o jogador pode se movimentar.

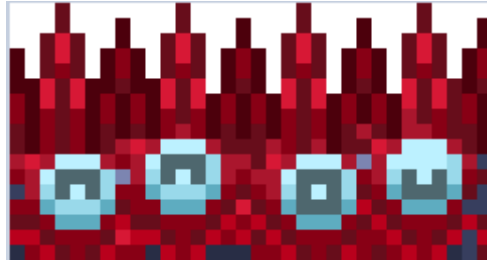


Figura 2: Espinhos

3.1.2.3 - Câmera

Elemento da fase com o propósito de desencorajar o jogador de estar em certas áreas. Tem um cone de visão de tamanho fixo, que rotaciona e para ao entrar em contato com o jogador.

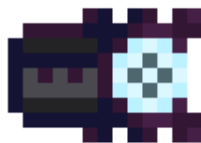


Figura 3: Câmera

3.1.2.4 - Cabine

Elemento da fase que reage ao estado de alarme. Quando em estado de alarme, a cabine se abre e surge de dentro dela um inimigo. Está dentre os elementos que aumentam a dificuldade ao se entrar no estado de alarme.

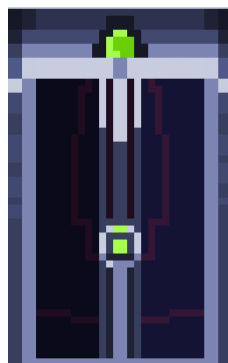


Figura 4: Cabine

3.1.2.5 - Barras

Elemento da fase que reage ao estado de alarme. Quando em estado de alarme, barras de metal se levantam e bloqueiam a passagem do jogador. Está dentre os elementos que aumentam a dificuldade ao se entrar no estado de alarme.

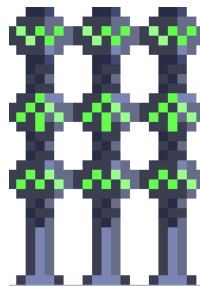


Figura 5: Barras

3.1.2.6 - Sirene

Elemento da fase que reage ao estado de alarme. Quando em estado de alarme, pode ser atacada pelo jogador, e após três ataques, desabilita o estado de alarme. É o elemento que possibilita ao jogador sair do estado de alarme, retornando o jogo ao seu nível de dificuldade habitual.

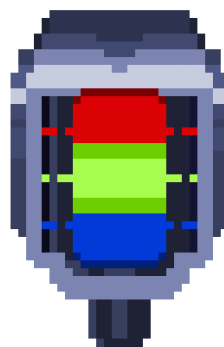


Figura 6: Sirene

3.1.3 - Design dos Inimigos

Os inimigos são os principais obstáculos que o jogador tem para progredir no jogo. Causam dano ao contato e possuem outros ataques e movimentações específicas que visam cumprir objetivos diferentes no design da fase. Para a fase descrita no projeto, existem quatro inimigos comuns e uma inimiga especial, a chefe

da fase. A chefe é o último desafio que o jogador enfrenta antes do final da fase e possui sua própria barra de vida, além de ataques mais perigosos e complexos.

3.1.3.1 - Inimigo 1

Inimigo básico, movimenta-se para frente e para trás num espaço pré-definido e quando o jogador entre em seu campo de atuação, para de se movimentar e começa a atirar projéteis na direção do jogador. Tem como objetivo familiarizar o jogador com o funcionamento dos inimigos em geral, além de servir como “coringa” quando se precisa aumentar levemente a dificuldade de alguma área.



Figura 7: Inimigo 1

3.1.3.2 - Inimigo 2

Movimenta-se para frente e para trás num espaço pré-definido e quando o jogador entre em seu campo de atuação acelera rapidamente na direção do mesmo. Tem como objetivo aplicar pressão e testar os reflexos do jogador em espaços horizontais abertos.



Figura 8: Inimigo 2

3.1.3.3 - Inimigo 3

Movimenta-se para frente e para trás num espaço pré-definido e quando o jogador entre em seu campo de atuação começa a circular o mesmo enquanto o

segue, ao mesmo tempo que atira projéteis em sua direção. Tem como objetivo aplicar pressão constante ao jogador, que é incentivado ao derrotá-lo o mais rápido possível, ao mesmo tempo que divide sua atenção com o resto dos obstáculos da fase.



Figura 9: Inimigo 3

3.1.3.4 - Inimigo 4

Movimenta-se para frente e para trás num espaço pré-definido e quando o jogador entre em seu campo de atuação, para de se movimentar e começa a atirar projéteis na direção do jogador. É diferenciado do Inimigo 1 por sua resistência maior a ataques do jogador, além de atirar projéteis mais rápido. É o inimigo que surge de dentro das cabines quando é ativado o estado de alarme. Tem como objetivo aumentar a dificuldade ao se atingir um estado de alarme.



Figura 10: Inimigo 4

3.1.3.5 - Chefe

Inimigo especial, mais complexo, com vida maior e maior número de ataques. Movimenta-se constantemente na direção do jogador. Se o jogador estiver a uma certa distância do inimigo, o mesmo para de se locomover por um certo tempo e gera um inimigo 4 para atacar o jogador. Se o jogador estiver mais próximo que a

distância anterior o inimigo movimenta-se rapidamente na direção do jogador e desfere um ataque, se o jogador estiver na área de efeito do ataque, recebe dano. Tem como objetivo ser um teste final para o jogador, exigindo o máximo de seus reflexos. Ao receber dano, sua barra de energia diminui, e ao atingir 0 o chefe é derrotado, e o jogo termina.



Figura 11: Chefe

3.1.4 - Design do personagem jogável

O personagem principal deve ter movimentação responsiva e rápida, além de possuir maneiras suficientes de interagir com o cenário e os inimigos. Ao colidir com um inimigo ou um projétil, recebe dano, que diminui o valor de sua barra de energia. Quando a energia chega a 0, o personagem principal é derrotado e o jogo termina.



Figura 12: Personagem jogável

4 - Protótipo Atual

A engine utilizada, Godot, funciona com base em uma estrutura de dados, o Node. Estes nodes podem ser hierarquizados, e seu código pode ser estendido para realizar as funções necessárias para a implementação. A base do jogo é, então, um node raiz, do qual todos os outros elementos são filhos. Executar este node é o equivalente a se executar o jogo.

A fase se resume a um conjunto de nodes para a User Interface, um conjunto de nodes que representa o personagem, um conjunto de nodes para representar o cenário e vários conjuntos de nodes que representam os inimigos.

O cenário em si é implementado por um node de tilemap, que contém toda a informação de imagem e colisão de cada parte, organizadas em quadrados de 32x32 pixels. Além disso existem outros conjuntos de nodes que implementam elementos interativos do cenário, sejam câmeras, espinhos ou outros elementos relacionados ao sistema de alarme.

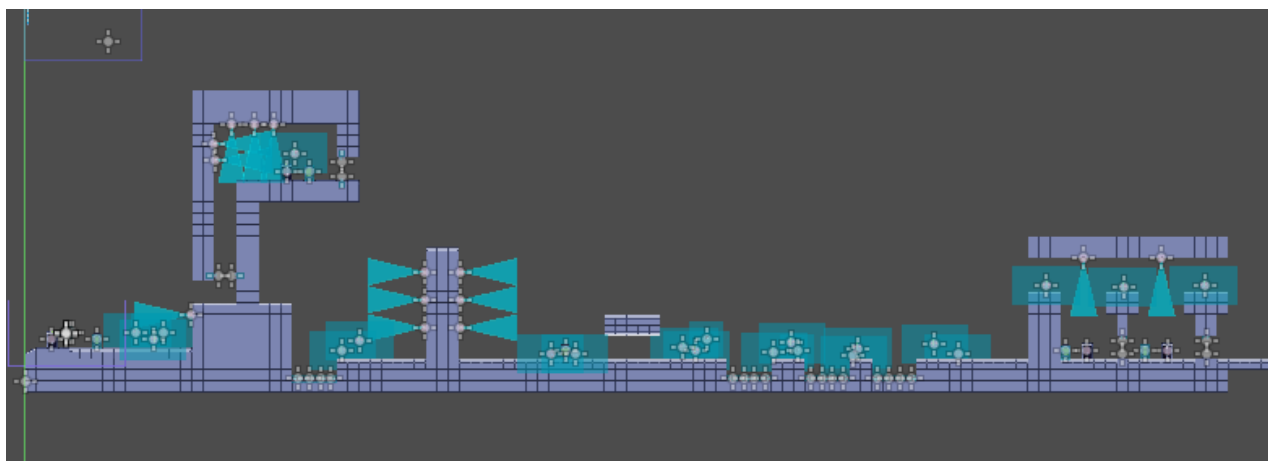


Figura 13 : Estrutura da fase Implementada

A camada de User Interface está implementada contando com uma barra de vida para o personagem, que diminui ao colidir com inimigos, assim como uma barra de vida para a chefe final da fase, que surge ao se aproximar da mesma. Ambas são compostas de nodes de GUI, que se adaptam a resoluções diferentes de tela e sempre ocupam o mesmo espaço, além de sprites vinculados por script aos valores de energia do jogador e do chefe.

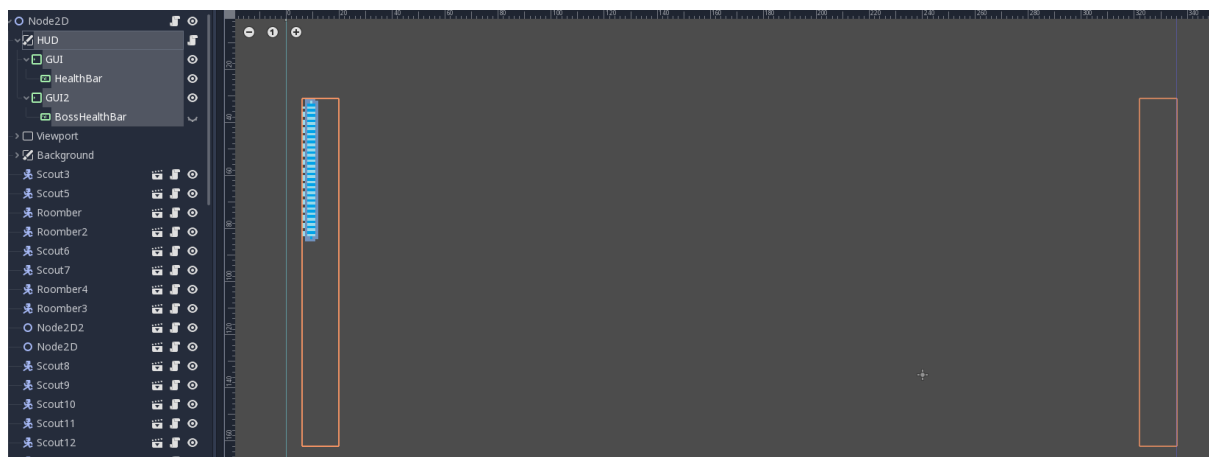


Figura 14 :Camada de GUI

O protótipo finalizado conta com uma implementação completa do personagem jogável, que pode se locomover pelo cenário por meio de movimentação normal, pulo, escalada de parede e movimentação rápida, além de possuir ataques corpo-a-corpo e a distância e se defender de inimigos.

O personagem é implementado como um conjunto de nodes, consistindo de um node de KinematicBody2D, que lida com as colisões com o cenário, um node de Sprite, que exibe os sprites do personagem, alguns nodes AnimationPlayer, que lidam com as animações dos sprites e alguns nodes Area2D de Hurtbox, Hitbox e Parry, que delimitam áreas que interagem com outros elementos do jogo, seja para detecção de ataque, contato com outros inimigos e etc. Além dos nodes, a programação do personagem está contida em um script principal e vários estados. A estrutura é uma máquina de estados finitos, cada ação que o personagem pode tomar é definida como um estado diferente, e os inputs do jogador e interação com outros elementos do jogo definem as transições entre os mesmos.

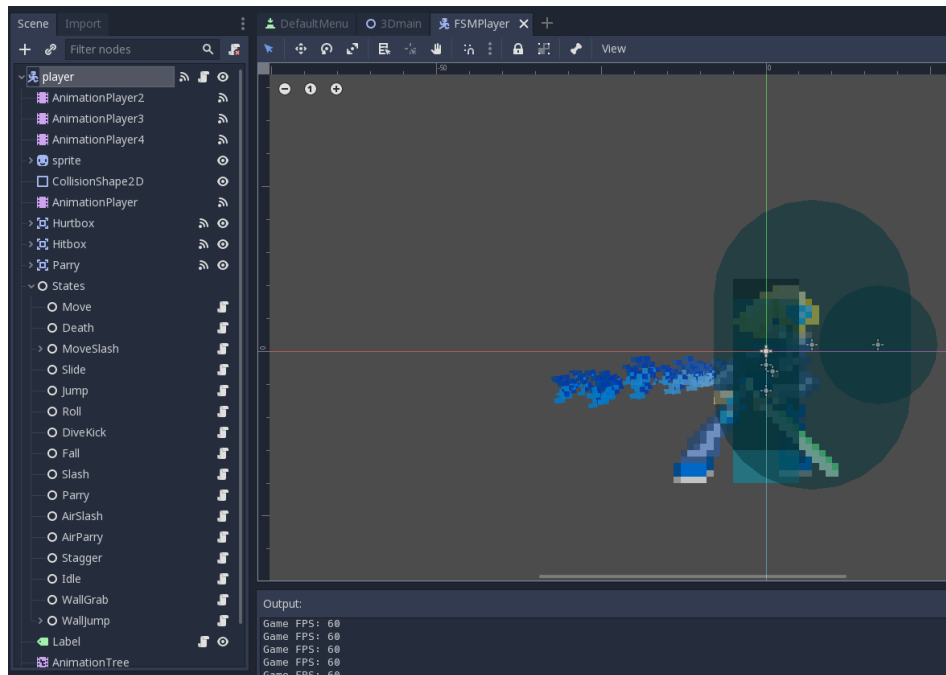


Figura 15 :Personagem implementado

Os inimigos de maneira geral, foram implementados de maneira similar ao personagem principal, com nodes de Sprite, AnimationPlayer e Area2D desempenhando funções análogas às do mesmo. Os scripts que definem seus comportamentos diferem, com inimigos mais básicos sendo dotados de apenas um script simples, e inimigos mais complexos como o Chefe utilizando uma máquina de estados finitos semelhante à do personagem jogável. A maior diferença entre as implementações se dá pelo fato que os inimigos reagem às ações do personagem jogável, enquanto que o próprio reage aos comandos do jogador.

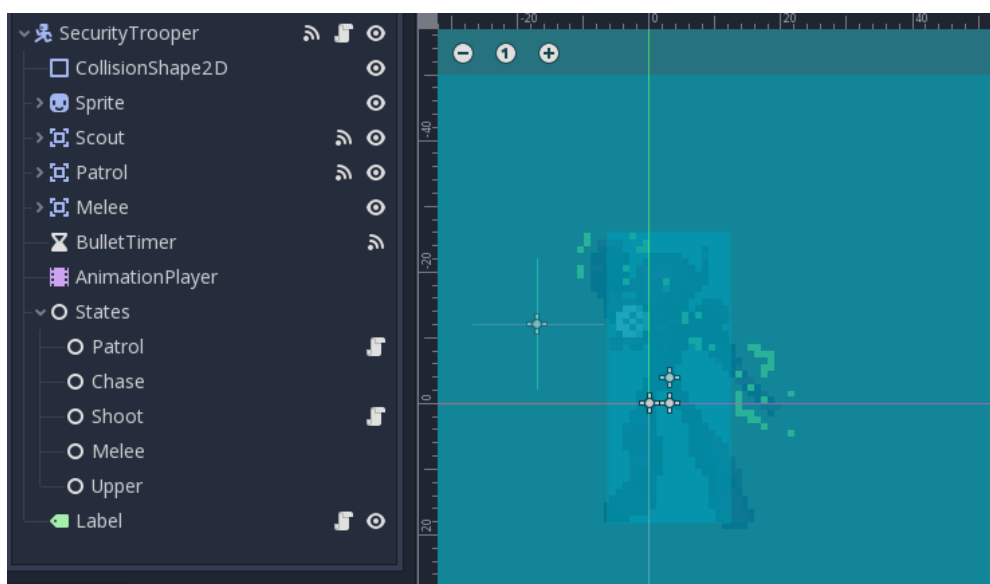


Figura 16 :Inimigo 4 implementado

5 - Considerações Finais

Ao final do projeto, foi possível implementar todas as funções planejadas, seguindo pela maior parte do tempo um cronograma rígido de produção de conteúdo. Durante o desenvolvimento do projeto, o design evoluiu para dar maior ênfase aos elementos de ação, com os elementos de stealth se simplificando ao sistema de alarme. Este processo se deu em parte pela interação às vezes conflitante entre os elementos de ação e stealth, com os sistemas naturalmente se lapidando tendo como fundamentos os objetivos definidos no início do projeto.

Em suma, o processo de design e implementação cumpriu os objetivos propostos, somando experiência e conhecimento sobre várias partes dos processos que fazem parte do design de video games.

Para trabalhos futuros seria interessante lapidar um pouco mais as mecânicas e elementos estabelecidos, e logo em seguida iniciar playtests com o público alvo. Após uma rodada inicial de playtests e correções, poderia-se então prosseguir para a expansão do conceito para um jogo de tamanho comercial, com diversas fases com temas diferentes, e produção constante de conteúdo seguindo um cronograma para uma deadline de release definida. A imagem a seguir demonstra alguns conceitos de personagens prontos para serem desenvolvidos junto a conceitos de possíveis fases futuras.

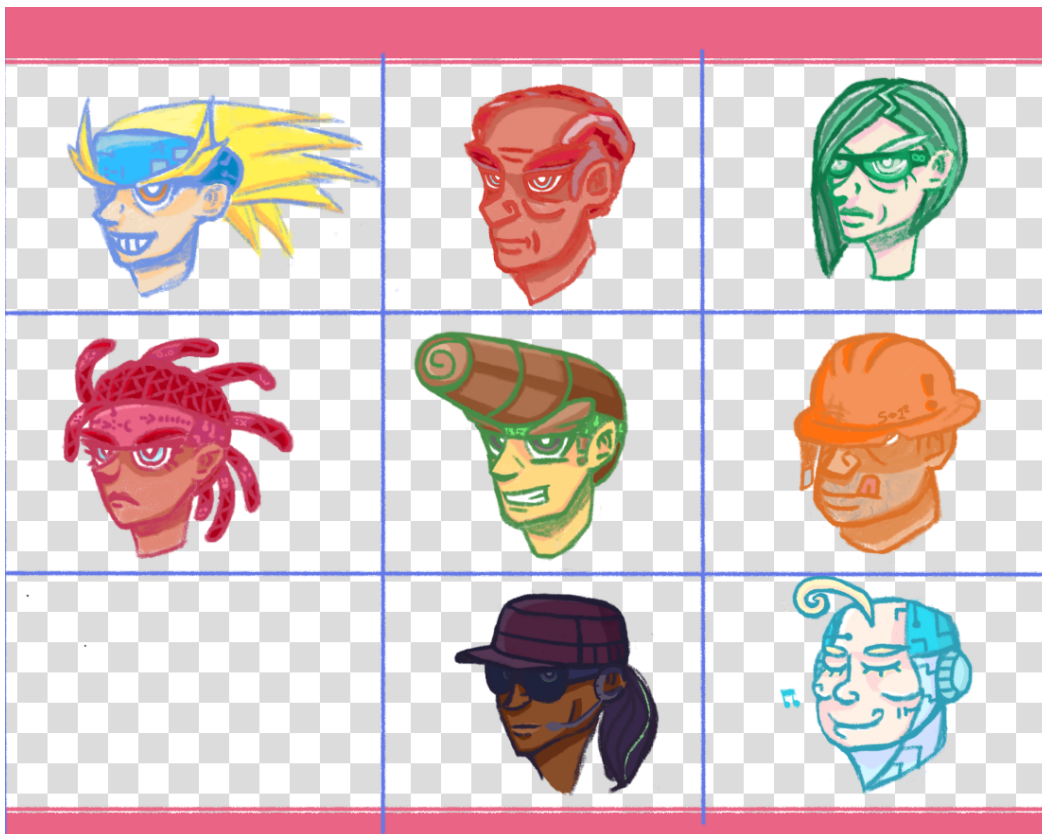


Figura 17 :Conceitos de personagens para fases futuras

Referências

[1]CAPCOM.Platinum Titles, 2019. Disponível em:
<<http://www.capcom.co.jp/ir/english/finance/million.html>>. Acesso em: 11/12/2019.

[2]COMCEPT USA. Mighty No. 9,2017.Disponível
em:<<https://www.kickstarter.com/projects/mightyno9/mighty-no-9>> Acesso em:
11/12/2019.

[3]GODOT. Godot Docs - 3.1,2014-2019. Disponível
em:<<http://docs.godotengine.org/en/3.1/>> . Acesso em: 11/12/2019.

[4]NYSTROM, R. Game Programming Patterns: 1. ed. Genever Benning, 2014