INTRODUÇÃO À ENGENHARIA - ENG1000

**Canos VS. Flappy Bird**

Relatório Final

**Autores:**

Andressa Muniz (1412959)

Júlia Aleixo (1411397)

Rafael Sanfins (1411141)

Rodrigo Leite (1413150)

Rio de Janeiro,

Maio de 2014

**RESUMO**

O projeto desenvolvido visa à criação de um jogo de estratégia em que o jogador é o Mario e utiliza seus canos para impedir que os Flappy Birds raptem a princesa Peach.

Para ele ser desenvolvido com sucesso, foi necessário a criação e organização da arte do jogo, a implementação de todos esses personagens e elementos no código, programar as funções que permitem que os inimigos e o Mario e o Luigi se movam e o resultado da colisão dos elementos do jogo.

O jogo foi desenvolvido com sucesso, permitindo que um ou dois jogadores desfrutem dele.

**ÍNDICE**

[1. INTRODUÇÃO 4](#_Toc388972301)

[2. METODOLOGIA 5](#_Toc388972302)

[3. IMPLEMENTAÇÃO 7](#_Toc388972303)

[3.1 Controle do jogador 7](#_Toc388972304)

[3.1.1 O controle de movimento 7](#_Toc388972305)

[3.1.2 Seleção de Canos 9](#_Toc388972306)

[3.2 Programar inimigos, Detecção de colisão e Sistema de pontuação 11](#_Toc388972307)

[3.3 Desenvolver mapas 12](#_Toc388972308)

[3.4 Criação da detecção de colisão 13](#_Toc388972309)

[3.5 Cutscenes 15](#_Toc388972310)

[4. RESULTADOS 17](#_Toc388972311)

[5. CONCLUSÕES 19](#_Toc388972312)

[BIBLIOGRAFIA 20](#_Toc388972313)

[ANEXO 1 – GAME DESIGN DOCUMENT 21](#_Toc388972314)

[ANEXO 2 – CÓDIGO FONTE 5](#_Toc388972315)

# 1. INTRODUÇÃO

Esse projeto baseia-se na criação e desenvolvimento de um jogo virtual. O jogo desse grupo chama-se “Canos VS. FlappyBird”, em que o objetivo principal é matar os FlappyBirds. Para isso, o jogador torna-se o personagem Mario, e mata-os colocando canos para obstruir seu caminho. O motivo pelo ataque desses pássaros são a captura da princesa Peach.

A programação será feita na linguagem lua e o interpretador será o love2d.

O resultado que se espera é um jogo sem *bugs* em que o jogador possa se divertir.

# 2. METODOLOGIA

Etapas:

1. Elaboração da ideia principal do jogo

O objetivo principal do jogo é matar os Flappy Birds, colocando canos na tela para bloquear seus caminhos. Para isso,o jogador torna-se o Mario.

2. Resolver os detalhes para criar o Game Design Document

Esse documento apresenta todas as informações básicas do jogo, assim como esboços da interface e mapas. Sua intenção era criar um resumo sobre os objetivos, gameplay e aspectos visuais facilitando, dessa forma, o desenvolvimento das demais etapas.

3. Selecionar/desenhar a arte dos personagens e a arte dos cenários:

Como o jogo se baseia em personagens e cenários que já existem, as suas imagens foram retiradas da internet. As que não estavam como idealizamos, modificamos nos programas Gimp e Adobe Photoshop.

4. Desenvolver o sistema de controle do jogador

Nessa etapa foi desenvolvido o controle sobre a movimentação do personagem do Mario e o gerenciamento de canos em tela.

5. Implementar inimigos

Desenvolve-se a diferenciação entre os inimigos por cores de velocidade e a geração aleatória de cores e posições.

6. Desenvolver sistema de mapas e fases

Será criado o sistema para carregar as imagens de cenários. Além disso, será “impressa” na tela a imagem do cenário.

7. Implementar a detecção de colisão

Será implementado um sistema que permite detectar quando o Flappy Bird atingiu o cano e se deve remover o cano ou matar o inimigo.

8. Desenvolver sistema de pontuação

O sistema de pontuação se baseia em moedas, que são adquiridos pelo jogador conforme ele mata os Flappy Birds. Essas moedas são utilizadas pelo jogador para mais tarde comprar mais canos. No inicio do jogo, são dados 50 moedas para o jogador utilizar.

Cada Flappy Bird morto tem um valor diferente, que é igual ao valor do cano da cor correspondente. A ordem de valor dos pássaros baseia-se na sua dificuldade (na ordem crescente de valor: verde,amarelo,azul,vermelho e preto).

9. Criar no jogo a opção de um jogo com dois jogadores

Nessa etapa, a mesma lógica que existe para o personagem do Mario será replicada para o Luigi.

10. Criar Cutscenes

Foi criada uma animação no inicio do jogo que explica a historia para o jogador. Essa pequena introdução foi implementada no código do jogo.

11. Melhorias no aspecto visual do jogo

O jogo e todas suas funções já estão funcionando e como planejado. Porém, o contador de moedas bem como o seletor de canos não está com um aspecto de terminado. Logo, serão editados com Photoshop para ser mais visualmente agradável.

12. Implementar o ranking online

# 3. IMPLEMENTAÇÃO

## 3.1 Controle do jogador

Para facilitar a programação essa etapa foi subdividida em duas etapas correlacionadas: controle do movimento do personagem e a seleção de cano. Dessa forma, foi possível dividir o grupo de trabalho em dois para programar seções que possuíam pouca dependência.

### 3.1.1 O controle de movimento

O controle da posição do personagem foi importante ao estabelecer e atualizar diversas variáveis que seriam usadas em etapas futura.  
O movimento do personagem é constituído de cinco etapas:

1. O jogador pressiona a tecla
2. Altera-se a imagem do personagem
3. Inicia-se a contagem de tempo
4. Em um tempo pré-estabelecido o personagem se move
5. Após outro intervalo de tempo o personagem recebe sua imagem novamente

Para isso foram definidas as variáveis para a movimentação para baixo representadas naTabela 1.1 e Diagrama 1.1.

|  |  |
| --- | --- |
| Variáveis | Aplicação |
| Mario.anim\_saida\_down | Verificada na função love.keypressed quando o jogador pressiona a tecla baixo, caso o jogador não esteja se movendo para fora das linhas do campo, torna-se verdadeira. Responsável por habilitar o movimento |
| Mario.tempo\_saida | Utilizada para o controle de tempo. A partir da confirmação do clique, realizado pela Mario.anim\_saida\_down, é incrementada para marcação do tempo. |
| Mario.animacao | Responsável por alterar na função love.draw a imagem que representa o Mario. |
| Mario.anim\_move | Visa a impedir a movimentação contínua do personagem, o jogador somente se move se seu valor for falso. Torna-se verdadeira no movimento, quando o personagem altera sua posição, e falsa ao fim da execução. |
| Mario.posy | Armazena a posição em***y*** do personagem. |
| Mario.Linha | Armazena a linha do mapa em que personagem está situado. |

Se o jogador pressionar a tecla para baixo a variável Mario.anim\_saida\_down torna-se verdadeira habilitando, assim, o movimento. Com essa variável verdadeira inicia a contagem de tempo por meio da variável Mario.tempo\_saida que executa o controle sobre as demais etapas e substitui a imagem em exibição alterando o valor de Mario.animacao.

Quando a variável a Mario.tempo\_saida atinge o valor estabelecido e como Mario.anim\_move é falso movimenta o Mario atualizando Mario.posy e Mario.Linha etorna a proposição falsa alterando o valor de Mario.anim\_move para verdadeira, assim, impedesse que o personagem ande indefinidamente.

Após outro período de tempo,marcado da mesma forma que anteriormente, retira a animação e termina a execução da seção de movimento retornando os valores inicias das variáveisMario.anim\_saida\_down eMario.animacao.

### 3.1.2 Seleção de Canos

A seleção de cano permite ao jogador escolher e colocar os diversos canos no mapa. Esse processo divide-se em escolher a cor do cano e por o cano no mapa.  
Para isso foi criada uma matriz nomeada de Matriz\_Cano que armazena a informação sobre que posição existe cano sua cor ou se está vazia. Essa variável será utilizada em outras etapas do projeto como na implantação da colisão.

|  |  |
| --- | --- |
| Variáveis e funções | Aplicação |
| Matriz\_Cano | Armazena a informação se há cano em cada posição do mapa e sua cor de acordo com a linha e coluna. |
| Mario.Cano\_Selecionado | Armazena a informação sobre que cano foi selecionado. É responsável por representar o cano que está selecionado. |
| Moedas | Responsável por controlar a quantidade dos canos na tela. É decrementada a cada cano colocado. Em uma etapa futura será habilitada a detecção de colisão que identificará quando o cano deve ser removido da tela, assim, incrementando a quantidade de moedas a ele referente. |
| Dificuldade.Nivel | Nessa etapa do projeto não existia sistema de atualização dessa variável. Entretanto seria modificada de acordo com a pontuação do jogador. Habilitaria a colocação de canos de cores diferentes e o surgimento, ainda não implementado,de flappybirds de outras cores. |
| Por\_Cano\_keypressed | Função que armazena os comandos para colocar os canos na tela |
|  |  |

Ao jogador pressionar a tecla de seleção de cano o jogo deverá alterá-lo por meio da variávelMario.Cano\_SelecionadoPara colocá-lo a tecla para por o cano deverá ser utilizada executando assim a função Por\_Cano\_keypressed que verifica se há possibilidade de colocar o cano, ou seja, na última coluna da linha desejada há uma posição livre, caso haja um espaço livre atribui a cada coluna o valor da coluna e na primeira o valor de Mario.Cano\_Selecionado.

## 3.2 Programar inimigos, Detecção de colisão e Sistema de pontuação

A criação dos FlappyBirds era importante para desenvolver as demais etapas, pois com ele seria possível criar o sistema de colisão e pontuação.

Assim, os inimigos são inicializados em um vetor que armazena a tabela sobre posição e velocidade de cada inimigoinimigos[i].Além disso, foi criadaa tabela Controle\_Inimigos que controla a velocidade de todos os pássaros, igualando a velocidade a expressão (Pontuacao+1)^(1/4),e o tempo de aparição, que a cada vez que gera um inimigo tem seu valor zerado.

Nessa etapa, também foi criada a função ChecarColisaoPassaro responsável por verificar se o pássaro colidiu com o cano, dessa forma, removendo o cano da Matriz\_Cano e incrementando a moedas referente a cor do pássaro. Além de incrementar a variável responsável pela Pontuacao.

Apesar de programar inimigos e o sistema de pontuação terem sido etapas consideradas separadamente na idealização do projeto foram unificadas durante a realização do projeto, uma vez que para programação estão conectadas. Além disso, uma pequena parte da colisão teve de ser feita com os inimigos.

Na implementação dos inimigos tivemos diversos contratempos:

O primeiro foi conseguir criar repetidas tabelas para armazenar os dados dos flappys, mas solucionamos criando o vetor inimigos[i] que armazena uma tabela em cada espaço.

O segundo foi nos livrarmos de um FlappyBird após ser destruído, então definimos uma condição de inexistência. Nisso também resolvemos as situações em que um pássaro, após ser removido, ainda tinha sua posição no eixo x (inimigos[i].posicaox) sendo atualizada de modo que interferia no posicionamento de novos canos devido sua informação na colisão, pois bastava configurar caso o inimigo não existisse sua posição era nula.

|  |  |
| --- | --- |
| Variáveis e funções | Aplicação |
| ChecarColisaoPassaro | Verifica a colisão do pássaro com o cano |
| Matriz\_Cano | Armazena a informação se há ou não cano em cada linha e coluna do mapa. Além de sua cor |
| Pontuacao | Armazena a pontuação do jogador e é incrementada a cada inimigo morto. |
| Controle\_Inimigos.velocidade | Controla a velocidade de todos os pássaros para que seja incrementada ao decorrer do jogo. Essa variável não controla diretamente a velocidade,embora exerça influência em todas independente da cor. |
| Controle\_Inimigos.tempo\_aparicao | Responsável por controlar o intervalo de surgimento dos inimigos. |
| Inimigos[i].linha | Armazena a linha do inimigo(posição em y) |
| Inimigos[i].posicaox | Armazena a posição em x do inimigo. |
| Inimigos[i].cores | Armazena a cor do inimigo. |

## 3.3 Desenvolver mapas

Para implantação do mapa foi usada à replicação de imagens por meio de programação.   
A imagem do piso é remetida inúmeras vezes criando a sensação ao jogador de um chão único.

O carregamento dos personagens e inimigos no dependeu das variáveis utilizadas anteriormente que determinavam sua posição. Embora os canos não tivessem suas coordenas armazenadas em uma variável bastou localizar a posição de cada linha e coluna e mandar desenhar em todas as posições todas as cores, porém com a condição de que essa cor deveria ser a armazenada naquela variável. Caso não houvesse nenhum cano não haveria igualdade, assim, não seria desenhado nenhum cano.

Apesar da programação para todos os mapas ser igual, devido a dificuldades na edição de imagens e temponão foi possívelcriar todos os cenários listados no (anexo1). Embora, o código para gera-los seja análogo ao empregado para criação do mapa em uso.

## 3.4 Criação da detecção de colisão

Como já havia se feito parte da detecção na etapa anterior, essa parte do desenvolvimento resumiu-se a criação do modo salve a princesa (ANEXO1).

|  |  |
| --- | --- |
| Variáveis e funções | Aplicação |
| Jogo | Controla em que modo o jogo se encontra |
| Princesa.Nivel | Armazena a quantidade de vezes que se acessou o modo salve a princesa |
| Princesa.PosX | Armazena a posição em x da princesa sendo resgatada |
| Princesa.PosY | Armazena a posição em y da princesa sendo resgatada |
| Princesa.Cor | Armazena a cor do FlappyBird que atingiu o final do mapa |
| Princesa.Contador | Contador do números de cliques para usar a bola de fogo que eliminará a princesa. Caso essa variável alcance o valor estabelecido sai do modo “Salve a Princesa” |

Se o pássaro atinge a última coluna então o jogo passará ao modo salve a princesa alterando a variável de controle do modo de jogo,Jogo, e reinicializa os inimigos os canos e as moedas do mapa.

A princesa começa a se mover tendo sua posição atualiza, ou seja, Princesa.PosXé alterada. Nesse período, o jogador deverá aperta o botão para incrementar Princesa.Contadorcaso atinja valor superior a 10\*Princesa.Nivelsai do modo “Salve a Princesa” e retorna ao jogo normal. Contudo, caso não atinja esse valor e a princesa chegue ao final da tela o jogo termina.

## 3.5 Cutscenes

O desenvolvimento da animação exigiu a criação de uma tabela para facilitar a organização das variáveis (Tabela).

|  |  |
| --- | --- |
| cutscenes.contador | Responsável por controlar a passagem de tempo |
| cutscenes.ativador | Inicia e finaliza a cutcene |
| cutscenes.bowser | Tabela que armazena as informações referente ao bowser |
| cutscenes.bowser.anim | Contém a informação sobre a imagem a ser desenhada |
| cutscenes.flappybird | Tabela para o primeiro FlappyBird |
| cutscenes.flappybird.contador | Controlador de tempo do tempo do primeiro FlappyBird |
| cutscenes.bandodeflappy | Tabela do bando que voa posteriormente a morte do primeiro pássaro |

A cutscenes é iniciada com o bowser andando até o centro da tela, ou seja, tendo sua posição atualizada por meio da variável cutscenes.bowser.posicaox. Posteriormente um cano é colocado na tela e o FlappyBird começa a se deslocar pela tela afim de atingir o cano. Ao chocar-se contra o cano rotacional e começa a cair atingindo chão morto. Surge, nesse momento um bando que atravessa a tela e a apresentação é finalizada.

Legenda:

Azul- condição

Vermelho- atribuição

## 3.6 Modo Coperativo

Para diferenciar o jogo sozinho de jogo cooperativo, foi definido que a variável jogo assume o valor 4 para jogo cooperativo, e 5 para salve a princesa cooperativo. Então bastou adicionar estruturas condicionais para habilitar os controles do segundo jogador (Luigi). Já a tela de escolha é ativada quando sua variável “Ativada” é igual a “true” e o jogo está no menu (Jogo == 0) , ou quando a tecla “enter” é pressionada na tela de “Game Over” (Jogo == 3)

# 

# 4. RESULTADOS

Para surpresa do grupo consegui-se realizar quase tudo prometido no GDD. Entretanto, o sistema de ranking online não foi implementado por falta de conhecimento do grupo de programação voltada para web. Além disso, não foi possível inserir a flor de gelo no jogo devido ao prazo do projeto. Os mapas do jogo ficaram sem o toque artístico característico da série Mario Bros.,e o jogo não esta em modo *fullscreen* pois não foi possível achar a fonte da deturpação das posições de elementos do jogo quando a tela cheia era ativada.

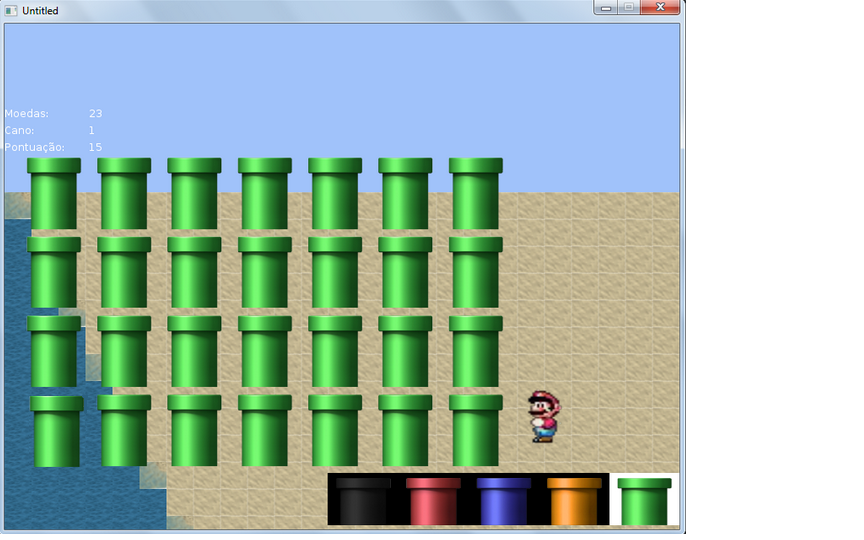


Figura 1- O máximo de canos permitidos na tela. No canto inferior direito encontra-se o seletor de canos, que mostra em branco a cor que o jogador escolheu.



Figura 2- Modo cooperativo; ambos Mario e Luigi se movimentam para cima e para baixo nas 4 fileiras. O seletor de canos da esquerda pertence ao jogador 2 (Luigi) e o da direita ao jogador 1( Mario).

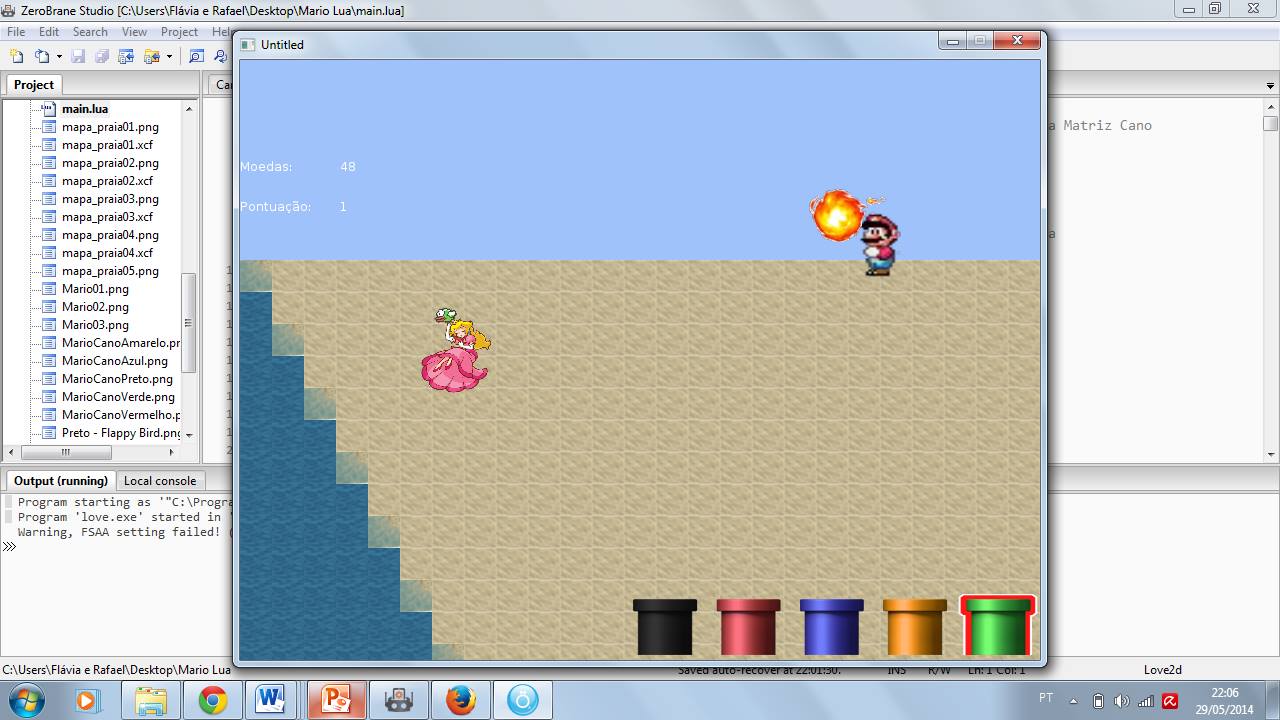


Figura 3- Modo salve a princesa: após deixar algum pássaro chegar à princesa o jogador tem mais uma chance de salvá-la, atirando bolas de fogo no Flappy Bird.

# 

# 5. CONCLUSÕES

# Durante o desenvolvimento do projeto foi perceptível dificuldades na programação em grupos separados e a incorporação desse código que demandavam um maior gasto de tempo. Eram criadas variáveis semelhantes exigindo, assim, um trabalho maior. Notou-se que um maior uso de funções poderia minimizar o problema, além da definição prévia das variáveis utilizadas. Porém, o principal objetivo do projeto era criar um jogo que provocasse entretenimento ao usuário foi alcançado. Apesar de visualmente o jogo não se tornou agradável e não foi possível entregar todos os cenários prometidos por dificuldades no tempo de edição. Além disso, o áudio também não foi introduzido no jogo. Apesar de a experiência do usuário ser prejudicada pela falta de alguns desses recursos, o jogo teve um resultado esperado na parte lógica no qual se apresentou nos últimos testes tem travamentos ou erros. Essa dificuldade e perda de tempo na incorporação de códigos feitos separadamente acarretou no atraso do projeto e da não realização de todos os cenários e do aúdio.

# Ao longo do curso aprendemos o básico de programação: a linguagem lua, como desenvolver códigos no *zbstudio* com interpretador love2d e o básico sobre jogos virtuais. Porém, o tempo que tivemos para aprender e programar foi muito curso, logo tiveram limitações no desenvolvimento do projeto. Caso houvesse outra oportunidade de fazermos esse mesmo projeto, teríamos nos organizado melhor antes da programação. Ou seja, os itens que melhoraram o aspecto visual do jogo já estariam prontos, e quando começasse a programação, eles só precisariam ser implementados no código. BIBLIOGRAFIA

1. Imagens: Google imagens
2. Duvidas no código: Love2d wiki
3. Tiles: http://opengameart.org/content/isometric-tiles

# ANEXO 1 – GAME DESIGN DOCUMENT



**Mario vs Flappy Bird**

Game Design Document

Versão: 2.0

**Autores:**

Andressa Muniz

Júlia Aleixo

Rafael Sanfins

Rodrigo Leite

Rio de Janeiro,

Março de 2014

Índice

[1.História 3](#__RefHeading__778_2127705544)

[2.Gameplay 4](#__RefHeading__780_2127705544)

[3.Personagens 5](#__RefHeading__782_2127705544)

[4.Controles 5](#__RefHeading__784_2127705544)

[5.Câmera 7](#__RefHeading__786_2127705544)

[6.Universo do Jogo 8](#__RefHeading__788_2127705544)

[7.Inimigos 9](#__RefHeading__790_2127705544)

[8.Interface 10](#__RefHeading__792_2127705544)

[9.Cutscenes 11](#__RefHeading__794_2127705544)

[10.Cronograma 11](#__RefHeading__796_2127705544)

1. **História**

Era uma vez, em um reino chamado Mushroom Kingdom, uma família de pássaros (espécie Flappy) que vivia feliz. Um dia, ao voltarem do seu passeio diário, aconteceu uma tragédia: um dos jovens da família morre ao bater em um cano. Os Flappy’s não descansam até encontrar o culpado pela implantação do cano em seu habitat, e quando descobrem que o cano era do Mario, buscam vingança.

Mario é avisado que os pássaros desejam raptar sua princesa e avisa seus amigos para tentar impedi-los. O que nenhum dos dois sabe é que foi Bowser que colocou aquele cano para incriminar Mario. E você vai ter que ajudar Mario, mais uma vez, a salvar a princesa Peach.

**2. Gameplay**

O jogo consiste em impedir que o Flappy Bird consiga chegar até a princesa. Para isso, o jogador poderá se movimentar na extremidade do mapa e colocar canos da mesma cor que os pássaros para matá-los. O cano escolhido entrará na 1ᴼ casa da fila e o que estava nela passará para 2ᴼ casa, e assim, sucessivamente.

A compra de canos depende da quantidade de moedas que o jogador tem. No inicio do jogo, ele é dado uma quantidade de moedas para colocar os primeiros canos. Conforme os Flappy Birds vão morrendo, as moedas vão aumentando. Cada pássaro tem um valor diferente, que corresponde ao preço do cano da mesma cor.

Quando um pássaro alcançar a extremidade da tela onde está o jogador, ao invés de derrota instantânea, ele começará a carregar a princesa para longe, então Mario terá a oportunidade de coletar uma flor de fogo. Então, pressionado a tecla do poder rapidamente, o jogador carregará a barra de energia para disparar a bola de fogo que matará o pássaro e trará o jogo de volta ao modo normal. Já a flor de gelo causa uma explosão que desacelera os inimigos em todo o mapa, mas só pode ser usada uma vez a cada 3 minutos (tempo a definir).

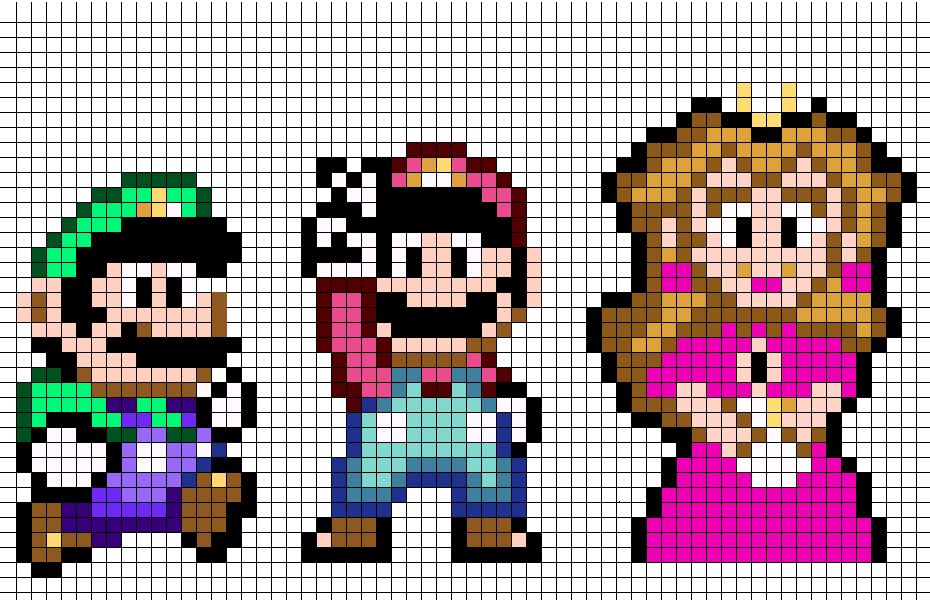
Os desafios durante o jogo permanecerão constantes, contudo durante a partida com o aumento do número de pontos, ganhos a cada inimigo derrotado, haverá o acréscimo de mais cores de pássaros/canos com um limite de cinco diferentes.Cada cor nova do pássaro que surgir terá uma velocidade maior do que a cor anterior(assim, a dificuldade aumenta durante o jogo).

Não há vitória, pois o jogo nunca acaba. A sensação de conquista está em ver sua pontuação superar a de outro num ranking online.

Já a derrota acontece quando um pássaro, após chegar ao final da tela, consegue levar a Princesa para o outro lado da tela.



**3. Personagens**

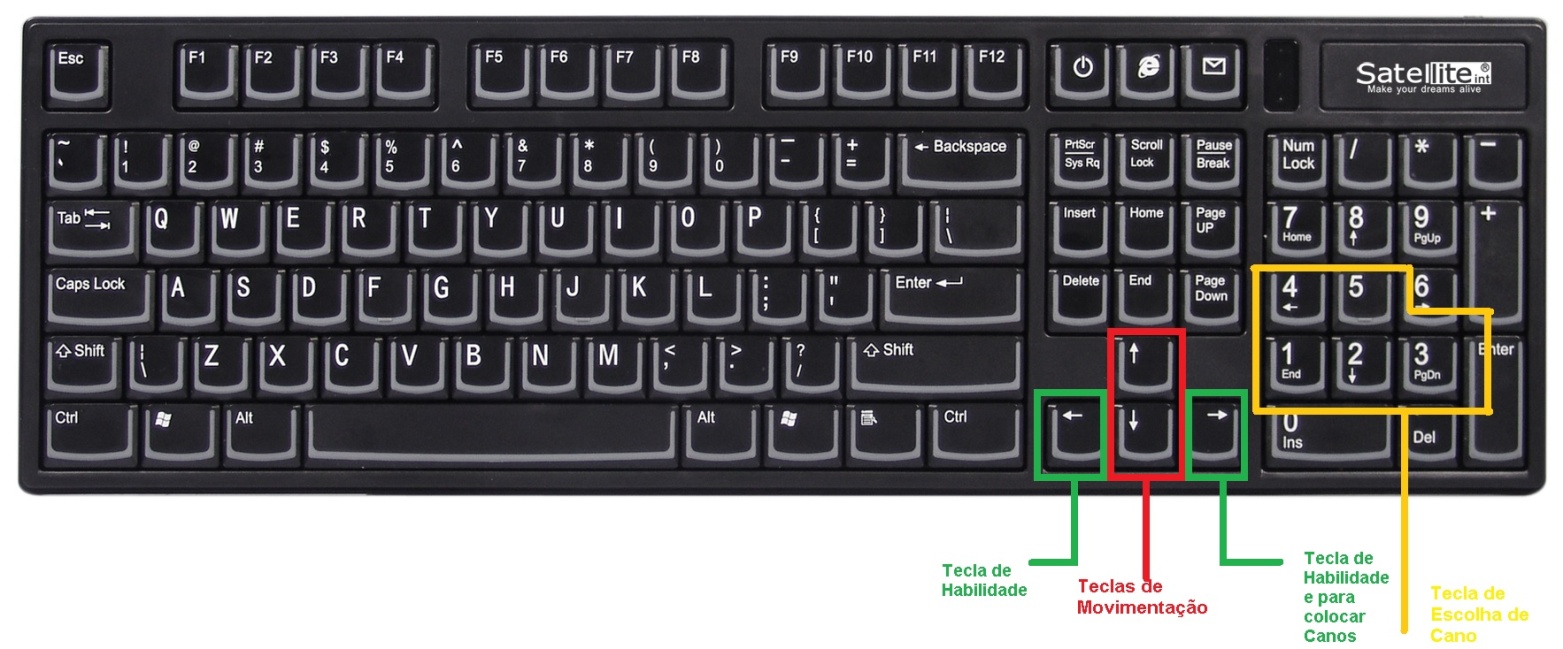


Mario:  
Mario, irmão gêmeo de Luigi, é um protetor da princesa Peach. Para salvá-la coloca canos no caminho do Flappy Birds movendo-se pela extremidade do mapa para que não possam capturá-la. Além disso, pode lançar água com seu cano empurrando o Flappy Bird algumas casas para trás ou caso tenha a flor de fogo e energia suficiente disparar contra o Flappy Bird que capturou a princesa. Ele Também pode fazer uso da Flor de Gelo para desacelerar os pássaros com uma explosão congelante, dando-o tempo de posicionar mais canos.

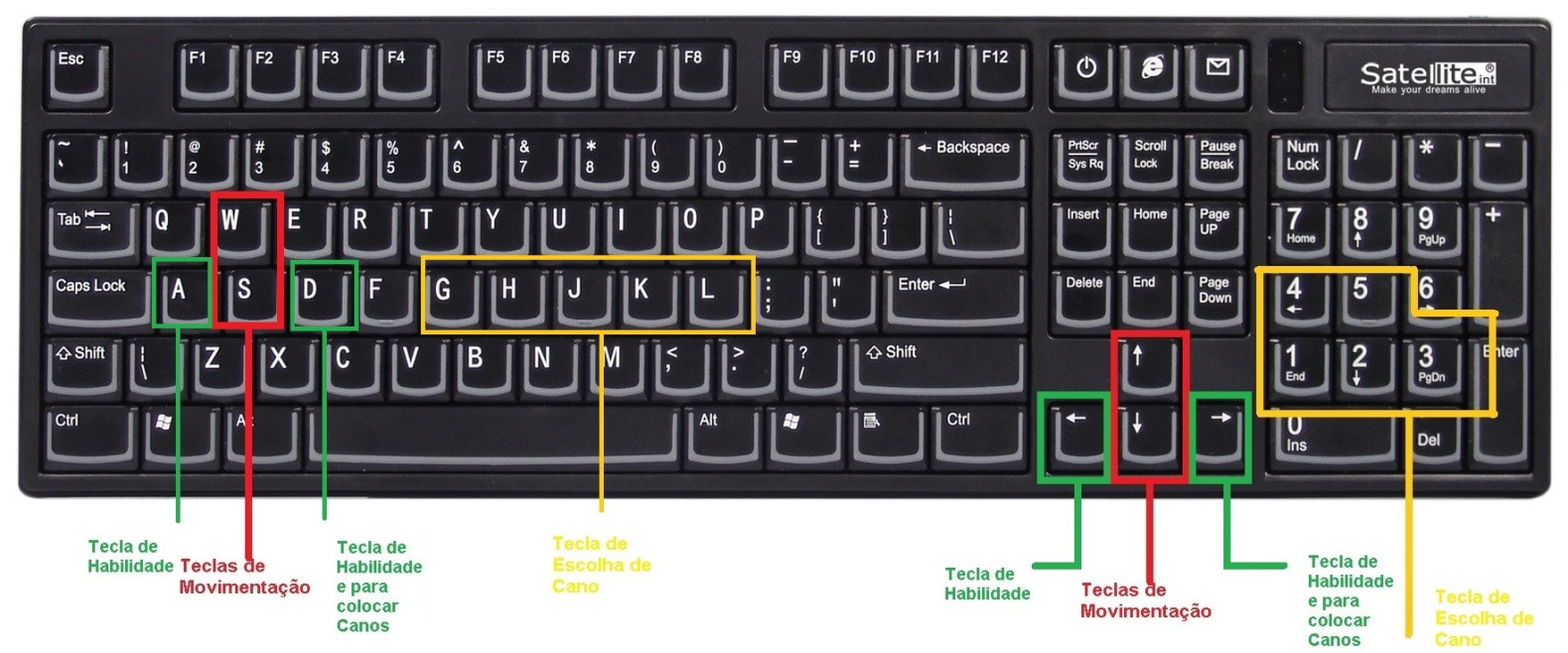
Luigi:  
Luigi, irmão do Mario, o ajuda a salvar a princesa Peach dos Flappy Bird. Para isso conta com os mesmos recursos que o Mario (Canos, Flor de Fogo e de Gelo).

**4. Controles**

Single Player:

****

Multi-Player:

****

**5. Câmera**

A câmera dá uma visão isométrica e o jogador visualiza o ambiente diagonalmente com uma sensação de profundidade

(A imagem mostra a noção de distancia que nosso jogo terá, mas a de profundidade será diferente.)



**6. Universo do Jogo**  
Como haverá apenas um modelo de mapa, optamos por dar ao jogador a possibilidade de escolher o cenário em que a partida se passa, e, em homenagem aos jogos em que nos inspiramos, haverá uma opção para cada Mundo dos jogos mais recentes do Mario, e uma opção para jogar com o tema inspirado no background do jogo Flappy Bird.  
Assim como os cenários, vamos inserir algumas músicas tema de jogos do Mario, a do Flappy Bird, e ainda a do Space Invaders, que representa exatamente a sensação transmitida por esse jogo: uma tensão que cresce quanto mais tempo se passa, por mais simples que seja.  
Abaixo as fases com sua emoção e a imagens que a inspiraram.

1. O mundo Básico

Emoções e sensações: alegria e felicidade

Áudio:   




1. Deserto

Emoções e sensações: solidão, arridez, e cansaço

Áudio:



1. Ilhas/Praias

Emoções e sensações: tranquilidade, diversão e descanso

Áudio:



1. Floresta

Emoções e sensações: Descoberta

Áudio:

1. Mundo da Neve   
   Emoção e sensação: Calmo e frio  
   Áudio:  
    
2. Mundo da Lava

Emoções e sensações: Conquista e perigo

Áudio:

1. Mundo do Flappy Bird  
   Emoção e sensações: Estresse, nervosismo  
   Áudio:  
   
2. Mundo do Arco-íris  
   Emoção e sensação: Felicidade e calma   
   Áudio:

**7. Inimigos**

Os inimigos são apenas Flappy Birds de cinco cores diferentes. Todos os comportamentos serão o mesmo: aparecer à esquerda da tela, planando sobre uma linha, até bater em um cano de sua cor, quando destruirá a si mesmo e ao cano. Se a cor do cano for diferente, o cano será destruído. Contudo, haverá diferença na sua velocidade.

As cinco cores, na ordem de aparição, são: verde, amarelo, laranja, azul e preto.

**Flappy Birds Verde**

Primeiro inimigo do jogo, sempre será o mais lento. Possui o menor custo de moeda para seu cano.

**Flappy Birds Amarelo**Aparece no início do jogo, enquanto o jogador tiver poucos pontos e tiver compreendido. Possui um cano com baixo custo de moedas, apesar de maior que o verde. Será sempre mais rápido que o verde.

**Flappy Birds Laranja**Aparece em um momento intermediário do jogo, enquanto o jogador tiver aprendido a mecânica. Possui um valor por cano e velocidade razoavelmente maior que o verde e amarelo.

**Flappy Birds Azul**

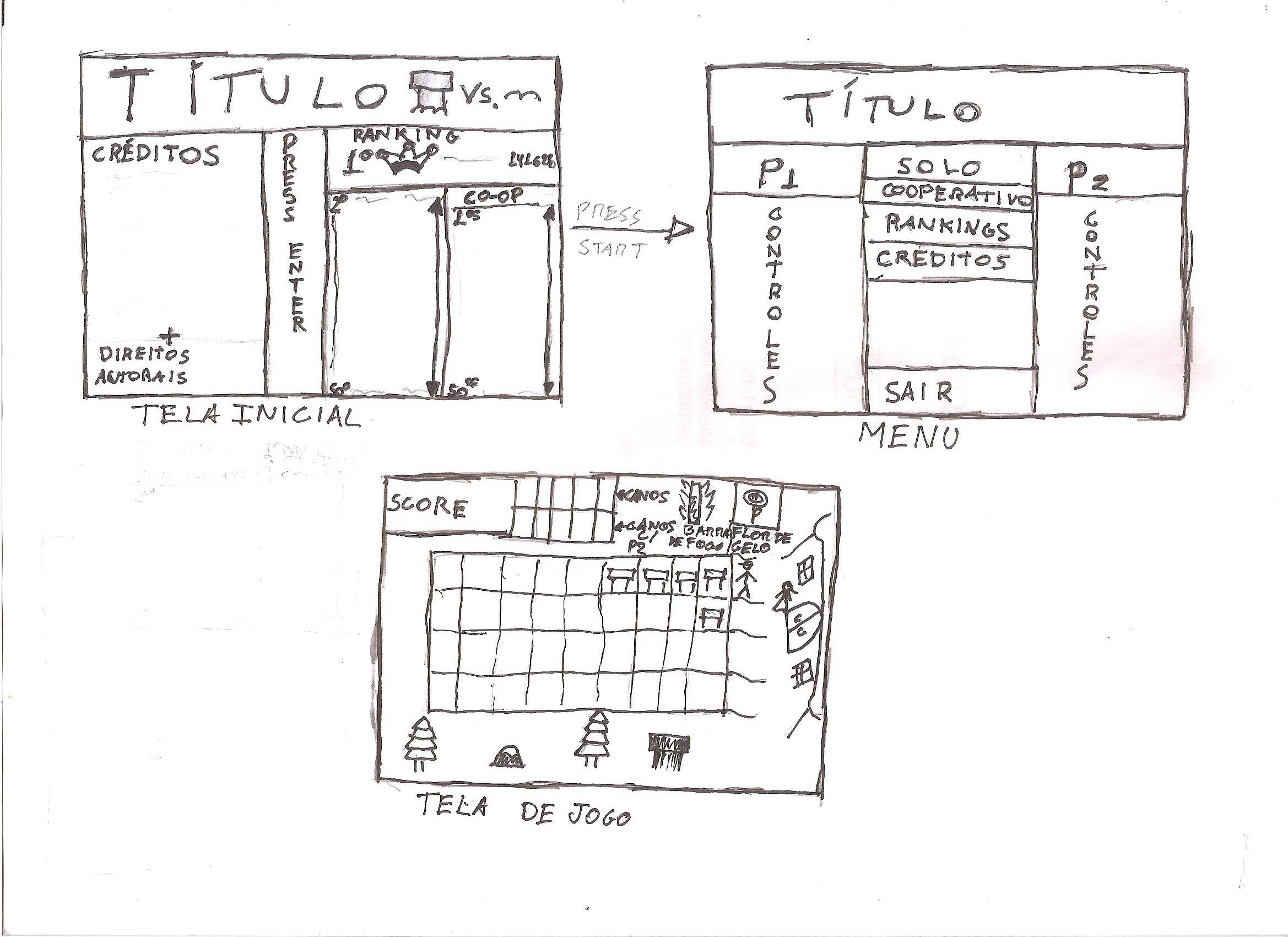
Aparece em um momento avançado do jogo, tem como fim aumentar a dificuldade. Seu cano será muito mais caro que os iniciais. Possui um alto custo e grande velocidade.

**Flappy Birds Preto**Será o ultimo inimigo a surgir, para os melhores jogadores. Inimigo mais difícil do jogo por causa do alto preço pelo seu cano correspondente e velocidade.

 🡪  🡪  🡪

 🡪 

1. **Interface**



**9.** **Cutscenes**

A cutscene será programada e exibida no início do jogo.

Sequencia de imagens:

* Bowser coloca um cano na tela
* Um pássaro voa e bate no cano
* Vários Flappy Birds aparecem na tela movendo-se na mesma direção rapidamente
* Um pássaro aparece na tela carregando a princesa Peach, e logo atrás vem o Mario atirando bolas de fogo.

**10.Cronograma**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Março** | | | | **Abril** | | | | **Maio** | | | | **Junho** | | | |
| **Tarefa/Semana** | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | **Progresso** |
| Escrever o GDD |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Completo |
| Apresentar GDD |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Completo |
| Selecionar/desenhar a arte dos personagens |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Completo |
| Selecionar/desenhar a arte dos cenários |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Completo |
| Desenvolver o sistema de controle do jogador |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Completo |
| Desenvolver sistema de mapas e fases |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Completo |
| Implementar a detecção de colisão |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Completo |
| Desenvolver sistema de pontuação |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Completo |
| Implementar inimigos |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Completo |
| Cutscenes |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Completo |
| Ranking Online |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Incompleto |

**ANEXO 2 – CÓDIGO FONTE**

-- Não está funcionando a subtração de moedas

function Por\_Cano\_keypressed(Cano\_Selecionado, Linhapers, Moedas) -- Armazena Cano\_Selecionado na Matriz Cano

for Coluna = 6, 1, -1 do -- Percorre a Matriz\_Cano ajustando a posição deles.

Matriz\_Cano[Linhapers][Coluna] = Matriz\_Cano[Linhapers][Coluna - 1]

end

Matriz\_Cano[Linhapers][0] = Cano\_Selecionado -- Põe o Cano\_Selecionado

end

function ChecarColisaoPassaro(XP, WP, XC, X) --x do Passaro, Largura do Passaro, x do Cano, Extra

return XC < XP + WP - X

end

function love.load()

Jogo = 0 -- 0-Menu 1-Jogo 2-Salve 3-Game Over 4-Co-op 5-Salve Co-op

Moedas = 50 -- Máximo de Moedas

inimigos = {} -- Vetor de tabelas dos inimigos

Matriz\_Cano = {} -- Armazena a Cor do cano em sua Posição

Pontuacao = 0

tam\_linha = 5\*love.window.getHeight()/32 -- Altura da linha em função da tela

tam\_coluna = 5\*love.window.getWidth()/48 -- Largura da Coluna em função da tela

Mario = { -- Biblioteca do Jogador 1

Cano\_Selecionado = 1, -- 1-Verde 2-Amarelo 3-Azul 4-Vermelho 5-Preto

posx = 3\*love.window.getWidth()/4,

posy = love.window.getHeight()/4,

Linha = 0, -- Armazena em que Linha o Jogador está

animacao = 1, --Imagem que será apresentada

anim\_saida\_down = false, --Controla o pressionameto do teclado(movimentação)

anim\_saida\_up = false,--Controla o pressionameto do teclado(movimentação)

anim\_move = false, --Controla a liberação do personagem para se movimentar

tempo\_saida = 0 --Contador de tempo de movimentação do personagem

}

Luigi = { -- Biblioteca do Jogador 2

Cano\_Selecionado = 1, -- 1-Verde 2-Amarelo 3-Azul 4-Vermelho 5-Preto

posx = 3\*love.window.getWidth()/4 + tam\_coluna,

posy = love.window.getHeight()/4,

Linha = 0, -- Armazena em que Linha o Jogador está

animacao = 1, --Imagem que será apresentada

anim\_saida\_down = false, --Controla o pressionameto do teclado(movimentação)

anim\_saida\_up = false,--Controla o pressionameto do teclado(movimentação)

anim\_move = false, --Controla a liberação do personagem para se movimentar

tempo\_saida = 0 --Contador de tempo de movimentação do personagem

}

Princesa = {

Nivel = 1,--Armazena quantas vezes ela já foi resgatada

PosX = 3\*love.window.getWidth()/4,

PosY = love.window.getHeight()/4,

Cor = 0,--Armazena a cor que o flappybird terá

Contador = 0 -- Armazena o clique de teclas para derrotar o pássaro

}

Controle\_Inimigos = {

Velocidade = 1,-- Cresce em função da pontuação

tempo\_aparicao = 0 --Tempo desde a ultima aparição de um passaro

}

Dificuldade = {

Nivel = 1,

}

BolaFogo = {

Rotacao = 0,

Tamanho = 0,

PosX = Mario.posx,

PosY = 0,

Diferenca = 0 -- Diferença entre o Y da Bola e da Princesa

}

cutscenes = {

contador = 0, --Medidor de tempo

ativador = false, --ativa cutscenes

bowser = {

anim = 2, --Guarda informação sobre a imagem a ser impresa do bowser

posicaox = love.graphics.getWidth(),

posicaoy = 300

},

flappybird = {

posicaox = 0 ,

posicaoy = 300,

contador = 0,

},

bandodeflappy = {

posicaox = {},

posicaoy = {}

}

}

Escolha = { -- Armazena informações da tela de Modo de Jogo

Ativada = false,

Destaque = "Solo" -- Solo ou Cooperativo

}

for i = 1 , 20 , 1 do -- Inicializa o Bando de Flappy da Cutscene

cutscenes.bandodeflappy.posicaox[i] = (-15 + i)\*10 - 120

cutscenes.bandodeflappy.posicaoy[i] = math.random( 50, 250)

end

bowser01 = love.graphics.newImage("browser01.png")

bowser02 = love.graphics.newImage("browser02.png")

bowser03 = love.graphics.newImage("browser03.png")

--Personagens 250x250

flappyverde = love.graphics.newImage("Verde - Flappy Bird.png")

flappyamarelo = love.graphics.newImage("Laranja - Flappy Bird.png")

flappyazul = love.graphics.newImage("Azul Escuro - Flappy Bird.png")

flappyvermelho = love.graphics.newImage("Vermelho - Flappy Bird.png")

flappypreto = love.graphics.newImage("Preto - Flappy Bird.png")

flappyverdemorto = love.graphics.newImage("flappymorto.png")

--128x128

BolaDeFogo = love.graphics.newImage("boladefogo.png")

Cano\_Verde = love.graphics.newImage("Cano\_Verde.png")

Cano\_Amarelo = love.graphics.newImage("Cano\_Laranja.png")

Cano\_Azul = love.graphics.newImage("Cano\_Azul.png")

Cano\_Vermelho = love.graphics.newImage("Cano\_Vermelho.png")

Cano\_Preto = love.graphics.newImage("Cano\_Preto.png")

-- Cano Selecionado

MarioCanoVerde = love.graphics.newImage("MarioCanoVerde.png")

MarioCanoAmarelo = love.graphics.newImage("MarioCanoAmarelo.png")

MarioCanoAzul = love.graphics.newImage("MarioCanoAzul.png")

MarioCanoVermelho = love.graphics.newImage("MarioCanoVermelho.png")

MarioCanoPreto = love.graphics.newImage("MarioCanoPreto.png")

LuigiCanoVerde = love.graphics.newImage("LuigiCanoVerde.png")

LuigiCanoAmarelo = love.graphics.newImage("LuigiCanoAmarelo.png")

LuigiCanoAzul = love.graphics.newImage("LuigiCanoAzul.png")

LuigiCanoVermelho = love.graphics.newImage("LuigiCanoVermelho.png")

LuigiCanoPreto = love.graphics.newImage("LuigiCanoPreto.png")

--32x32

areia1 = love.graphics.newImage("mapa\_praia01.png")

areia2 = love.graphics.newImage("mapa\_praia02.png")

areia3 = love.graphics.newImage("mapa\_praia03.png")

areia4 = love.graphics.newImage("mapa\_praia04.png")

areia5 = love.graphics.newImage("mapa\_praia05.png")

agua = love.graphics.newImage("agua.png")

mario\_normal = love.graphics.newImage("Mario01.png")

mario\_movimento = love.graphics.newImage("Mario02.png")

luigi\_normal = love.graphics.newImage("Luigi01.png")

luigi\_movimento = love.graphics.newImage("Luigi02.png")

-- Princesa

PrincesaVerde = love.graphics.newImage("princesaverde.png")

PrincesaAmarelo = love.graphics.newImage("princesaamarelo.png")

PrincesaAzul = love.graphics.newImage("princesaazul.png")

PrincesaVermelho = love.graphics.newImage("princesavermelho.png")

PrincesaPreto = love.graphics.newImage("princesapreto.png")

for i = 0, 10, 1 do -- Inicializa inimigos[i]

inimigos[i] = {

linha = 0,

posicaox = 0,

cores = 0

}

end

for i= 0, 3, 1 do -- Inicializa a Matriz\_Cano toda como vazia

Matriz\_Cano[i] = {}

for j=0, 6, 1 do

Matriz\_Cano[i][j] = 0

end

end

end

function love.keypressed(key)

if Jogo == 1 or Jogo == 4 then -- Jogo

if key == "kp1" and Dificuldade.Nivel >= 1 then -- Seleção de Canos, com bloqueio das cores ainda não liberadas

Mario.Cano\_Selecionado = 1 -- Verde

elseif key == "kp2" and Dificuldade.Nivel >= 2 then

Mario.Cano\_Selecionado = 2 -- Amarelo

elseif key == "kp3" and Dificuldade.Nivel >= 3 then

Mario.Cano\_Selecionado = 3 -- Azul

elseif key == "kp4" and Dificuldade.Nivel >= 4 then

Mario.Cano\_Selecionado = 4 -- Vermelho

elseif key == "kp5" and Dificuldade.Nivel >= 5 then

Mario.Cano\_Selecionado = 5 -- Preto

end

if key == "left" and Moedas >= Mario.Cano\_Selecionado and Matriz\_Cano[Mario.Linha][6] == 0 then -- Põe o Cano na Matriz-- Mario

Por\_Cano\_keypressed(Mario.Cano\_Selecionado, Mario.Linha, Moedas)

Moedas = Moedas - Mario.Cano\_Selecionado -- Gasta a Moeda

end

if key == "down" and not(Mario.posy == (23\*love.window.getHeight()/32)) then --Movimento do Mario -- Limite Inferior

Mario.anim\_saida\_down = true --Caso o jogador tenha pressionado

Mario.tempo\_saida = 0 --Inicializa o tempo de aniamação

elseif key == "up" and not(Mario.posy == love.window.getHeight()/4) then -- Limite Superior

Mario.anim\_saida\_up = true --Caso o jogador tenha pressionado

Mario.tempo\_saida = 0 --Inicializa o tempo de animação

end

if Jogo == 4 then -- Jogo Cooperativo

if key == "g" and Dificuldade.Nivel >= 1 then -- Seleção de Canos Luigi

Luigi.Cano\_Selecionado = 1 -- Verde

elseif key == "h" and Dificuldade.Nivel >= 2 then

Luigi.Cano\_Selecionado = 2 -- Amarelo

elseif key == "j" and Dificuldade.Nivel >= 3 then

Luigi.Cano\_Selecionado = 3 -- Azul

elseif key == "t" and Dificuldade.Nivel >= 4 then

Luigi.Cano\_Selecionado = 4 -- Vermelho

elseif key == "y" and Dificuldade.Nivel >= 5 then

Luigi.Cano\_Selecionado = 5 -- Preto

end

if key == "a" and Moedas >= Luigi.Cano\_Selecionado and Matriz\_Cano[Luigi.Linha][6] == 0 then -- Põe o Cano na Matriz-- Luigi

Por\_Cano\_keypressed(Luigi.Cano\_Selecionado, Luigi.Linha)

Moedas = Moedas - Luigi.Cano\_Selecionado -- Gasta a Moeda

end

if key == "s" and not(Luigi.posy == (23\*love.window.getHeight()/32)) then -- Movimento Luigi -- Limite Inferior

Luigi.anim\_saida\_down = true --Caso o jogador tenha pressionado

Luigi.tempo\_saida = 0 --Inicializa o tempo de aniamação

elseif key == "w" and not(Luigi.posy == love.window.getHeight()/4) then -- Limite Superior

Luigi.anim\_saida\_up = true --Caso o jogador tenha pressionado

Luigi.tempo\_saida = 0 --Inicializa o tempo de animação

end

end

elseif Jogo == 2 or Jogo == 5 then -- Salve a Princesa

if key == "left" then -- Cresce a Bola de Fogo

if Jogo == 2 and Princesa.Contador < 10 \* Princesa.Nivel then

Princesa.Contador = Princesa.Contador + 1

BolaFogo.Tamanho = (Princesa.Contador/(10\*Princesa.Nivel))

elseif Jogo == 5 and Princesa.Contador < 20 \* Princesa.Nivel then

Princesa.Contador = Princesa.Contador + 1

BolaFogo.Tamanho = (Princesa.Contador/(20\*Princesa.Nivel))

end

end

if Jogo == 5 then

if key == "f" then

Princesa.Contador = Princesa.Contador + 1

BolaFogo.Tamanho = (Princesa.Contador/(Princesa.Nivel\*20))

end

end

elseif Jogo == 3 then -- Game Over

if key == "return" then -- Reinicia o Jogo

for i = 0, 3, 1 do

for j = 0, 6, 1 do

Matriz\_Cano[i][j] = 0

end

end

for i = 0, 10, 1 do

inimigos[i].cores = 0

end

Pontuacao = 0

Mario.Cano\_Selecionado = 1

Luigi.Cano\_Selecionado = 1

Mario.posy = love.window.getHeight()/4

Luigi.posy = love.window.getHeight()/4

Mario.Linha = 0

Luigi.Linha = 0

Princesa.Nivel = 1

Princesa.Contador = 0

Princesa.PosX = 3 \* love.window.getWidth()/4

Princesa.Cor = 0

Moedas = 50

Jogo = 0

Escolha.Ativada = true

end

elseif Jogo == 0 then -- Menu

if key == "return" and cutscenes.ativador == false and Escolha.Ativada == false then

cutscenes.ativador = true

end

if Escolha.Ativada == true then

if key == "up" or key == "down" then

if Escolha.Destaque == "Cooperativo" then

Escolha.Destaque = "Solo"

elseif Escolha.Destaque == "Solo" then

Escolha.Destaque = "Cooperativo"

end

end

if key == "return" then

if Escolha.Destaque == "Solo" then

Jogo = 1

elseif Escolha.Destaque == "Cooperativo" then

Jogo = 4

end

Escolha.Ativada = false

end

end

if cutscenes.ativador == true then

if key == "kpenter" then

cutscenes.ativador = false

Escolha.Ativada = true

end

end

end

if key == "escape" then -- Saída do Programa

love.event.quit()

end

end

function love.update(dt)

if Jogo == 1 or Jogo == 4 then-- Jogo

if Mario.anim\_saida\_down == true then -- Movimento do Mario para Baixo

Mario.tempo\_saida = Mario.tempo\_saida + dt --Conta o tempo

if Mario.tempo\_saida > 0.07 then

Mario.animacao = 1 --Desativa animação

Mario.anim\_saida\_down = false --Desativa a sinalização tecla pressionada

Mario.anim\_move = false --Controle do elseif

elseif Mario.tempo\_saida > 0.05 and Mario.anim\_move == false then

Mario.posy = Mario.posy + tam\_linha --Movimentação

Mario.Linha = Mario.Linha + 1 --Controle da linha que o mário se encontra

Mario.anim\_move = true --Controle do elseif

elseif Mario.tempo\_saida > 0.025 then

Mario.animacao = 2 --Carrega a animação

end

end

if Mario.anim\_saida\_up == true then -- Movimento do Mario para Cima

Mario.tempo\_saida = Mario.tempo\_saida + dt --Conta o tempo

if Mario.tempo\_saida > 0.07 then

Mario.animacao = 1 --Desativa animação

Mario.anim\_saida\_up = false --Desativa a sinalização de tecla pressionada

Mario.anim\_move = false --Controle do elseif

elseif Mario.tempo\_saida > 0.05 and Mario.anim\_move == false then

Mario.posy = Mario.posy - tam\_linha --Movimentação

Mario.Linha = Mario.Linha - 1 --Controle da linha que o mário se encontra

Mario.anim\_move = true --Controle do elseif

elseif Mario.tempo\_saida > 0.025 then

Mario.animacao = 2 --Carrega a animação

end

end

if Jogo == 4 then

if Luigi.anim\_saida\_down == true then -- Movimento do Luigi para Baixo

Luigi.tempo\_saida = Luigi.tempo\_saida + dt --Conta o tempo

if Luigi.tempo\_saida > 0.07 then

Luigi.animacao = 1 --Desativa animação

Luigi.anim\_saida\_down = false --Desativa a sinalização tecla pressionada

Luigi.anim\_move = false --Controle do elseif

elseif Luigi.tempo\_saida > 0.05 and Luigi.anim\_move == false then

Luigi.posy = Luigi.posy + tam\_linha --Movimentação

Luigi.Linha = Luigi.Linha + 1 --Controle da linha que o Luigi se encontra

Luigi.anim\_move = true --Controle do elseif

elseif Luigi.tempo\_saida > 0.025 then

Luigi.animacao = 2 --Carrega a animação

end

end

if Luigi.anim\_saida\_up == true then -- Movimento do Luigi para Cima

Luigi.tempo\_saida = Luigi.tempo\_saida + dt --Conta o tempo

if Luigi.tempo\_saida > 0.07 then

Luigi.animacao = 1 --Desativa animação

Luigi.anim\_saida\_up = false --Desativa a sinalização de tecla pressionada

Luigi.anim\_move = false --Controle do elseif

elseif Luigi.tempo\_saida > 0.05 and Luigi.anim\_move == false then

Luigi.posy = Luigi.posy - tam\_linha --Movimentação

Luigi.Linha = Luigi.Linha - 1 --Controle da linha que o Luigi se encontra

Luigi.anim\_move = true --Controle do elseif

elseif Luigi.tempo\_saida > 0.025 then

Luigi.animacao = 2 --Carrega a animação

end

end

end

for i = 0 , 10, 1 do --Atualizar a posição do pássaro

if inimigos[i].cores == 1 then

inimigos[i].posicaox = inimigos[i].posicaox + love.window.getWidth()/8 \* dt \* Controle\_Inimigos.Velocidade -- 100

elseif inimigos[i].cores == 2 then

inimigos[i].posicaox = inimigos[i].posicaox + 6 \* love.window.getWidth()/40 \* dt \* Controle\_Inimigos.Velocidade -- 120

elseif inimigos[i].cores == 3 then

inimigos[i].posicaox = inimigos[i].posicaox + 7 \* love.window.getWidth()/40 \* dt \* Controle\_Inimigos.Velocidade -- 140

elseif inimigos[i].cores == 4 then

inimigos[i].posicaox = inimigos[i].posicaox + love.window.getWidth()/5 \* dt \* Controle\_Inimigos.Velocidade -- 160

elseif inimigos[i].cores == 5 then

inimigos[i].posicaox = inimigos[i].posicaox + 9 \* love.window.getWidth()/40 \* dt \* Controle\_Inimigos.Velocidade -- 180

end

end

--Controle de criação de inimigos

Controle\_Inimigos.tempo\_aparicao = Controle\_Inimigos.tempo\_aparicao + dt

if Jogo == 1 then

if Controle\_Inimigos.tempo\_aparicao > 1.5 then

local i = 0

while (i < 10) do

if inimigos[i].cores == 0 then

inimigos[i].cores = math.random(1,Dificuldade.Nivel)

inimigos[i].linha = math.random(0,3)

inimigos[i].posicaox = 0

i = 10

end

i = i + 1

end

Controle\_Inimigos.tempo\_aparicao = 0

end

elseif Jogo == 4 then

if Controle\_Inimigos.tempo\_aparicao > 1 then

local i = 0

while (i < 10) do

if inimigos[i].cores == 0 then

inimigos[i].cores = math.random(1,Dificuldade.Nivel)

inimigos[i].linha = math.random(0,3)

inimigos[i].posicaox = 0

i = 10

end

i = i + 1

end

Controle\_Inimigos.tempo\_aparicao = 0

end

end

for i = 0, 10, 1 do --Colisão dos Passaros

for j = 6, 0, -1 do

if ChecarColisaoPassaro(inimigos[i].posicaox, tam\_coluna, 3 \* love.window.getWidth()/4 - (j+1) \* tam\_coluna, 2 \* tam\_coluna/5) and Matriz\_Cano[inimigos[i].linha][j] == inimigos[i].cores and inimigos[i].cores ~= 0 then

Pontuacao = Pontuacao + inimigos[i].cores

inimigos[i].cores = 0

Moedas = Moedas + Matriz\_Cano[inimigos[i].linha][j]

Matriz\_Cano[inimigos[i].linha][j] = 0

elseif ChecarColisaoPassaro(inimigos[i].posicaox, tam\_coluna, 3 \* love.window.getWidth()/4 - (j+1) \* tam\_coluna, 2 \* tam\_coluna/5) and Matriz\_Cano[inimigos[i].linha][j] ~= inimigos[i].cores and Matriz\_Cano[inimigos[i].linha][j] ~= 0 then

Moedas = Moedas + Matriz\_Cano[inimigos[i].linha][j]

Matriz\_Cano[inimigos[i].linha][j] = 0

end

end

if inimigos[i].posicaox >= Mario.posx - 2 \* tam\_coluna/5 then -- Controle da mudança de jogo

Princesa.Cor = inimigos[i].cores

Princesa.Linha = inimigos[i].linha

Princesa.PosY = love.window.getHeight()/4 + tam\_linha \* Princesa.Linha

BolaFogo.PosY = Mario.posy

BolaFogo.Diferenca = (Princesa.PosY + (Princesa.Linha \* tam\_linha)) - BolaFogo.PosY

Controle\_Inimigos.tempo\_aparicao = 0

for t = 0, 10, 1 do

inimigos[t].cores = 0

inimigos[t].posicaox = -tam\_coluna

end

for t = 0, 3, 1 do

for g = 0, 6, 1 do

Matriz\_Cano[t][g] = 0

end

end

if Jogo == 1 then

Jogo = 2

elseif Jogo == 4 then

Jogo = 5

end

end

if inimigos[i].cores == 0 then -- Remove os pássaros inexistentes do caminho

inimigos[i].posicaox = - tam\_coluna

end

end

Controle\_Inimigos.Velocidade = ((Pontuacao+1)^(1/4)) -- mudança da velocidade dos passaros

elseif Jogo == 2 or Jogo == 5 then -- Salve a princesa

if Princesa.PosX <= 0 then

BolaFogo.Tamanho = 0

Jogo = 3

end

if Jogo == 2 then

if Princesa.Contador < (10 \* Princesa.Nivel) then

Princesa.PosX = Princesa.PosX - love.window.getWidth()/8 \* dt

elseif Princesa.Contador >= (10 \* Princesa.Nivel) then

BolaFogo.PosX = BolaFogo.PosX - (1500 \* dt)

if BolaFogo.PosY < Princesa.PosY + 30 then

BolaFogo.PosY = BolaFogo.PosY + 3000 \* dt

elseif BolaFogo.PosY > Princesa.PosY + 50 then

BolaFogo.PosY = BolaFogo.PosY - 3000 \* dt

end

if BolaFogo.PosX <= Princesa.PosX + 75 then

BolaFogo.Tamanho = 0

BolaFogo.PosX = Mario.posx

Pontuacao = Pontuacao + Princesa.Cor

Princesa.Cor = 0

Princesa.PosX = 3 \* love.window.getWidth()/4

Princesa.Nivel = Princesa.Nivel + 1

Princesa.Contador = 0

Moedas = 50

Jogo = 1

end

end

elseif Jogo == 5 then

if Princesa.Contador < (20 \* Princesa.Nivel) then

Princesa.PosX = Princesa.PosX - love.window.getWidth()/8 \* dt

elseif Princesa.Contador >= (20 \* Princesa.Nivel) then

BolaFogo.PosX = BolaFogo.PosX - (1500 \* dt)

if BolaFogo.PosY < Princesa.PosY + 30 then

BolaFogo.PosY = BolaFogo.PosY + 3000 \* dt

elseif BolaFogo.PosY > Princesa.PosY + 50 then

BolaFogo.PosY = BolaFogo.PosY - 3000 \* dt

end

if BolaFogo.PosX <= Princesa.PosX + 75 then

BolaFogo.Tamanho = 0

BolaFogo.PosX = Mario.posx

Pontuacao = Pontuacao + Princesa.Cor

Princesa.Cor = 0

Princesa.PosX = 3 \* love.window.getWidth()/4

Princesa.Nivel = Princesa.Nivel + 1

Princesa.Contador = 0

Moedas = 50

Jogo = 4

end

end

end

elseif Jogo == 0 then -- Menu

--cutscenes

if cutscenes.ativador == true then

if cutscenes.bandodeflappy.posicaox[1] > love.window.getWidth() then

cutscenes.ativador = false

Escolha.Ativada = true

end

cutscenes.contador = cutscenes.contador + dt

if cutscenes.bowser.posicaox >= love.window.getWidth()/2 then

cutscenes.bowser.posicaox = cutscenes.bowser.posicaox - 75 \* dt --15\*dt

if cutscenes.contador >= 0.3 then

if cutscenes.bowser.anim == 2 then

cutscenes.bowser.anim = 3

cutscenes.contador = 0

elseif cutscenes.bowser.anim == 3 then

cutscenes.bowser.anim = 2

cutscenes.contador = 0

end

end

else --Nesse momento bowser já colocou cano

if cutscenes.flappybird.posicaox < cutscenes.bowser.posicaox - 165 then --Flappybird se move

cutscenes.flappybird.posicaox = cutscenes.flappybird.posicaox + 90 \* dt

cutscenes.contador = 0

elseif cutscenes.flappybird.posicaox >= cutscenes.bowser.posicaox - 165 and cutscenes.flappybird.posicaoy <= 380 then

cutscenes.flappybird.posicaoy = cutscenes.flappybird.posicaoy + 40 \* dt

elseif cutscenes.flappybird.posicaoy >= 380 and cutscenes.contador >= 1.5 then

for i = 1 ,20 ,1 do

cutscenes.bandodeflappy.posicaox[i] = cutscenes.bandodeflappy.posicaox[i] + (30+i) \* (20 \* dt)

end

end

end

end

end

if Pontuacao < 15 then -- Controle da Dificuldade em função da Pontuação

Dificuldade.Nivel = 1

elseif Pontuacao >= 15 and Pontuacao < 40 then

Dificuldade.Nivel = 2

elseif Pontuacao >= 40 and Pontuacao < 80 then

Dificuldade.Nivel = 3

elseif Pontuacao >= 80 and Pontuacao < 140 then

Dificuldade.Nivel = 4

elseif Pontuacao >= 140 then

Dificuldade.Nivel = 5

end

end

function love.draw()

if Jogo ==1 or Jogo == 2 or Jogo == 4 or Jogo == 5 then

love.graphics.setBackgroundColor(160,194,250)

for i = 0, 800, 32 do --Impressão do mapa

for j = 200, 600, 32 do

if i < (j - 200)/2 then

love.graphics.draw(agua, i, j)

elseif i == (j-200)/2 then

love.graphics.draw(areia2, i, j)

else

love.graphics.draw(areia4, i, j)

end

end

end

if Mario.animacao == 1 then -- Animação do Mario normal

love.graphics.draw(mario\_normal,

Mario.posx,

Mario.posy,

0,

5\*love.window.getWidth()/1536,

15\*love.window.getHeight()/4096

)

elseif Mario.animacao == 2 then -- Animação do Mario andando

love.graphics.draw(mario\_movimento,

Mario.posx,

Mario.posy,

0,

5\*love.window.getWidth()/1536,

15\*love.window.getHeight()/4096

)

end

--Seleção de canos

love.graphics.draw(Cano\_Verde,

love.window.getWidth() - tam\_coluna,

love.window.getHeight() \*8 /9,

0,

tam\_coluna/128,

tam\_linha/196)

love.graphics.draw(Cano\_Amarelo,

love.window.getWidth() - 2 \* tam\_coluna,

love.window.getHeight() \*8/9,

0,

tam\_coluna/128,

tam\_linha/196)

love.graphics.draw(Cano\_Azul,

love.window.getWidth() - 3 \* tam\_coluna,

love.window.getHeight() \*8/9,

0,

tam\_coluna/128,

tam\_linha/196)

love.graphics.draw(Cano\_Vermelho,

love.window.getWidth() - 4 \* tam\_coluna,

love.window.getHeight() \*8/9,

0,

tam\_coluna/128,

tam\_linha/196)

love.graphics.draw(Cano\_Preto,

love.window.getWidth() - 5 \* tam\_coluna,

love.window.getHeight() \*8/9,

0,

tam\_coluna/128,

tam\_linha/196)

if Mario.Cano\_Selecionado == 1 then

love.graphics.draw(MarioCanoVerde,

love.window.getWidth() - tam\_coluna,

love.window.getHeight() \*8 /9,

0,

tam\_coluna/128,

tam\_linha/196)

elseif Mario.Cano\_Selecionado == 2 then

love.graphics.draw(MarioCanoAmarelo,

love.window.getWidth() - 2 \* tam\_coluna,

love.window.getHeight() \*8/9,

0,

tam\_coluna/128,

tam\_linha/196)

elseif Mario.Cano\_Selecionado == 3 then

love.graphics.draw(MarioCanoAzul,

love.window.getWidth() - 3 \* tam\_coluna,

love.window.getHeight() \*8/9,

0,

tam\_coluna/128,

tam\_linha/196)

elseif Mario.Cano\_Selecionado == 4 then

love.graphics.draw(MarioCanoVermelho,

love.window.getWidth() - 4 \* tam\_coluna,

love.window.getHeight() \*8/9,

0,

tam\_coluna/128,

tam\_linha/196)

elseif Mario.Cano\_Selecionado == 5 then

love.graphics.draw(MarioCanoPreto,

love.window.getWidth() - 5 \* tam\_coluna,

love.window.getHeight() \*8/9,

0,

tam\_coluna/128,

tam\_linha/196)

end

if Jogo == 4 or Jogo == 5 then

if Luigi.animacao == 1 then -- Animação do Luigi normal

love.graphics.draw(luigi\_normal,

Luigi.posx,

Luigi.posy,

0,

5\*love.window.getWidth()/1536,

15\*love.window.getHeight()/4096

)

elseif Luigi.animacao == 2 then -- Animação do Luigi andando

love.graphics.draw(luigi\_movimento,

Luigi.posx,

Luigi.posy,

0,

5\*love.window.getWidth()/1536,

15\*love.window.getHeight()/4096

)

end

love.graphics.draw(Cano\_Verde,

love.window.getWidth()/2 - tam\_coluna,

love.window.getHeight() \*8 /9,

0,

tam\_coluna/128,

tam\_linha/196)

love.graphics.draw(Cano\_Amarelo,

love.window.getWidth()/2 - 2 \* tam\_coluna,

love.window.getHeight() \*8/9,

0,

tam\_coluna/128,

tam\_linha/196)

love.graphics.draw(Cano\_Azul,

love.window.getWidth()/2 - 3 \* tam\_coluna,

love.window.getHeight() \*8/9,

0,

tam\_coluna/128,

tam\_linha/196)

love.graphics.draw(Cano\_Vermelho,

love.window.getWidth()/2 - 4 \* tam\_coluna,

love.window.getHeight() \*8/9,

0,

tam\_coluna/128,

tam\_linha/196)

love.graphics.draw(Cano\_Preto,

love.window.getWidth()/2 - 5 \* tam\_coluna,

love.window.getHeight() \*8/9,

0,

tam\_coluna/128,

tam\_linha/196)

if Luigi.Cano\_Selecionado == 1 then

love.graphics.draw(LuigiCanoVerde,

love.window.getWidth()/2 - tam\_coluna,

love.window.getHeight() \*8 /9,

0,

tam\_coluna/128,

tam\_linha/196)

elseif Luigi.Cano\_Selecionado == 2 then

love.graphics.draw(LuigiCanoAmarelo,

love.window.getWidth()/2 - 2 \* tam\_coluna,

love.window.getHeight() \*8/9,

0,

tam\_coluna/128,

tam\_linha/196)

elseif Luigi.Cano\_Selecionado == 3 then

love.graphics.draw(LuigiCanoAzul,

love.window.getWidth()/2 - 3 \* tam\_coluna,

love.window.getHeight() \*8/9,

0,

tam\_coluna/128,

tam\_linha/196)

elseif Luigi.Cano\_Selecionado == 4 then

love.graphics.draw(LuigiCanoVermelho,

love.window.getWidth()/2 - 4 \* tam\_coluna,

love.window.getHeight() \*8/9,

0,

tam\_coluna/128,

tam\_linha/196)

elseif Luigi.Cano\_Selecionado == 5 then

love.graphics.draw(LuigiCanoPreto,

love.window.getWidth()/2 - 5 \* tam\_coluna,

love.window.getHeight() \*8/9,

0,

tam\_coluna/128,

tam\_linha/196)

end

end

for i = 0, 3, 1 do -- Desenhar a Matriz\_Cano

for j = 0, 6, 1 do

if Matriz\_Cano[i][j] == 1 then --Canos verdes

love.graphics.draw(Cano\_Verde,

3 \* love.window.getWidth()/4 - (j+1) \* tam\_coluna,

love.window.getHeight()/4 + i \* tam\_linha,

0,

tam\_coluna/128,

tam\_linha/128

)

elseif Matriz\_Cano[i][j] == 2 then --Canos amarelos

love.graphics.draw(Cano\_Amarelo,

3 \* love.window.getWidth()/4 - (j+1) \* tam\_coluna,

love.window.getHeight()/4 + i \* tam\_linha,

0,

tam\_coluna/128,

tam\_linha/128)

elseif Matriz\_Cano[i][j] == 3 then --Canos azuis

love.graphics.draw(Cano\_Azul,

3 \* love.window.getWidth()/4 - (j+1) \* tam\_coluna,

love.window.getHeight()/4 + i \* tam\_linha,

0,

tam\_coluna/128,

tam\_linha/128

)

elseif Matriz\_Cano[i][j] == 4 then --Canos vermelhos

love.graphics.draw(Cano\_Vermelho,

3 \* love.window.getWidth()/4 - (j+1) \* tam\_coluna,

love.window.getHeight()/4 + i \* tam\_linha,

0,

tam\_coluna/128,

tam\_linha/128

)

elseif Matriz\_Cano[i][j] == 5 then --Canos pretos

love.graphics.draw(Cano\_Preto,

3 \* love.window.getWidth()/4 - (j+1) \* tam\_coluna,

love.window.getHeight()/4 + i \* tam\_linha,

0,

tam\_coluna/128,

tam\_linha/128

)

end

end

end

for i = 0, 10, 1 do

if inimigos[i].cores == 1 then

love.graphics.draw(flappyverde,

inimigos[i].posicaox,

love.window.getHeight()/4 + inimigos[i].linha \* tam\_linha,

0,

tam\_coluna/250,

tam\_linha/250

)

elseif inimigos[i].cores == 2 then

love.graphics.draw(flappyamarelo,

inimigos[i].posicaox,

love.window.getHeight()/4 + inimigos[i].linha \* tam\_linha,

0,

tam\_coluna/250,

tam\_linha/250

)

elseif inimigos[i].cores == 3 then

love.graphics.draw(flappyazul,

inimigos[i].posicaox,

love.window.getHeight()/4 + inimigos[i].linha \* tam\_linha,

0,

tam\_coluna/250,

tam\_linha/250

)

elseif inimigos[i].cores == 4 then

love.graphics.draw(flappyvermelho,

inimigos[i].posicaox,

love.window.getHeight()/4 + inimigos[i].linha \* tam\_linha,

0,

tam\_coluna/250,

tam\_linha/250

)

elseif inimigos[i].cores == 5 then

love.graphics.draw(flappypreto,

inimigos[i].posicaox,

love.window.getHeight()/4 + inimigos[i].linha \* tam\_linha,

0,

tam\_coluna/250,

tam\_linha/250

)

end

end

--Imprime a princesa no modo salve a princesa

if Princesa.Cor == 1 then

love.graphics.draw(PrincesaVerde,

Princesa.PosX,

Princesa.PosY,

0,

tam\_coluna/592,

tam\_linha/750

)

elseif Princesa.Cor == 2 then

love.graphics.draw(PrincesaAmarelo,

Princesa.PosX,

Princesa.PosY,

0,

tam\_coluna/592,

tam\_linha/750

)

elseif Princesa.Cor == 3 then

love.graphics.draw(PrincesaAzul,

Princesa.PosX,

Princesa.PosY,

0,

tam\_coluna/592,

tam\_linha/750

)

elseif Princesa.Cor == 4 then

love.graphics.draw(PrincesaVermelho,

Princesa.PosX,

Princesa.PosY,

0,

tam\_coluna/592,

tam\_linha/750

)

elseif Princesa.Cor == 5 then

love.graphics.draw(PrincesaPreto,

Princesa.PosX,

Princesa.PosY,

0,

tam\_coluna/592,

tam\_linha/750

)

end

--Interface

love.graphics.print("Moedas: ", 0, 100)

love.graphics.print(Moedas, 100, 100)

love.graphics.print("Pontuação: ", 0, 140)

love.graphics.print(Pontuacao, 100, 140)

love.graphics.draw(BolaDeFogo, BolaFogo.PosX, BolaFogo.PosY, 0, BolaFogo.Tamanho, BolaFogo.Tamanho, (128 \* BolaFogo.Tamanho)/2, (128 \* BolaFogo.Tamanho)/2)

elseif Jogo == 3 then

love.graphics.setBackgroundColor( 0, 0, 0)

love.graphics.print("Pontuação", 370, 285)

love.graphics.print(Pontuacao, 400, 300)

elseif Jogo == 0 then

if cutscenes.ativador == false and Escolha.Ativada == false then

love.graphics.setBackgroundColor( 0, 0, 0)

love.graphics.print("Iniciar jogo",370, 265)

love.graphics.print("pressione enter",400, 300)

elseif cutscenes.ativador == true then --Useiário pressionou enter

if cutscenes.bowser.posicaox <= love.window.getWidth()/2 then

--Após o bowser parar de se mover

if cutscenes.contador<=0.33 then

love.graphics.draw(bowser01,cutscenes.bowser.posicaox,cutscenes.bowser.posicaoy)--Fixa a imagem do bowser

end

love.graphics.draw(Cano\_Verde, cutscenes.bowser.posicaox - 138,cutscenes.bowser.posicaoy)--Coloca o cano

if cutscenes.flappybird.posicaox < cutscenes.bowser.posicaox - 165 then

love.graphics.draw(flappyverde, cutscenes.flappybird.posicaox, cutscenes.flappybird.posicaoy, 0, 1/4, 1/4)

else

love.graphics.draw(flappyverdemorto, cutscenes.flappybird.posicaox + 50, cutscenes.flappybird.posicaoy + 50, -3, 1/4, 1/4)

if cutscenes.contador >0.4 then

for i = 1, 20 , 1 do

love.graphics.draw(flappyverde,

cutscenes.bandodeflappy.posicaox[i],

cutscenes.bandodeflappy.posicaoy[i],

0,

1/4,

1/4)

end

end

end

else

--Controle da animação do bowser enquanto o bowser caminha na tela

if cutscenes.bowser.anim == 1 and not(cutscenes.bowser.posicaox >= love.window.getWidth()/2) then

love.graphics.draw(bowser01, cutscenes.bowser.posicaox, cutscenes.bowser.posicaoy)

elseif cutscenes.bowser.anim == 2 then

love.graphics.draw(bowser02, cutscenes.bowser.posicaox, cutscenes.bowser.posicaoy)

elseif cutscenes.bowser.anim == 3 then

love.graphics.draw(bowser03, cutscenes.bowser.posicaox, cutscenes.bowser.posicaoy)

end

end

elseif Escolha.Ativada == true then

love.graphics.print("Escolha um Modo de Jogo :", 300, 200)

if Escolha.Destaque == "Solo" then

love.graphics.print("->", 368, 290)

elseif Escolha.Destaque == "Cooperativo" then

love.graphics.print("->", 368, 310)

end

love.graphics.print("Solo", 380, 290)

love.graphics.print("Cooperativo", 380, 310)

end

end

end

--Nota: Não esquecer de colocar na cutscenes o valor do cano em vez do 138(Valor também usado na update