**Controles Funcionais**

No jogo o personagem principal só tem a movimentação.

Direita, esquerda e pulo.

**Artes**

Somente a arte do Personagem Principal foi desenhada por mim, às demais eu peguei de alguns pack prontos da Unity.

**Desenvolvimento**

Depois da arte do personagem, eu comecei pelo **movimento** dele que são o andar e o pular juntamente com a animação.

Posteriormente a isso decidi tentar construir o **level design** para tentar entender onde os itens que eu tinha planejado iriam ficar. Após isso, resolvi tentar colocar o **Parallax** para o personagem não ficar andando em um fundo escuro.

**Movimento**

O movimento acho que não tem muito segredo, pois foi utilizado o Rigidbody.

**Animação**

Sem segredos também, foi usado Ifs para entender se estava pulando, andando para ativar as animações. E foi utilizado um “detector de chão” para identificar se o personagem realmente está no chão.

**Level Design**

Foi construído vários tiles maps ao longo dessa entrega, pois houve muitos problemas na parte das ordens dos layers por conta da iluminação, a princípio eram somente 3 tilesmaps ( detalhes, escadas e Tilemap). Um outro fator para aumentar os tiles, foi a passagem secreta do player que estava na mesma camada de detalhes, mas tive que retirar, pois os detalhes o Player precisa estar na frente e a passagem secreta o Player precisa ficar para trás.

Logo no início da fase, o Player encontra uma bola de fogo para começar a entender o objetivo de jogo.

Para o Player completar o cenário ele precisa capturar todas as bolas de fogo que na verdade são espíritos de crianças aprisionadas, mas antes de completar ele precisará enfrentar o Anjo que na verdade é o inimigo dessa fase.

**Parallax**

Para a câmera do Parallax usei o Cinemachine porque estava utilizando para fazer seguir o personagem, sem precisar programar. Antes eu tinha deixado a câmera como a filha do personagem, mas estava ficando muito estranha a visualização e acabei encontrando o Cinemachine.

Para o código do Parallax foi utilizado:

* a câmera para ter como referência
* o tamanho da imagem e a sua posição inicial
* velocidade para controlar a movimentação do parallax

Não foi utilizado o Update porque a movimentação que utilizamos no personagem principal é o fixed update.

Foi criado mais uma variável chamada temp para pegar a posição da câmera e multiplicado por 1 que é multiplicado pela velocidade.

E uma outra “dist” que é a posição da câmera multiplicada pela velocidade.

Para a movimentação funcionar foi criado um Vector3 sem mexer em y e z, já que o parallax só acontece no eixo x.

Coloquei também uma condição para que sempre que o “temp” for maior que a posição câmera e o comprimento da imagem, ele vai jogar a imagem para frente.

E depois o else if para inverter essa regra para conseguirmos voltar o sprite.

A divisão por 2 é para garantir que não demore para aparecer a próxima sprite.

**Iluminação**

Foi utilizado o “Universal RP” que se encontra no Package Manager para usar o Shader Graph.

O primeiro passo foi criar um Pipeline Asset, depois um “2D renderer” que é necessário colocar no “Pipeline Asset” na parte “Renderer List”.

E com isto, a gente consegue editar as configurações do projeto.

Edit > Project Settings > Graphics > Scriptable Render Pipeline Settings

Neste local é só adicionar o PipelineAsset criado anteriormente.

Em Sprite Renderer de todos os cenários ou objetos que você quer que tenha interação com a luz em Material coloquei o Sprite Lit Default. Depois disso adicionei um Global light.

Algumas coisas ficaram escuras e outas não, ai que deu problema no meu parallax e eu tive que refazer depois que fiz as luzes.

Arrumei a intensidade para nem tudo ficar escuro

Depois adicionei um Point Light para fazer luz na lanterna da personagem principal e deixei como filho do personagem para conseguir seguir.

Adicionei um Global Volume e dei um check no Post Processing para aparecer.

Em PipelineAsset também precisei dar um check no HDR.

Para um melhor brilho na lanterna, fiz um layer na animação em preto e branco e adicionei como segunda textura.

Depois disso, adicionei um Sprite Unlit Graph e um material para colocar no “Sprite renderer”.

Abri o Shader para configurar.  
Essa parte eu não sei se fiz direito, mas basicamente foi adicionar a primeira e a segunda textura e depois adicionar uma cor. Nessa parte de cor que dá para ajustar a intensidade do brilho do objeto.

Como não entendi o que aconteceu, eu acabei deletando essa parte e deixei só as luzes mesmo sem o shade.

**Escada**

O código da escada detecta basicamente a colisão do player e caso o jogador fizer uma movimentação vertical pra cima, ele vai pra cima ou se for pra baixo, ele vai pra baixo.

**Nadar**

Não foi necessário um script para isso porque o personagem afunda.

**Água**

Para a água foi utilizado o Buoyance Effect.

Alterei as configurações e deixei que só o objeto com a tag “ chao” fosse influenciada, já o jogador não tem efeito para ele realmente cair no fundo da água.

**Fase 01 - Kira Kira**

**Artes**

Personagem Principal - ok

Cenários - ok

Pena(feather) - ok

Bola de Fogo(espírito) - ok

Fogo(inimigo) - ok

Anjo(inimigo) - ok

Espinhos(armadilha) - ok

Água(armadilha) - ok

Escadas(interação) - ok

Porta(interação) - ok

**Desenvolvimento**

Movimentação do Player - ok

Movimentação Fogo - nok

Movimentação Anjo - nok

Ataque Anjo - nok

Acionamento do Buraco - nok

Acionamento da Porta - nok

Dano Fogo - nok

Dano Água - nok

Dano do Anjo - nok

Pontuação Bola de Fogo - ok

Pontuação Pena - ok

Interação com Escada - ok

Interação com a água - ok

Cena Morte - nok

Cena Vitória - nok

Parallax - ok

Iluminação - ok

**Sons**

Fundo - ok

coleta da pena - nok

coleta da bola de fogo - nok

Tudo que não está ok ficará para o desenvolvimento da próxima entrega e possivelmente haverá itens a serem acrescentados ao longo do jogo conforme haja necessidade.