

Programação 3D - Assignment III

Grupo 02

Francisco Campaniço 83463

João Rafael 83482

Rodrigo Oliveira 83558

Maio 2019

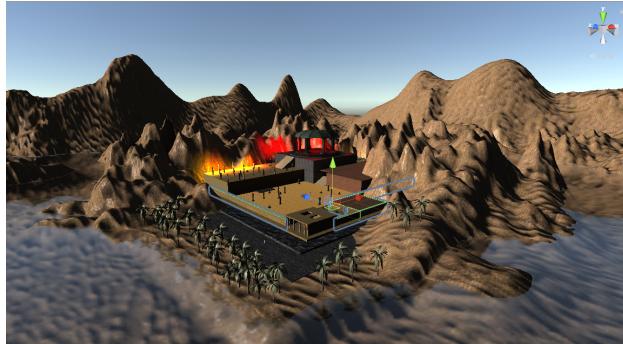


Figure 1: A cena de jogo.

Start / End Screens

Ao abrir o jogo, aparece um ecrã de início simples com um botão "start". Este ecrã trata-se de uma cena separada do jogo principal, e quando o jogador carrega no botão, a cena muda para a principal.

Ao acabar o jogo (quer por o jogador ficar com zero vida ou chegar ao fim do nível), aparece um ecrã de fim, indicando a pontuação do jogador num *leaderboard*.

HUD

O HUD mostra uma barra com a vida do jogador, um contador de munição da arma e um minimapa.

O minimapa é feito com uma câmara ortográfica a ver a posição do jogador de cima. A câmara envia a informação para uma *raw texture* situada no HUD.

Colisões

O jogador pode disparar sobre barris explosivos para causar uma explosão. Se houverem vários barris perto uns dos outros, dá-se uma explosão em cadeia. As explosões também repelem outros *rigidbodies* aos quais se aplicam físicas de jogo, e tiram metade da vida do jogador.

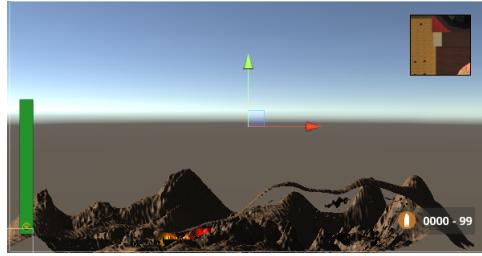


Figure 2: HUD.

O jogador pode pegar em objetos ao manter premida a roda do rato (*MouseButton3*). Esta funcionalidade usa o *raycasting* do Unity para verificar se o jogador está a apontar para um *rigidbody*.

Uma das salas do jogo está escura até o jogador passar por *triggers* escondidos na sala, que ativam *spotlights* (usando *OnTriggerEnter()*).

Há um "elevador" que, quando o jogador se posiciona em cima do mesmo, eleva-se até o jogador sair ou até a um limite dado por um *collider* invisível. (Usa-se *OnTriggerEnter()*, *OnTriggerStay()* e *OnTriggerExit()*.)

Existem *turrets* que disparam balas para o jogador. Estes seguem a posição do jogador, rodando na sua direção se ele se mexer.

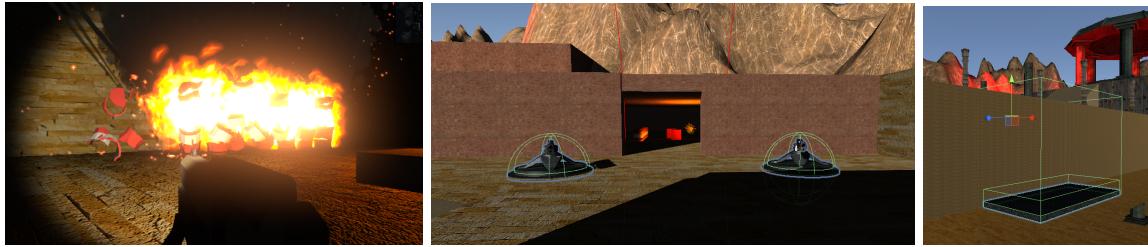


Figure 3: Barris explosivos, *turrets* e elevadores.

Mecânicas de jogo

O jogador tem uma arma com munições finitas, e 100 pontos de vida. Caso os pontos de vida cheguem a zero, o jogador morre e chega-se ao ecrã de fim. Estar perto de uma explosão tira 50 pontos de vida, e levar com uma bala ou ser atacado por um esqueleto tira 10 pontos. Se o jogador interagir com uma caixa de munições, ganha 100 balas extra para a arma.

Se chegar à "sala do tesouro", a pontuação do jogador duplica. A pontuação baseia-se no tempo desde o jogo começar.

Efeitos Visuais

O *bump mapping* é feito por aplicar uma textura normal (*normal map*) ao material de certos elementos de jogo.

As *lens flares* são aplicadas à *directional light*, para parecer como um "sol".

O *environment mapping* é feito através de uma reflection probe, cujo *environment map* é aplicado a uma esfera para demonstração.



Figure 4: *Lens flares, environment mapping.*

Usam-se vários sistemas de **partículas** do Unity, como vapor, fogo e explosões.

O Unity tem uma *stack* de pós-processamento incluída. Isto permite a adição de **ambient occlusion** e **HDR bloom** ao jogo. A *bloom* tem um *threshold* alto, e só é visível em luzes intensas (numa explosão, por exemplo). O componente HDR tem cor branca, aumentando a intensidade para branco quando há bloom no ecrã.



Figure 5: *Ambient Occlusion* ligado/desligado, *bloom*.

Iluminação Global

Para **realtime lighting/shadows** aplica-se a grande parte das luzes a propriedade "*realtime*". Isto inclui, por exemplo, o "sol", o que dá origem a uma espécie de ciclo dia-noite, dado que a luz direcional que representa um sol roda á volta do nível, afetando a posição das sombras no nível todo.

Para **baked lighting**, aplicou-se a uma point light vermelha no topo do nível a propriedade "*baked*", assegurando que a geometria à sua volta têm um *lightmap* aplicado que é computado no *engine*, antes da execução do jogo.

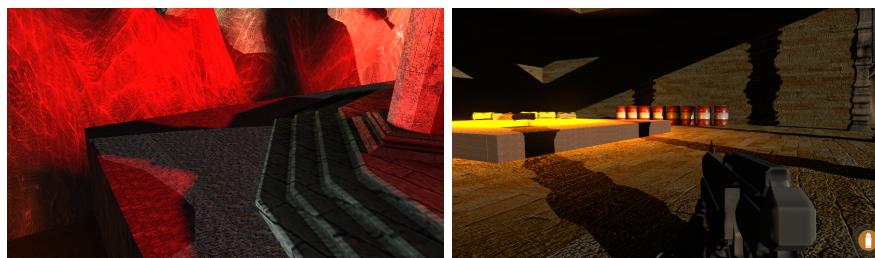


Figure 6: *Baked/realtime shadows.*

Extras

Dado o componente de Pós-processamento do Unity, também adicionámos *motion blur* e *color grading* ao jogo, tal como *depth of field*, que só se ativa quando o jogador carrega no botão direito do rato e olha pela mira da arma (simulando a falta de foco na arma em si). Também foi adicionado scripting à *bloom*: se o jogador perder vida, o *threshold* baixa e a intensidade aumenta temporariamente, como se a personagem tivesse sido ”atordoada”.



Figure 7: *Motion blur*, *depth of field* e *color grading* ligado/desligado.

Além de *turrets*, existem esqueletos que seguem o jogador e tentam atacá-lo. Estes esqueletos podem mexer-se por uma *navmesh*, que é pré-computada, e consiste das *meshes* que foram marcadas como ”navigation static”. Os esqueletos também tem as suas próprias barras de vida.

A ilha onde se passa o jogo está rodeada de água, que tem *screen-space reflections*.

Há uma parte do nível com esferas que se mexem sozinhas, seguindo uma animação.

Quando o jogador é atingido por uma bala, aparecem esferas vermelhas a simular ”sangue”.

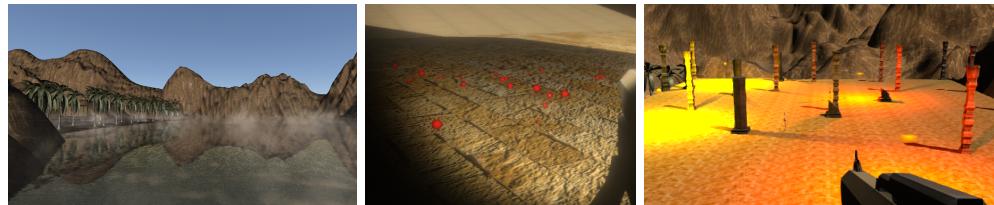


Figure 8: *Screen-space reflections*, esferas de ”sangue” e esferas em movimento.