

# Elaboração de um jogo com o tema de Tower Defense no Unity3D

Valmy Leôncio Ribeiro<sup>1</sup>

<sup>1</sup>Departamento de Ciência da Computação - Computação Gráfica  
Universidade Federal Rural do Rio de Janeiro (UFRRJ) – Rio de Janeiro, RJ – Brazil

{valmy}@ufrrj.br

**Abstract.** *Tower Defense game developed in Unity 3D for the Computer Graphics discipline. This article describes the development, mechanics, map, tower, enemies and important components of the game.*

**Resumo.** *Jogo Tower Defense desenvolvido no Unity 3D referente à disciplina de Computação Gráfica. Esse artigo descreve como foi realizado o desenvolvimento, as mecânicas, o mapa, a torre, inimigos e componentes importantes sobre o jogo.*

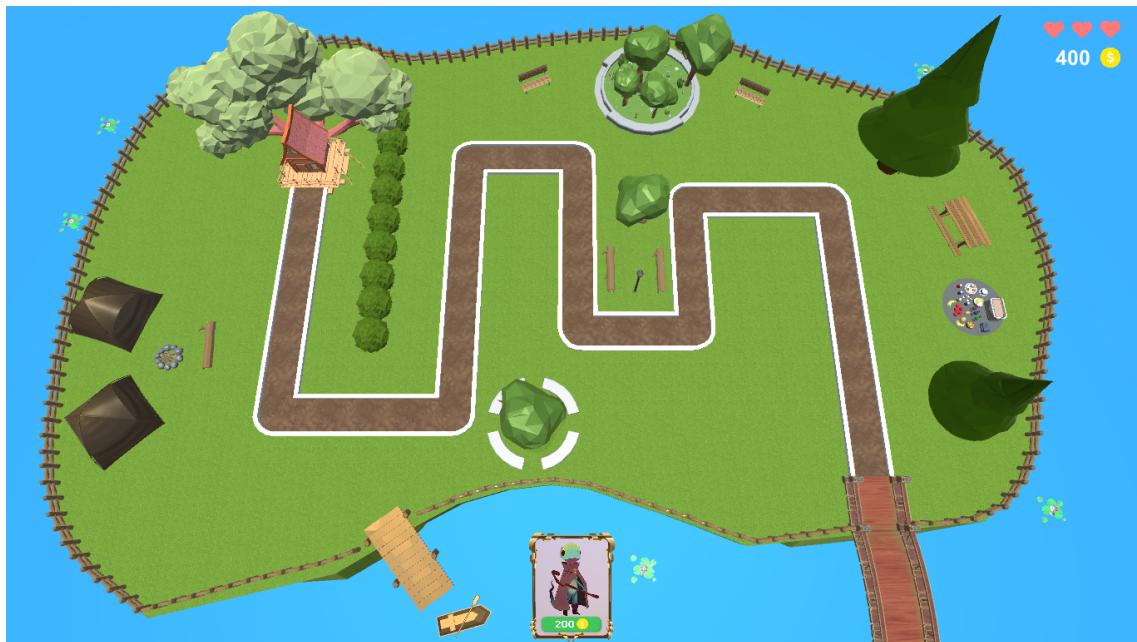
## 1. Introdução

O jogo foi desenvolvido sobre um tema e um cenário com objetos mais natural, onde a jogabilidade acontece em cima de uma ilha, onde animais que detém de poderes e habilidades, no jogo atual se tem apenas um, mas a ideia era oferecer mais possibilidades, tentam defender seu lar de uma invasão inimiga, que acontece por meio de uma ponte conectada à ilha. Sua estética conta com objetos modelados, low poly e animações. As interfaces de interações, como o GUI das torres, sistema de vida e dinheiro foram desenvolvidos no Figma e importado para o projeto. Alguns modelos foram feitos por mim, outros foram utilizando modelos e animações de terceiro para compor o jogo, pois o desenvolvimento desse jogo não se configura para fins lucrativos, apenas para aprendizagem e demonstrações.

## 2. Mapa

O mapa do jogo conta com diversos tipos de modelos de objeto, tanto low poly quanto modelado, por exemplo, com cores verdes vivas para se referir a um cenário mais natural. Alguns desses objetos, como a ilha, foram desenvolvidos pessoalmente, na plataforma de modelagem Blender. Os objetos em si não oferecem nenhum tipo de interação, foram posicionados unicamente para delimitar áreas nas quais o player não poderá posicionar suas defesas e para fins estéticos, apenas a ponte realiza uma função, servir como entrada de entidades, para simular uma invasão. O mapa apresenta um caminho de terra que é por onde as tropas/inimigos seguirão, por meio de waypoints que começa um pouco antes da ponte e está presente em todas as curvas do caminho, onde quando um chegar em um desses waypoints, ele realiza o cálculo para encontrar o waypoint mais próximo. O objetivo é impedir que as ondas de inimigos cheguem na casa da árvore, que para este jogo, simula a torre principal que deverá ser defendida. O

mapa tem certos locais definidos para o jogador posicionar suas torres. Uma vez posicionado, esse espaço não poderá receber outra torre.



**Figura 1. Mapa do Jogo**

### 3. Torres

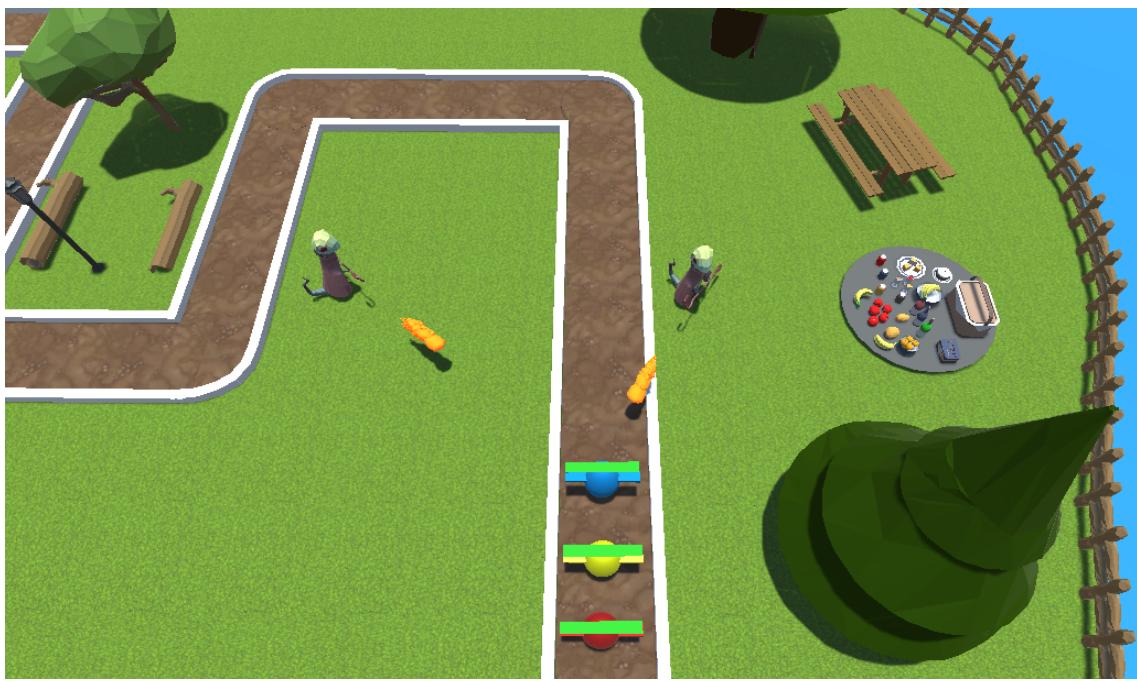
No momento atual do jogo, ele conta com apenas uma torre que o jogador poderá usar para se defender. É uma espécie de lagarto, da classe mago, onde eles disparam um aspecto flamejante animado em direção aos inimigos. As torres usam como target para realizar seus disparos inimigos que estiverem em seu raio de alcance, caso seja mais de um, ele dará foco no inimigo com índice de respawn mais baixo, ou seja, sejam instanciados dois inimigos, azul e vermelho, respectivamente, como o azul foi instanciado primeiramente, ele apresenta um índice de respawn mais baixo, pode-se entender também como uma espécie de contador. Para posicionar uma torre, o jogador deverá ter disponível 200 de gold, moeda do jogo. Essa torre não causa dano em área, apenas no inimigo em que elas estão focando. Caso o inimigo saia de seu alcance, a torre encontrará um inimigo com as características apresentadas acima. A torre também contará com um mecanismo de atualização, onde ficará mais robusta, aumentando seu dano, alcance e velocidade de disparo, com um custo de 150 gold.



**Figura 2. GUI da torre**



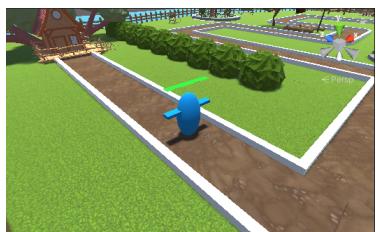
**Figura 3. Modelo da torre e upgrade**



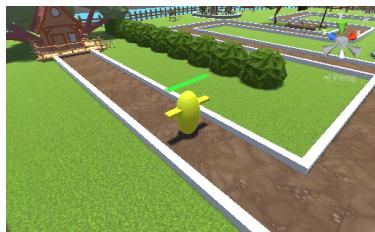
**Figura 4. Torres em ação**

#### 4. Inimigos

Existem duas classes de inimigos atualmente no jogo: os de onda e boss. Os inimigos de onda são os que serão instanciados para compor as hordas de inimigos, enquanto os da classe boss serão instanciados periodicamente ao longo do jogo. Existem três subclasses para os inimigos de onda: runner, tank e normal . Os da classe runner são mais rápidos, porém apresentam pouca vida, se identificam pela cor amarela. Os da classe tank são os que mais tem vida, porém tem pouca velocidade, se identificam pela cor vermelha. Os da classe tank são balanceados, tanto em questão de vida quanto de velocidade, se identificam pela cor azul. Já os bosses, são mais fortificados, são maiores e spawnam periodicamente, representando maior desafio na experiência do jogador. A manta, uma espécie de arraia voadora com um ar meio alienígena, na qual é animada similarmente como na vida real, e o Kermit, por fins de diversão, ele fica dançando ao longo do caminho. Esses dois são exemplos de bosses. Todos os inimigos presente no jogo contém uma barra de vida à mostra, indicando quanto faltam para serem derrotas, caso a barra de vida chegue a 0, os inimigos desaparecerão.



**Figura 5. Normal**



**Figura 6. Runner**



**Figura 7. Tank**



Figura 8. Manta



Figura 9. Kermit

## 5. Ondas de inimigos

Se configuram em um grupo de inimigos instanciados ao mesmo tempo no mapa. Sempre após 5 segundos, uma nova onda é instanciada. Os inimigos para compor as ondas são escolhidos aleatoriamente, podendo ser um runner, tank ou normal. A cada round, os inimigos ficam mais fortificados, tanto em questão de vida, quanto velocidade, aumentando assim a dificuldade. Sempre após 4 rodadas, as ondas passam a ter 1 inimigo a mais permanentemente. Sempre a cada 5 rodadas, 1 boss é colocado no mapa, sem nenhuma onda de inimigo junto com ele e novas ondas não poderão ser instanciadas até o boss ser eliminado ou se chocar com a casa da árvore .



Figura 10. Onda de inimigos e o boss Kermit

Percebe-se, que na Figura 5, nenhuma onda de inimigos foi instanciada após um chefão estar vivo no mapa.

## 6. Sistema de vida e ouro

O jogo também conta com um sistema de vida e gold, moeda do jogo, presente no canto superior direito da tela. Toda vez que um inimigo é abatido, 50 de gold é acrescentado para utilização na compra de uma nova torre ou realizar a atualização. O jogador tem 3 vidas, representada por 3 corações, toda vez que o jogador sofrer dano,

ou seja inimigo chegando na casa da árvore, será decrementado 1 coração, atualizando o sistema de vida. Um inimigo normal de onde causa 1 de dano ao jogador, e um boss causa 2 de dano. Caso o jogador esteja com 1 de vida e receba dano, uma janela de Game Over aparecerá ao jogador, oferecendo a opção de tentar novamente, ou menu (este ainda não funcionando).



Figura 11. Gold e jogador com 3 vidas



Figura 12. Gold e jogador com 1 vida

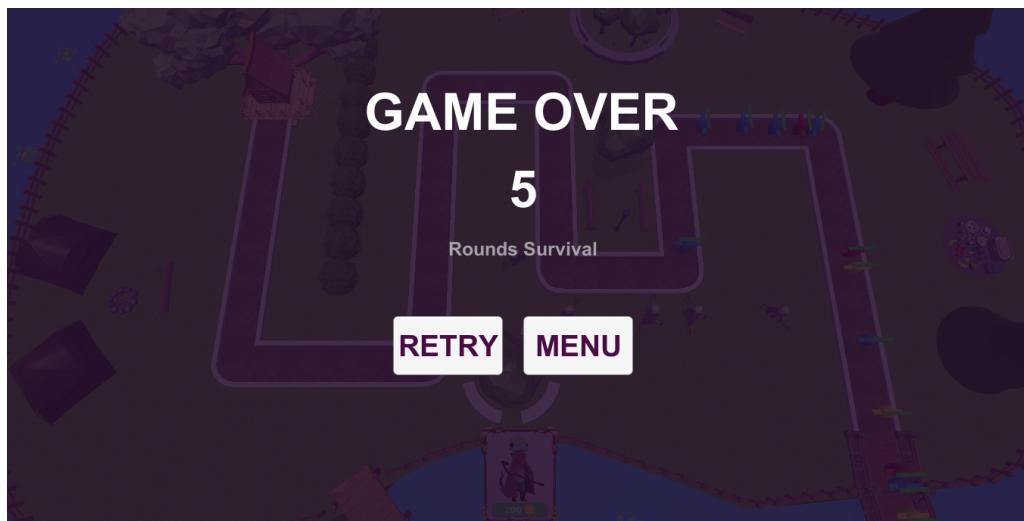


Figura 13. Tela de Game Over

## 7. Últimas considerações

Foi bem interessante desenvolver esse jogo, assim como conhecer melhor a engine Unity3D, ainda mais fazer um jogo conhecido do 2D, no geral, adaptado para 3D. Meu intuito era desenvolver e aprender novas práticas de computação gráfica, por isso me desafiei em tentar deixar o jogo com “minha” cara e expor criatividade, sem deixar com visuais e aspectos parecidos de tutoriais. Ao final foi bem divertido e aproveitável, talvez me influenciando um dia a ir mais fundo e experimentar o OpenGL para o desenvolvimento gráfico, apresentado em aula. Presenciei as principais dificuldades na criação de jogos, como por exemplo criar e manipular o design, a arte, juntamente com a parte dos códigos.