

Relatório do trabalho de Linguagem de Programação

Dilmar Aparecido Castanheiro

Bernardo Quintella

Matheus Constantin

December 2024

1 Introdução

O nosso projeto é um jogo do estilo Run'n Gun, inspirado em clássicos como *Contra* e *Metal Slug*, onde o personagem pode andar, pular, atirar para os lados, recarregar, morrer caso a vida acabe, e o seu objetivo é "derrotar todos os inimigos, ultrapassar os desafios da fase e alcançar o final". Ao atingir o bloco verde, o jogador deve pressionar para baixo, "encerrando a partida com sucesso".

2 Design

Nosso jogo possui uma única fase, onde existem 3 tipos de inimigos. O nível é composto por "cinco andares conectados por plataformas, criando desafios verticais e horizontais". O personagem deve descer os andares enfrentando obstáculos e inimigos.

Os inimigos são:

- AR: Portam um fuzil, "realizam patrulhas simples" e atiram quando veem o player.
- Sniper: Portam uma sniper, "são estrategicamente posicionados" e atiram quando veem o player.
- Bazooka: Portam uma bazooka, "representam grande ameaça com explosões" e atiram quando veem o player.

3 Gameplay e Mecânicas

O personagem possui movimentos básicos como andar, pular e atirar, além de mecânicas mais avançadas, como "recarregar a arma e utilizar estratégias para desviar de tiros inimigos". Os inimigos apresentam comportamentos distintos, "oferecendo variedade e aumentando a complexidade conforme o jogador avança". A fase é estruturada de forma linear, incentivando o jogador a descer andares e enfrentar inimigos.

4 Aspectos Visuais e Sonoros

Os gráficos do jogo seguem uma estética pixelada, devido aos sprites utilizados, "inspirando-se em jogos clássicos e criando um apelo nostálgico". Em relação ao som, há "efeitos sonoros dinâmicos que acompanham as ações, como tiros, pulos e impactos", aprimorando a imersão.

5 Divisão de Tarefas

Dividimos o desenvolvimento do jogo da seguinte forma: Matheus ficou responsável pelo menu e design, Dilmar fez o relatório, e cuidou do mapa e jogabilidade, Bernardo fez os testes, colisões e inimigos e todos participaram colaborativamente na implementação do código principal.

6 Organização do Código

O código está dividido em 10 arquivos principais dentro da pasta *src*, conforme detalhado a seguir:

- **entity.py**: contém a classe responsável pelas entidades do jogo (jogador, inimigos, bala).
- **player.py**: contém a classe responsável pelo jogador, com métodos para "executar ações como atirar, pular e recarregar".
- **enemy.py**: contém as classes responsáveis pelos inimigos, como o sniper, bazooka e AR.
- **bullet.py**: contém a classe responsável pela bala, que "gera dano ao colidir com alvos ou desaparece ao sair do cenário".
- **map.py**: contém o código responsável pelo mapa.
- **menu.py**: contém a classe responsável pelo menu.
- **settings.py**: contém as configurações principais, como "valores padrão para vida, munição e comportamento dos inimigos".
- **game.py**: cria o jogo a partir das classes contidas nos outros arquivos.
- **main.py**: executa o jogo.
- **util.py**: carrega algumas funções úteis.

7 Testes e Melhorias

Realizamos testes de jogabilidade para garantir que o jogo estivesse balanceado, identificando ajustes necessários em dificuldade, IA dos inimigos e responsividade dos controles. Além disso, recebemos feedback de "usuários externos que testaram o jogo", o que nos ajudou a refinar a experiência final.

8 Desafios Enfrentados

Durante o desenvolvimento, enfrentamos desafios como "implementar comportamentos inteligentes para os inimigos" e "resolver problemas de colisão das balas com precisão", mas conseguimos ultrapassar essas dificuldades.