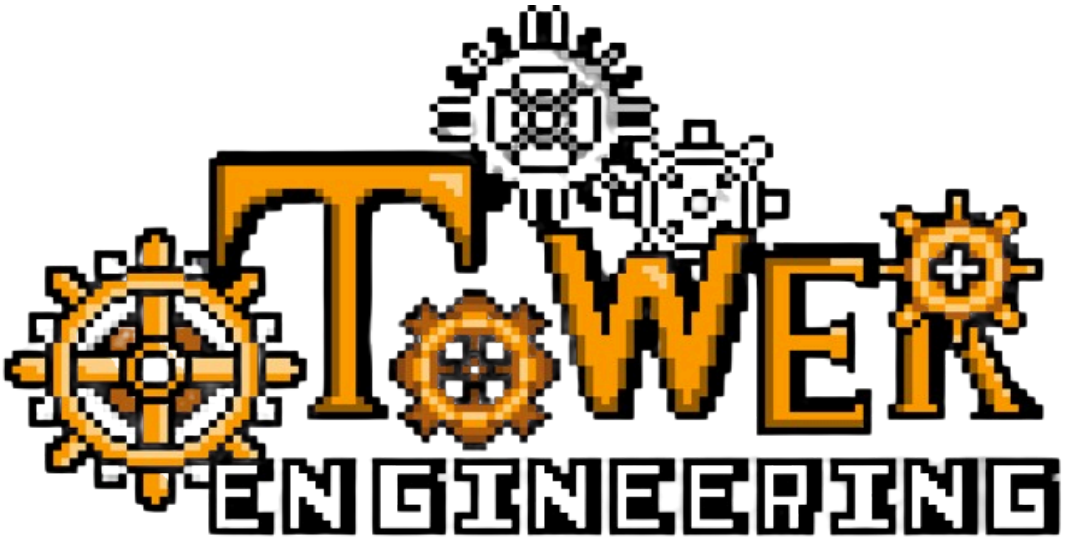
U.PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Laboratório de Computadores



Licenciatura em Engenharia Informática e Computação
2º Semestre - 2023/2024

Turma 5

Pedro Fonseca – up202108853

José Quintas - up202108712

Maria Laranjeira – up202004453

Manuel Silva – up202108874

Índice

1. Introdução

Tower Engineering é um jogo inspirado no género Tower Defense. No entanto, em vez de somente construir torres através de uma visão aérea, o nosso jogador apresenta uma influência direta e física no jogo.

Uma vez que o nosso jogo 2D de estratégia, com elementos de jogo RPG, nasceu de uma ideia completamente inovadora gostaríamos de o contextualizar para dar vida ao projeto.

A personagem principal é um engenheiro, cuja formação pode ou não ter sido na FEUP, com a principal missão de defender a sua base que está sob o ataque de uma rebelião de inimigos. No início do jogo, o engenheiro possui uma quantidade inicial de moedas, que pode gastar na loja para desbloquear torres ou na base para comprar armas. As torres são posicionadas em locais fixos ao longo do mapa, enquanto as armas podem ser usadas diretamente pelo jogador durante a defesa. Por cada inimigo eliminado o jogador adquire uma determinada quantia de dinheiro.

Para uma maior compreensão do jogo estilo Tower Defense, o jogo mais semelhante que conhecemos é o Bloons Tower Defense.

2. Instruções de Utilização do Programa

2.1. Menu Principal

No início do jogo temos o menu principal com vários botões que podemos escolher, cada um com uma funcionalidade diferente.

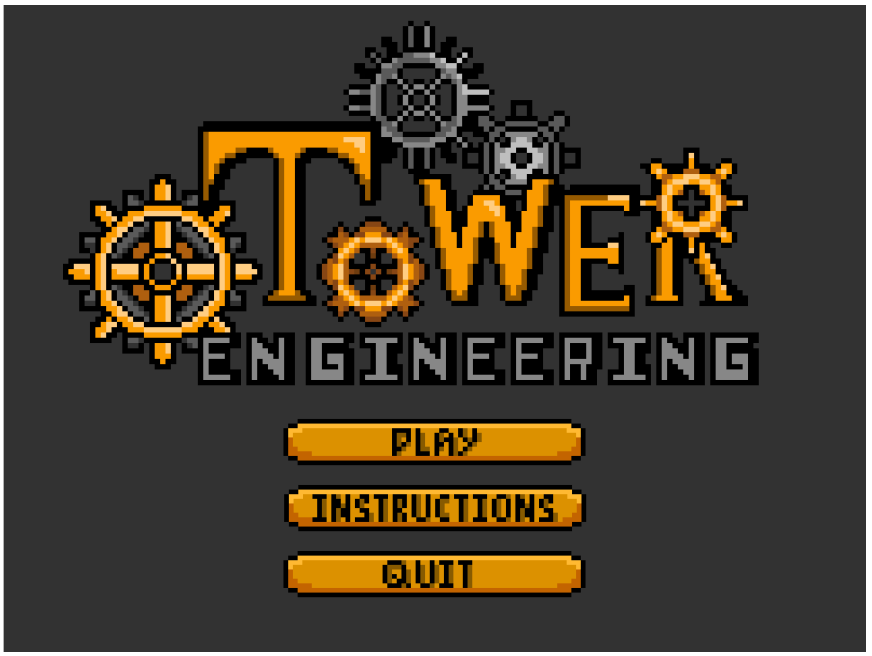


Figura 2.1 - Menu Principal

- PLAY: Leva o jogador a um segundo menu para selecionar o mapa de jogo.
- INSTRUCTIONS: Apresenta as instruções do jogo.
- QUIT: Permite-nos sair jogo.

Para selecionar uma das opções devemos usar o rato e pressionar no botão esquerdo.

2.2. Play

2.2.1 Menu de Seleção de Mapa

Ao clicar no botão PLAY somos redirecionados para um segundo menu de seleção de mapas. Cada mapa distingue-se principalmente pelo percurso dos inimigos.

Figura 2.2.1 - Menu de Seleção de Mapa

Após escolher um mapa e pressionar no botão PLAY do segundo menu começamos uma partida.

2.2.2 Mapa

Quando iniciamos o jogo é possível ver o mapa detalhadamente e os diferentes elementos que o constituem:

- Os inimigos e o seu percurso em direção à base.
- A personagem, que conseguimos movimentar com WASD.
- A loja que permite desbloquear as diferentes torres.
- Os diferentes locais de posicionamento de torres.
- O número inicial de moedas.
- A base onde o jogador pode comprar a sua arma pessoal

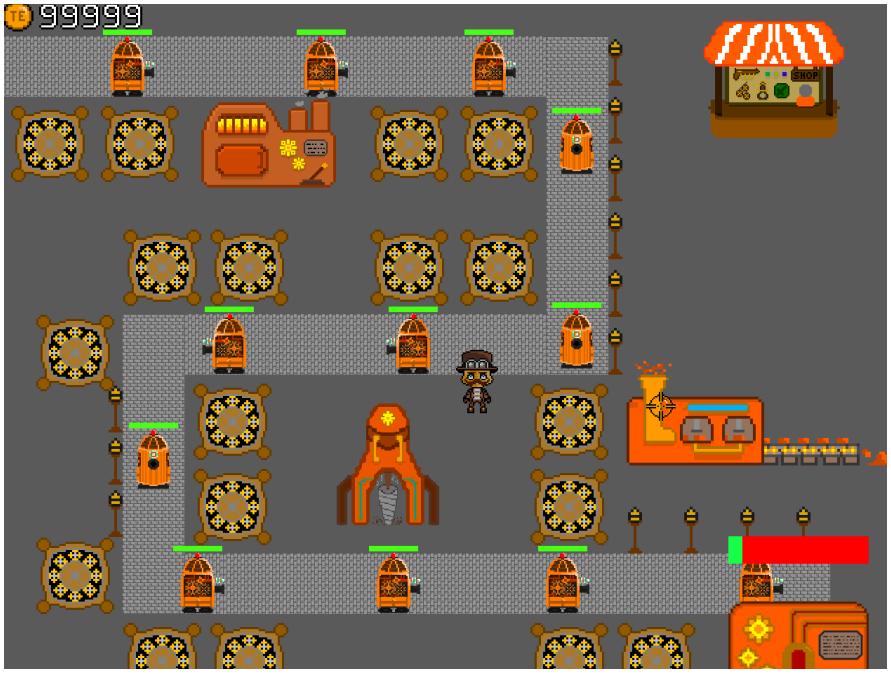


Figura 2.2.2 - Mapa do Jogo

O objetivo é defender a base dos inimigos através da plantação de torres estratégicas que podem ser desbloqueadas na loja ou do uso de armas adquiridas na base.

A base apresenta uma barra de vida que vai diminuindo conforme a chegada de inimigos Quando um inimigo colide na base, é removida dela a percentagem de vida que o inimigo tinha.

2.2.3 Menu de Pausa

Durante a partida, ao clicar em ESC o jogador é levado para o Menu de Pausa, onde pode decidir se quer continuar a partida ou voltar ao Menu Principal.

Figura 2.2.3 - Menu de Pausa

2.3. Instructions

Após clicar em INSTRUCTIONS no Menu Principal, somos levados a uma página com todas as informações necessárias para a compreensão do nosso jogo.

Figura 2.3 - Ecrã de Instruções

2.4. Game Over Menu

Após uma partida, conseguimos ver o resultado final. Se a base perder toda a sua vida, o jogador é automaticamente levado para o ecrã de derrota, fim do jogo, GAME OVER Menu.

Figura 2.4.1 - Ecrã de Fim do Jogo

3. Estado do Projeto

DISPOSITIVO	FUNCIONALIDADE	INTERRUPÇÕES
Timer	Animações, Frames per Second, Lógica do Jogo.	Sim
Keyboard	Navegar nos menus, controlar o movimento da personagem.	Sim
Mouse	Navegar nos menus. Seleção de torres in-game, acesso e navegação na loja, mirar o disparo in-game.	Sim
Graphics Card	Apresentar as interfaces do jogo e todos os sprites associados.	Não
Real Time Clock	Controlar a frequência de spawn dos inimigos.	Sim

3.1. Timer

O timer é crucial para o desenvolvimento do jogo. É usado tanto para FPS quanto para ticks de física por segundo. Isto significa que a jogabilidade não está ligada aos valores de FPS, que podem variar, mas tem uma taxa fixa dentro de um segundo. Isto permite uma melhor gestão possível com as diferentes funcionalidades. Por fim, o timer é usado para realizar comandos de UI "em movimento" que podem aparecer. O movimento deles estará ligado à taxa de ticks de física, pois esta é uma forma constante de medir o tempo dentro do motor do jogo.

Funções Relevantes:

- Funções do ficheiro timer.c

3.2. Keyboard

Sendo um dos principais dispositivos de um computador, o papel do teclado é crucial neste jogo. Ele proporciona uma forma de navegar por vários menus, controlar o movimento das personagens e plantar torres.

Funções Relevantes:

- Funções do ficheiro kbc.c, keyboardkeys.c

3.3. Mouse

O papel do rato é semelhante ao do teclado no contexto de um jogo. É essencial para ajudar a navegar nos menus e disparar e mirar as balas para os inimigos.

Funções Relevantes:

- Funções do ficheiro mouse.c

3.4. Graphics Card

A placa gráfica tem um papel claro e simples. Proporciona a interação da UI para o jogador.

Funções Relevantes:

- Funções do ficheiro graphics.c

3.5. RTC

Os alarmes do RTC são usados para tempos de cooldown do spawnrate dos inimigos. Os interrupts do rtc estão configurados para se realizarem de meio em meio segundo sendo que um inimigo é criado a cada duas interrupções.

Funções Relevantes:

- Funções do ficheiro rtc.c

4 Organização e Estrutura do Código

4.1. Timer Module

O módulo Timer define as rotinas de configuração e manipulação do timer, incluindo a inicialização, configuração da frequência e interrupções do temporizador.

Weight: 1%

4.2. Kbc Module

O módulo Kbc fornece funções para inicializar o controlador de teclado, configurar interrupções e ler dados da entrada do teclado. É crucial para a interação com o teclado, capturando entradas do usuário e traduzindo em eventos que podem ser utilizados pelo sistema.

Weight: 1%

4.3. Keyboardkeys Module

O módulo Keyboardkeys lida com a interpretação das teclas pressionadas no teclado. Ele define funções para mapear *scancodes* em ações específicas ou caracteres, facilitando a tradução das entradas do teclado em comandos utilizados pela aplicação.

Weight: 1%

4.4. Mouse Module

O módulo Mouse gere a interface e o controle do mouse. Inclui funcionalidades para inicializar o dispositivo, configurar interrupções e processar dados de movimento e cliques do mouse. Este módulo é essencial para a captura e interpretação das ações do usuário realizadas com o mouse.

Weight: 1%

4.5. Graphics Module

O módulo Graphics inclui funções para inicializar o modo gráfico, desenhar formas e imagens na tela, e manipular buffer de vídeo. Este módulo é fundamental para a renderização visual, permitindo a criação e exibição de interfaces gráficas e elementos visuais do jogo.

Weight: 1%

4.6. RTC Module

O módulo RTC gere o relógio de tempo real (Real-Time Clock) do sistema. Fornece funções para inicializar o RTC, configurar alarmes e ler a data e hora atuais.

Weight: 1 %

4.7. Utils Module

O módulo Utils contém funções utilitárias usadas por outros módulos do sistema. Estas funções incluem operações básicas e repetitivas, facilitando a reutilização de código e simplificando a implementação de funcionalidades mais complexas.

Weight: 1 %

4.8. Device_controller Module

O módulo Device_controller fornece uma interface unificada para inicializar e controlar dispositivos como teclado, mouse, timer e RTC, garantindo que todos funcionem em harmonia e de forma eficiente.

Weight: 1 %

4.9. Game Module

O módulo Game define a estrutura principal do jogo. Gere os estados do jogo, como inicialização, execução, pausa e fecho, além de coordenar a interação entre diferentes componentes do jogo, incluindo gráficos, entrada do usuário e lógica de jogo.

Weight: 1 %

4.10. Menu Module

O módulo Menu implementa a interface do menu do jogo. Define funções para criar, exibir e interagir com os menus, permitindo que os jogadores naveguem entre diferentes opções e configurações do jogo.

Weight: 1 %

4.11. Instructions Module

O módulo Instructions fornece o ecrã de instruções do jogo. Exibe informações sobre como jogar, controles e objetivos, ajudando os jogadores a entenderem as mecânicas e regras do jogo.

Weight: 1%

4.12. Gameover Module

O módulo Gameover gere o ecrã de fim de jogo. Exibe opções para reiniciar o jogo ou voltar ao menu principal, permitindo o fecho adequado após o fim de um jogo.

Weight: 1 %

4.13. Button Module

O módulo Button define a estrutura e comportamento dos botões na interface gráfica do jogo. Inclui funções para criar, exibir e detectar interações com botões, facilitando a navegação e interação do usuário com a interface.

Weight: 6 %

4.14. Arena Module

O módulo Arena gere a lógica e a representação da arena de jogo. Define a estrutura da arena, controla a disposição dos elementos e trata a interação entre o jogador e o ambiente, essencial para a jogabilidade.

Weight: 5 %

4.15. Bullet Module

O módulo Bullet gere a lógica dos tiros no jogo. Inclui funções para criar, mover e detectar colisões de tiros, sendo crucial para a implementação de mecânicas de combate.

Weight: 5 %

4.16. Enemy Module

O módulo Enemy define a lógica dos inimigos no jogo. Gere a criação, comportamento e interação dos inimigos com o jogador.

Weight: 5 %

4.17. Money Module

O módulo Money gere a lógica de dinheiro no jogo. Inclui funções para acumular, gastar e exibir a quantidade de dinheiro do jogador, fundamental para mecânicas de compra e upgrade.

Weight: 5 %

4.18. Player Module

O módulo Player define a lógica do personagem do jogador. Gere os atributos, movimentos, e interações do jogador com o ambiente e outros elementos do jogo, sendo central para a experiência do usuário.

Weight: 5 %

4.19. Playerbase Module

O módulo Playerbase gere a base do jogador no jogo. Possui uma vida total que diminui a cada inimigo que consiga escapar a todas as torres e ao jogador vivo e complete o seu caminho.

Weight: 3 %

4.20. Shop Module

O módulo Shop implementa a lógica da loja no jogo. Permite que os jogadores desloqueiem novas towers mais poderosas que poderão montar nas bases espalhadas pela arena/mapa.

Weight: 3 %

4.21. Towers Module

O módulo Towers gere a lógica das torres de defesa no jogo. Inclui funções para as montar e desmontar, mudar o raio do alcance da torre, dar upgrade à torre incrementando o seu dano e alterar o tipo de targetting da torre (primeiro inimigo na fila, último inimigo na fila, inimigo mais perto da torre)

Weight: 7 %

4.22. Weapons Module

O módulo Weapons define a lógica das armas no jogo. Gere a criação, uso e efeitos das armas, sendo fundamental para o combate e a estratégia do jogador.

Weight: 2 %

4.23. Economy Module

O módulo Economy lida com a economia do jogo. Encapsula todos os custos presentes no jogo, quer na loja, quer na base, quer em cada torre.

Weight: 3 %

4.24. Gameplay Module

O módulo Gameplay coordena a lógica principal do jogo. Integra diversos aspectos do jogo, como combate, economia e deteção de colisão.

Weight: 30%

4.25. Gameobject Module

O módulo Gameobject define a estrutura e as interações dos objetos do jogo. Fornece uma base para a criação e manipulação de diversos elementos do jogo, como personagens, tiros e estruturas.

Weight: 10 %

4.26. Sprite Module

O módulo Sprite gere a renderização e animação de sprites no jogo. Inclui funções para carregar, exibir e animar imagens.

Weight: 10 %

4.27. Proj Module

O módulo Proj é o ponto de entrada principal do projeto. Coordena a inicialização do sistema, a configuração dos diversos módulos e o loop principal do jogo, sendo essencial para o funcionamento geral da aplicação.

Weight: 1 %

4.28. Function Call Graph

Figura 4 - Imagem do Module Graph gerado pelo Doxygen

5. Detalhes de Implementação

5.1. Real Time Counter (RTC)

Os alarmes do RTC foram usados para os tempos de cooldown do spawnrate dos inimigos. Os interrupts do rtc estão configurados para se realizarem de meio em meio segundo sendo que um inimigo é criado a cada duas interrupções.

5.2. Deteção de Colisões

De modo a permitir que as personagens não saiam do ecrã e evitar que subam objetos, como o caso da loja, tivemos de implementar deteção de colisões. No nosso caso, utilizamos if-statements. Caso o jogador se encontre dentro de determinados valores de x e y não se consegue movimentar para esses locais.

Também temos presente deteção de colisões quando uma bala colide num inimigo para causar dano, onde todos os inimigos e todas as balas são analisadas para verificar se se encontram nos mesmo blocos de interseção.

5.3. Animações

Durante o jogo, temos presentes imensas animações. O sprite dos jogadores e dos inimigos alteram em movimento e conforme a direção. As torres alteram a sua rotação conforme a posição dos inimigos. O nosso mapa está em constante movimento para permitir uma maior imersão e realismo.

6. Conclusões

Com muita satisfação, conseguimos concluir este trabalho com o rigor esperado. Apesar de não termos cumprido com o desenvolvimento do Serial-Port, desenvolvemos um jogo apelativo e divertido. Muitas das ideias formuladas no início do projeto foram concretizadas, o que permitiu que conseguíssemos desenvolver um jogo com bastantes detalhes e bastante criatividade.

É importante afirmar que produzir um jogo desta dimensão em C não foi uma tarefa fácil, uma vez que foi necessário compreender de raíz o funcionamento do software, o que dificultou o debug. No entanto, este projeto tornou-se um desafio bastante recompensador, não só pela aprendizagem que adquirimos, mas também pela qualidade e solidez final do nosso jogo, tanto a nível tecnológico como criativo.

Estamos orgulhosos do resultado alcançado e acreditamos que este jogo reflete o nosso empenho e dedicação.