

Especificação de requisitos

Projeto Tower Capture

Versão: 1.1

Data: 18/05/2022

| Versão | Autor(es) | Data | Ação |
|--------|---------------------------------|------------|---|
| 1.0 | Leonardo Nunes Matheus Bueno | 02/05/2022 | Definição dos requisitos do projeto |
| 1.1 | Leonardo Nunes Matheus Bueno | 18/05/2022 | Refinamento dos casos de uso do projeto |

Conteúdo:

1. Introdução
2. Visão geral do sistema
3. Requisitos de software

1.Introdução

1.1. Objetivo do desenvolvimento

Desenvolvimento de um programa de computador que implementa um jogo chamado "Tower Capture" entre dois jogadores.

1.2. Especificação do jogo

O jogo consiste em um mapa contendo uma torre, um arqueiro, um escudeiro e um guerreiro para cada jogador, onde se inicia um de cada lado do mapa. O objetivo do jogo é fazer com que a torre seja destruída ou que todos personagens do jogador sejam eliminados. Para isso, o jogador poderá selecionar um dos seus personagens (arqueiro, escudeiro e guerreiro) a cada turno para caminhar e atacar seus inimigos e/ou a torre inimiga, sendo que cada classe de personagens possui alcance de movimentação e de ataque diferentes entre si.

Ao abrir o jogo, é apresentado a tela inicial com o nome do jogo e um botão para iniciar a partida:

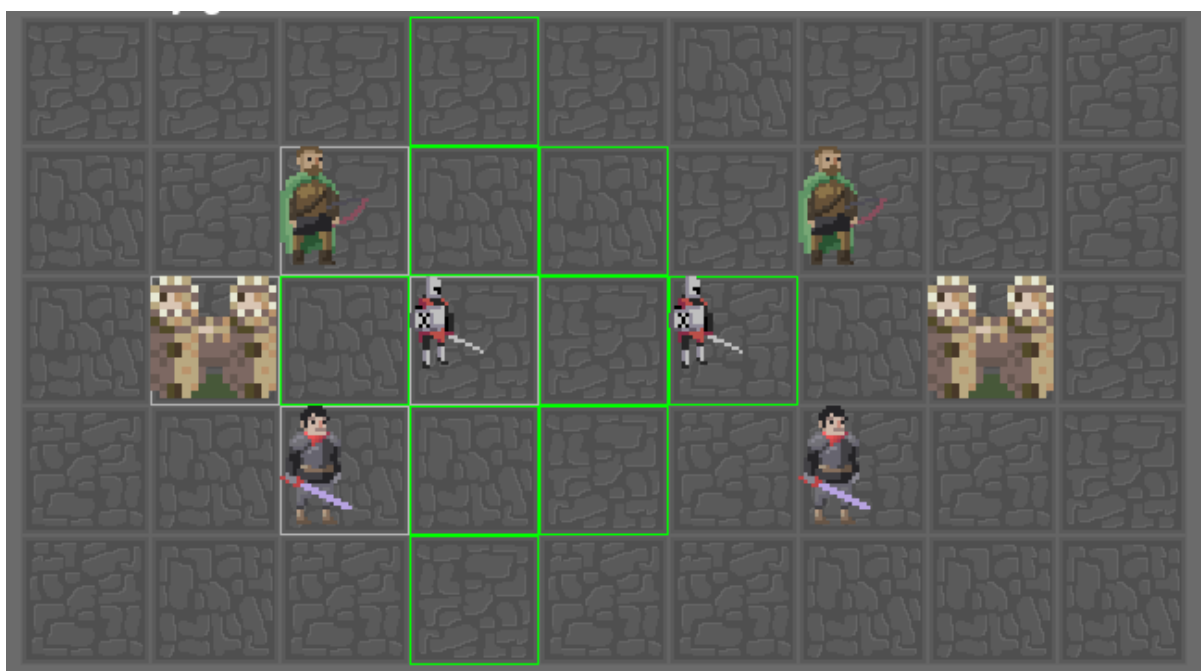
Tower Capture

Iniciar Partida

Após clicar no botão de Iniciar Partida, o jogo inicia com 3 personagens e uma torre para cada jogador, um à esquerda e outro à direita. O jogador da esquerda sempre inicia primeiro.



Quando for a sua vez, o jogador deve selecionar o seu personagem, se ele clicar em um personagem válido, então será aberto um range de opções para que ele possa andar com aquele personagem conforme exemplo abaixo do personagem a esquerda após ter sido selecionado:

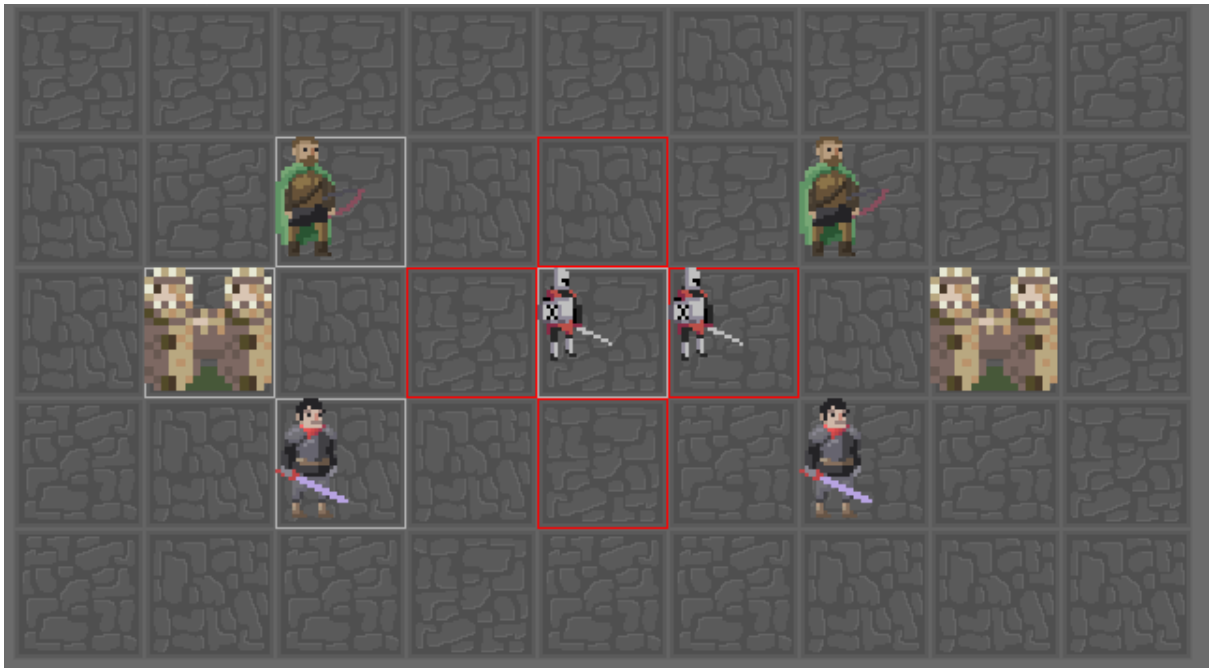


Caso o jogador tenha se arrependido de selecionar aquele personagem, pode-se clicar mais uma vez neste mesmo personagem para desselecioná-lo e poder escolher outro personagem, também é OBRIGATÓRIO que em cada turno, um dos personagens do jogador ande para outra posição.

Para cada tipo de personagem, existe uma variação no alcance de ataque e de movimentação, bem como a vida e a defesa que são apresentados na tabela abaixo:

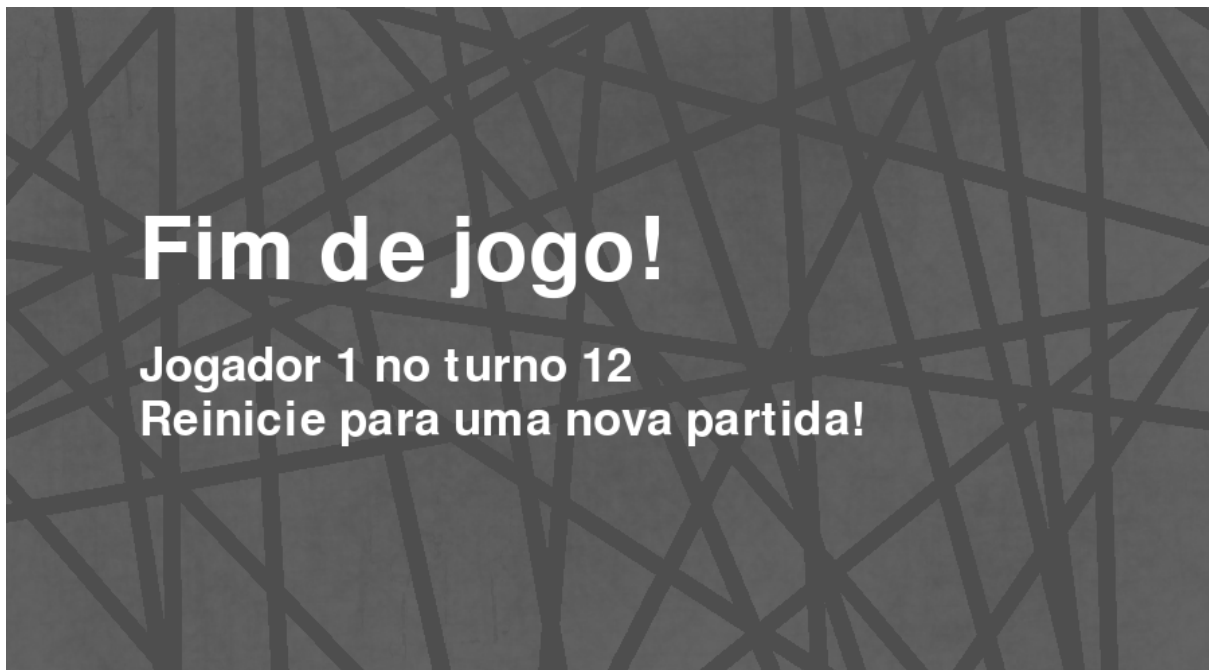
| Tipo | Vida | Ataque | Defesa | Range ataque | Range movimentação |
|-----------|------|--------|--------|--------------|--------------------|
| Arqueiro | 20 | 15 | 10 | 3 | 2 |
| Guerreiro | 25 | 20 | 10 | 1 | 3 |
| Escudeiro | 30 | 10 | 30 | 1 | 2 |
| Torre | 100 | 0 | 0 | 0 | 0 |

Após andar, o personagem pode escolher entre atacar outro personagem ou passar o seu turno para o outro jogador, sendo assim, o ataque é OPCIONAL, dependendo da estratégia de cada jogador em determinada situação.



Se o jogador atacar outro personagem, é então passado a jogada para o jogador da esquerda e o contador de turnos é atualizado.

Quando um jogador eliminar a torre inimiga ou matar o último inimigo, é então finalizado o jogo e apresentando a tela de fim da partida, mostrando o número do jogador vencedor e o número do turno em que a partida foi finalizada.



2. Visão geral do sistema

2.1. Arquitetura do software

- Centralizado (stand alone)

2.2. Premissas de desenvolvimento

- Deverá ser implementado na linguagem de programação Python
- Deverá utilizar a interface gráfica 2D utilizando a biblioteca Tkinter
- A apresentação do jogo será para dois jogadores utilizando a mesma interface gráfica
- A modelagem do desenvolvimento deve utilizar a linguagem UML 2.

3. Requisitos da aplicação

3.1. Requisitos funcionais

[RF1] Iniciar partida

What: Corresponde a apresentação da tela para iniciar um novo jogo, bem como os botões para inicializar a partida ou finalizar o programa.

Why: Este requisito é necessário para que, ao final de uma partida, o jogo reinicie e dê as opções para os jogadores de iniciar uma nova partida ou finalizar o programa.

Who: Qualquer um dos dois jogadores pode iniciar uma nova partida.

When: Ao abrir o programa ou ao final de uma partida.

Where: Através dos botões de (“INICIAR”) ou (“SAIR”).

How: A definição de quem será o jogador com a torre do lado esquerdo e a do lado direito será decidido entre os jogadores antes da inicialização da partida. Tendo isso definido, os jogadores irão alternando a medida que o turno for dado para o jogador da esquerda ou o da direita.

[RF2] Selecionar personagem

What: Consiste na etapa de seleção de qual personagem o jogador vai utilizar.

Why: Esta é uma das ações que cada jogador deve fazer para concluir a sua jogada.

Who: Será realizado por um dos jogadores quando for a sua vez.

When: Caso seja o início da partida, será o jogador da esquerda, ou, caso a partida ainda esteja em andamento, será após a finalização da jogada do jogador anterior

Where: Será na posição no mapa onde está o personagem que o jogador deseja selecionar. Além disso, será apresentado na lateral esquerda, acima do botão de passar o turno, os status daquele personagem selecionado, tal como vida, defesa, movimentação e ataque.

How: Através de um clique do mouse sobre o personagem. Caso o jogador se arrependa de ter selecionado aquele personagem, ao clicar novamente com o mouse nele, o personagem é desselecionado e o jogador deve escolher novamente outro personagem.

[RF3] Selecionar posição andar

What: Consiste no procedimento de seleção da posição para o personagem selecionado andar a partir das posições disponíveis para ele no mapa.

Why: Para que o personagem possa se aproximar dos personagens inimigos e da torre inimiga. Bem como permitir traçar estratégias para o jogador proteger a sua própria torre. Esta também é uma ação necessária para que cada jogador possa finalizar sua jogada.

Who: Será realizado por um dos jogadores quando for a sua vez.

When: Após realizada a seleção de personagem.

Where: Em uma posição válida do mapa.

How: Após selecionado uma posição válida para andar, que será destacada em cor verde, o personagem irá caminhar até esta posição.

[RF4] Selecionar posição atacar

What: Selecionar uma posição ao alcance do personagem onde reside um inimigo no mapa.

Why: A partir da interação de ataques entre os personagens e a torre, será definido quem é o vencedor da partida.

Who: Será realizado por um dos jogadores quando for a sua vez.

When: Após o personagem ter finalizado a operação de andar. Vale ressaltar que o jogador pode escolher não selecionar posição para atacar e simplesmente passar o seu turno para o próximo jogador.

Where: Na posição do inimigo selecionado

How: Após selecionado a posição onde reside o inimigo, destacada em vermelho, o personagem irá desferir um ataque que irá reduzir o nível de vida do inimigo na mesma proporção do seu poder de ataque para aquele inimigo.

[RF5] Passar turno

What: Consiste em realizar a ação de passar o turno para que o próximo jogador execute a sua jogada.

Why: Foi pensado na lógica de turnos para que os jogadores pudessem pensar qual jogada poderiam fazer, semelhante a ideia de jogos de tabuleiro.

Who: O jogador que finalizou a sua jogada.

When: Ao final de uma jogada.

Where: Através do botão ("PASSAR TURNO") localizado na lateral esquerda da tela.

How: Ao final de uma jogada, é então clicado para passar o turno para o próximo jogador.

[RF6] Finalizar partida

What: Apresenta a tela final e anuncia o vencedor, habilitando reiniciar a partida ou encerrar o programa.

Why: Para apresentar o jogador vencedor e a quantidade de turnos que a partida durou.

Who: O jogador que ganhou o jogo.

When: Ao final de uma jogada onde o jogador vencedor destruiu a torre ou a última entidade do oponente.

Where: Através da tela de encerramento.

How: Ao final de uma jogada que decide o fim da partida, é então mostrada a tela de fim de jogo, anunciando o jogador vencedor. O usuário final então tem a opção de iniciar uma nova partida clicando na tela ou de encerrar o programa fechando-o.

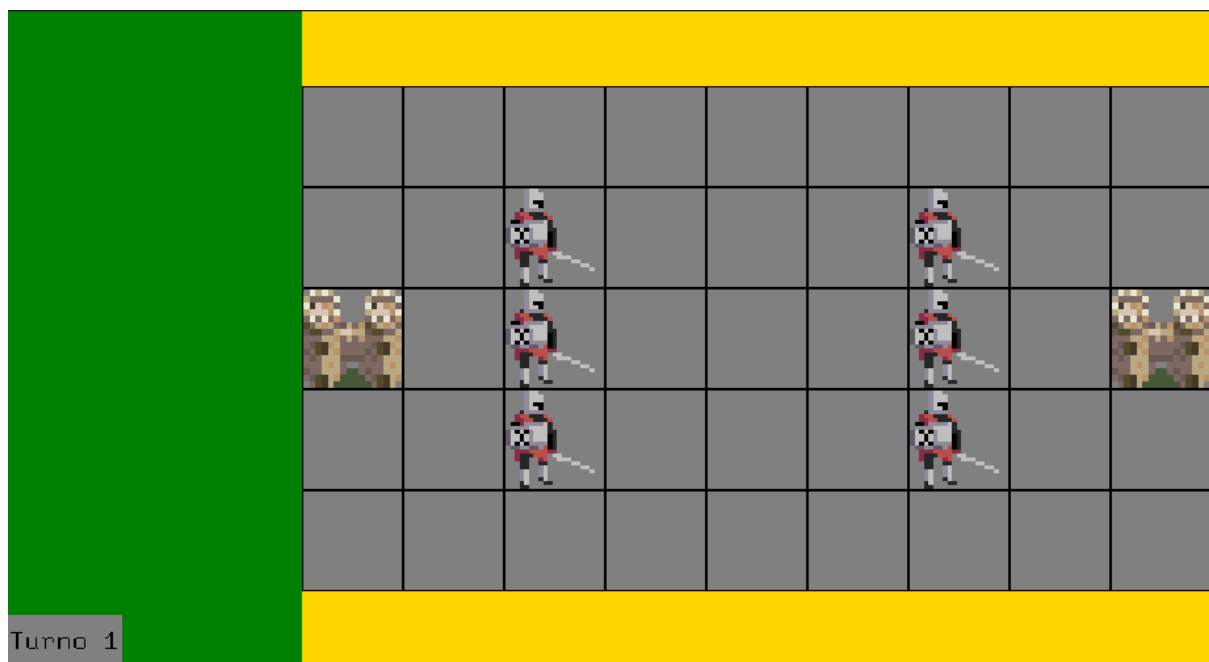
3.2. Requisitos não funcionais

[RN1] Implementação

O jogo será implementado utilizando a versão 3.9 da linguagem de programação Python e utilizando a biblioteca Tkinter para a criação da interface gráfica.

[RN2] Interface gráfica

Apresentação visual da implementação do protótipo utilizando a biblioteca python: Tkinter.



Tela de início da partida

[RN3] Tela simultânea

O programa necessita que dois jogadores interajam utilizando a mesma interface gráfica.