

## Engenharia Elétrica – Programação Orientada Objetos

### Trabalho 2 - Bomberman

#### Informações Gerais

- **Data de Entrega:** 29/11/2023 (23:59)
- **Método de Entrega:** Um arquivo zip contendo os códigos dos trabalhos devem ser enviados até a data limite para o e-mail [filipe.mutz@ufes.br](mailto:filipe.mutz@ufes.br). O título do e-mail deve ser “EE – POO – TRAB 2 – [nomes completos dos integrantes]” e o corpo do e-mail deve conter os nomes e números de matrículas dos alunos.
- Os trabalhos deverão ser feitos em duplas.
- **Entregas além da data limite ou entregues em formato diferente do especificado acima não serão corrigidos.**
- **Trabalhos que resultarem em erro de execução receberão nota zero. Se um trecho de código não estiver funcionando até a data de entrega, comente o trecho ou remova. Só receberão pontuação as funcionalidades que estiverem funcionando.**

#### Descrição

O trabalho consiste no desenvolvimento de uma réplica simplificada do jogo Bomberman usando a biblioteca pygame. Ao executar o jogo, deve ser exibida uma cena em que o jogador poderá escolher realizar a partida com 1 ou 2 jogadores. A Figura 1 ilustra a cena da partida. Se ela for realizada com um jogador, este deve iniciar no canto esquerdo superior. Se forem dois jogadores, o segundo deverá iniciar no canto direito inferior. A partida acontece em um mapa retangular com paredes ao redor. Além do muro exterior, no mapa existem também muros que são indestrutíveis e outros que são destrutíveis por bombas. Estes muros são criados em posições aleatórias e devem ocupar aproximadamente 70% do mapa. No meio do mapa existe o quartel general dos inimigos a partir do qual novos inimigos surgem a cada 2 segundos. A partida tem uma duração pré-definida de 2 minutos e termina se o tempo acabar, se a estrutura central dos inimigos for destruída ou se os jogadores morrerem. Ao final da partida, deve ser executada uma cena que informa se o jogador venceu ou perdeu e a pontuação dos jogadores. Mais detalhes sobre os componentes do jogo são dados a seguir.



Figura 1: Exemplo de início de partida.

**Personagem, modo de controle e movimentação:** O personagem do jogador 1 deve ser controlado usando as teclas ASDW para movimentação e espaço para liberar bombas. O personagem do jogador 2 deve ser controlado usando as teclas direcionais ( ←↑→↓ ) e O para liberar bombas. Os personagens podem colocar até 4 bombas simultaneamente em diferentes locais do ambiente. À medida que as bombas forem explodindo, o jogador ganha novamente o direito de liberar novas bombas. Isto é, se o jogador liberou 2 bombas no ambiente, ele pode liberar mais duas. Se uma das duas iniciais explodir, ele terá o direito de liberar 3 bombas.

**Bombas e seus efeitos:** Assim que são liberadas, bombas ficam em um local por um período de 2 segundos e então explodem. Ao explodir, elas geram chamas em formato de cruz que, por padrão atingem a célula central e 5 células para a esquerda, para a direita, para cima e para baixo. As chamas podem atingir objetos do cenário (muros, etc.) e personagens (jogadores, inimigos e o quartel). Ao atingir objetos do jogo, as chamas param de se propagar (e.g., se existia um inimigo na célula à direita da posição da bomba, a chama irá apenas até esta casa e não continuará por 5 casas). Chamas destroem imediatamente muros destrutíveis, o personagem do jogador e inimigos. O quartel dos inimigos tem 20 pontos de vida e, portanto, deve ser atingido por chamas 20 vezes para ser destruído.

**Inimigos e magias:** Inimigos se movimentam em linha reta até tocar um obstáculo, quando passam a se movimentar em uma direção aleatória entre esquerda, direita, para cima e para baixo. Os personagens dos jogadores morrem se forem tocados por inimigos. Inimigos podem ser de dois tipos:

- Um deles é um fantasma capaz de andar sobre bombas e sobre muros destrutíveis. O fantasma possui uma aura que altera o tempo. Ao ser criado, é escolhido aleatoriamente se o fantasma acelera ou atrasa o tempo. Quando jogadores ou bombas estão em um raio de 10 células do fantasma, as velocidades de movimentação dos jogadores e de explosão das bombas aumentam ou diminuem em 3x dependendo do tipo de aura do fantasma. A aura do fantasma deve ser exibida como um círculo cinza na tela do jogo.
- O segundo tipo de inimigo é um alienígena capaz de lançar projéteis. Os projéteis se movimentam em linha reta até atingir um muro indestrutível, ou uma bomba ou o personagem do jogador. Os projéteis voam sobre muros destrutíveis ou sobre outros inimigos sem atingi-los. Ao colidir com uma bomba, o projétil a faz explodir imediatamente. Ao colidir com o personagem, ele causa a morte do personagem. Projéteis são lançados a cada 2s em uma das quatro direções aleatórias (para cima, para baixo, para a esquerda ou para a direita).

**Pontuação:** Os jogadores ganham 1 ponto sempre que destruírem um muro, 10 pontos por matar um inimigo e 100 pontos por destruir o quartel general dos inimigos.

### **Classes Obrigatórias**

No trabalho devem ser criadas pelo menos as seguintes classes:

- Personagem
- Fantasma
- Alienígena
- Projétil
- Quartel
- Mapa
- Bomba

- Jogo

Outras classes podem ser criadas se o grupo julgar que podem contribuir no desenvolvimento como Vetor2D, Cronometro, Controle, etc.

A classe Jogo deverá possuir listas para armazenar bombas, inimigos e projéteis atualmente existentes no jogo. Quando um objeto deixar de existir (e.g., bomba explodir), ele deve ser removido da lista.

### **Restrições**

- O código não deve conter valores mágicos *hard-coded* e sem uma semântica clara. Devem ser criadas constantes em um arquivo de configuração do jogo.
- Cada classe deve ser armazenada em um arquivo .py separado.
- Os arquivos .py devem importar apenas classes e métodos que forem utilizados no arquivo.
- O nome de atributos e métodos privados devem iniciar com “\_” e estes não devem ser utilizados diretamente fora da classe em que eles foram definidos.
- Deve ser utilizada uma função main definida em um arquivo chamado main.py.
- Todos os atributos e métodos devem possuir *type hints*.