

Trabalho A2: Desenvolvimento de Jogo com Pygame

1. Objetivo

O propósito deste trabalho é desenvolver um jogo em Python utilizando a biblioteca Pygame e aplicar conceitos de orientação a objetos aprendidos em sala de aula. O projeto visa também incentivar boas práticas de desenvolvimento e qualidade de software, como modularização, documentação, testes unitários e controle de versão.

2. Divisão de Grupos e Tarefas

2.1 Turma de Matemática Aplicada: o trabalho deve ser realizado em grupos de **3 alunos**. A divisão de tarefas dentro do grupo deve ser clara e descrita em detalhes no relatório final.

2.2 Turma de Ciência de Dados: o trabalho deve ser realizado em grupos de **4 ou 5 alunos**. A divisão de tarefas dentro do grupo deve ser clara e descrita em detalhes no relatório final.

3. Entrega e Utilização do GitHub

Todo o desenvolvimento do projeto deve ser feito em um repositório no GitHub, que será utilizado como meio de entrega. O grupo deverá enviar um e-mail para o professor com o link do repositório até **04/12 às 23:59h**. Não haverá prorrogação.

É obrigatório o uso adequado do GitHub, com commits relevantes e consistentes ao longo do período de desenvolvimento. O histórico do repositório será avaliado em conjunto com a descrição detalhada da divisão de tarefas do grupo. Qualquer commit após a data limite acarretará um desconto mínimo de 2 pontos.

4. Critérios de Avaliação

- ☐ Funcionalidade do Jogo: O jogo deve funcionar corretamente, sem erros críticos, e implementar as funcionalidades básicas definidas no enunciado (e.g., movimentação do jogador, colisão, pontuação, etc.).
- ☐ Aplicação de Orientação a Objetos: O código deve ser estruturado utilizando classes e objetos, demonstrando o uso de conceitos como encapsulamento, herança e polimorfismo.
- ☐ Complexidade e Criatividade: O jogo deve apresentar um nível de complexidade adequado ao escopo do trabalho, com elementos de jogabilidade e design criativos.
- ☐ Qualidade do Código: O código deve ser claro, conciso, bem organizado e seguir as convenções de estilo do Python (PEP 8).
- ☐ Documentação: O código deve ser documentado de forma clara e completa, utilizando comentários e docstrings para explicar a lógica e o funcionamento do jogo.

- ☐ Testes Unitários: O projeto deve incluir testes unitários para as principais funcionalidades do jogo, garantindo a qualidade e a robustez do código.
- ☐ Uso adequado de controle de versão no GitHub: O histórico do repositório deve refletir o desenvolvimento do projeto, com commits frequentes e mensagens descritivas.
- ☐ Relatório: O grupo deve entregar um relatório descrevendo o processo de desenvolvimento do jogo, incluindo a divisão de tarefas, as decisões de design, as dificuldades encontradas e as soluções implementadas.

5. Instruções do Trabalho

O projeto consiste no desenvolvimento de um jogo 2D utilizando Pygame, com tema e mecânicas a serem definidos pelo grupo. O jogo deve incluir os seguintes elementos:

- **Jogador:** Um personagem controlável pelo jogador, com movimentação e animações.
- **Inimigos:** Elementos que desafiam o jogador, com diferentes comportamentos e níveis de dificuldade.
- **Cenário:** Um ambiente visual para o jogo, com elementos estáticos e dinâmicos.
- **Objetivo:** Uma meta clara para o jogador alcançar (e.g., coletar itens, derrotar inimigos, alcançar um determinado ponto).
- **Pontuação:** Um sistema de pontuação que acompanha o progresso do jogador.

Sugestões de Jogos:

- Jogo de plataforma (e.g., Super Mario Bros., Sonic the Hedgehog)
- Jogo de tiro (e.g., Space Invaders, Galaga)
- Jogo de labirinto (e.g., Pac-Man)
- Jogo de quebra-cabeça (e.g., Tetris)

Sugestão de Organização do Código:

É importante que o repositório GitHub seja bem organizado e de fácil navegação. Uma sugestão de estrutura de arquivos é a seguinte:

```
repo/  
docs/    # Diretório para documentação e relatórios  
src/     # Código-fonte do projeto, dividido em módulos  
assets/  # Diretório para imagens, sons e outros recursos do jogo  
tests/   # Scripts de testes unitários  
readme.md # Arquivo README explicando o projeto e como executá-lo  
requirements.txt # Arquivo com dependências do projeto (bibliotecas)
```

O arquivo `readme.md` deve conter uma descrição clara do projeto, instruções para instalação das dependências e para executar o código. O arquivo `requirements.txt` deve listar todas as bibliotecas necessárias para que o projeto funcione corretamente.

Milestone – Apresentação da Proposta:

No dia **01/11**, haverá uma apresentação em sala de aula para verificar a adequação das propostas de jogos. Cada grupo deverá preparar de 2 a 3 slides, mencionando o nome dos integrantes, o tema escolhido, as mecânicas principais do jogo e um esboço do design do jogo (imagens, personagens, etc.). Este checkpoint é essencial para evitar problemas na entrega final.