CES22 - Programação Orientada a Objetos - Comp 21

Relatório Individual 03

Adriano Soares Rodrigues {sadrianorod@gmail.com}

#Introdução

Após o término do projeto PyKombat é que se pode fazer uma análise geral a respeito de todo o processo de estudo, criação, organização e gerenciamento, tanto do código quanto da equipe de desenvolvimento, superando diversos problemas de implementação de código encontrados ou até mesmo de conflitos entre documentos no GitHub ao fazermos um merge dos códigos, problemas esses que só a prática pode apresentar e que fazem parte do cotidiano de um programador. Sendo assim a prática cobrada pela matéria de CES-22 foi de grande importância para o aprendizado da teoria de Programação Orientada a Objetos(objetivo principal do curso), implementada na prática com esse projeto.

Soma-se ao descrito o acima o aprendizado da linguagem Python, linguagem muito útil e de uso obrigatório para esse projeto, que se destacou por sua facilidade e simplicidade para minha equipe, não necessitando de nenhum estudo prévio devido ao seu alto nível.Diante dessa leve explicação, inicia-se uma discussão detalhada a respeito do processo final de criação do projeto Pykombat.

Figura 1. Tela de Menu do jogo PyKombat

#Objetivo

O meu objetivo principal foi implementar a parte da movimentação dos personagens com a classe Fighter, desenvolvendo toda a mecânica de um jogo de luta respeitando a física da movimentação. Inicialmente foi estipulado pela equipe desenvolver todo o código baseado numa movimentação utilizando diversos vetores de imagens do tipo png,onde cada vetor representa um tipo de movimento,por exemplo: jump, AHit, special, dentre outros. Além da movimentação que apresentou lógica e dinâmica de movimentação muito boas e fluídas para manter uma movimentação mais próxima da realidade,um segundo ponto positivo foi na criação das sprites através do Photoshop que deram ao projeto uma animação condizente com o clássico que foi Mortal Kombat na sua época,o jogo pode ser observado nas Figuras 1 e 2.

#Desafios e Mudanças do Planejamento

Entretanto, o principal desafio foi criar as colisões usando a abordagem descrita acima, quase no final do projeto a parte da equipe que estava focada em colisões resolveu o problema com muito mais velocidade usando o método das sprites, que são funções disponibilizadas pela própria biblioteca do python, o pygame. Assim tivemos que mudar a abordagem do código, mas devido a lógica de organização e pela facilidade das classes criadas, quase todo o código foi aproveitado ao utilizarmos as sprites criadas.

Além disso ajudei no suporte quando bugs surgiam, como foi o caso do pulo, que por problemas na lógica demorou para ser implementado, ou na implementação de parte dos códigos, alguns exemplos foram ajudar no desenvolvimento das barras de vida dos personagens além de implementar boa parte do arquivo projectile.py responsáveis por criar e deslocar os poderes especiais de ambos os personagens, particulares de cada um.Assim, o objeto criado pelo projectile.py se deslocam e ao acertar o personagem retiram vida que implica na diminuição nas barras de vidas.Uma amostra do jogo pode ser observada na **Figura 2**.

Assim, pode-se inferir que um dos principais pontos negativos além da falta de tempo, foi ter que refazer o código de toda a movimentação dos personagens, agora usando as sprites que facilitava muito mais a colisão, entretanto esse problema foi superado devido ao código estar bem

limpo e organizado, implicando numa rápida adaptação para a nova abordagem, parte do código antigo pode ser visto no link fornecido no anexo.

Além disso, uma deficiência notada foi a falta de comunicação entre a equipe nessa fase final, uma vez que alguns trechos de códigos foram feitos por mais de um membro da equipe, caso a comunicação fosse mais efetiva tempo e esforços seriam poupados e redirecionados para os problemas restantes. Nessa lógica, a falta de tempo também propiciou o não cumprimento de toda a agenda estipulada, como a tela de pause e o fatality, que só não foram implementados mas que a equipe conseguiria caso o tempo estipulado fosse maior.

É importante salientar que mesmo diante de todos esses problemas a equipe se mostrou como um todo muito unida e os principais problemas foram resolvidos, como pode-se notar no sucesso do jogo e na sua qualidade.

Figura 2. Tela de início de combate no jogo Py Kombat.

#Conclusão

Assim a disciplina de CES-22 tem se mostrado muito útil e prática para adquirir conhecimentos em amplas áreas, seja na programação ou no gerenciamento de projetos,

consequentemente sendo muito importante para o curso de computação em geral. Diante de todo os pontos positivos e negativos debatidos chega-se a conclusão que necessita-se de uma maior comunicação para evitar gerar código desnecessários ou repetidos além de uma agendada mais condizente com o que se espera que será produzido de acordo com a demanda de tempo exigida e com o tempo disponível em cada semana. Acredito que essa primeira parte do curso se mostrou muito eficiente para o aprendizado na prática, sendo muito condizente para o ensino mais moderno,onde o aluno possui a liberdade de criar seu próprio projeto, será de grande proveito para as turmas futuras.

#Anexo

Link para o código inicial da classe Fighter antes da adaptação:
https://github.com/vidalmatheus/pyKombat/commit/38eb0aab82be2724b6ffe73a76e3f4bf07d4009f

Link para o código final do PyKombat : https://github.com/vidalmatheus/pyKombat