

Universidade Federal de Pernambuco  
Centro de Informática - CIn  
Sistemas de Informação  
Programação 1 (SI) - Projeto da Disciplina

Tema 1: Sistema Interativo  
Jogo Quadradin

Equipe:

Angel Willyan (awrl)  
Antônio Lins (algm)  
Gabriel Monteiro (gmcs)  
João Lucas (jlvs)  
Lucas de Castro (lclmg)  
Lucas Gabriel (lgsn)

Repositório:

<https://github.com/gmonteiroc21/projeto.py>

## Organização do código

Nosso projeto tem como propósito um Sistema Interativo modelado com Orientação a Objetos com o auxílio da biblioteca principal “pygame”.

Nosso programa é um jogo estilo Pacman, onde o jogador coleta diferentes objetos determinados por cores distintas que possuem diferentes habilidades, sendo elas: ciano: se torna invencível até coletar outro objeto de outra cor; vermelho: ganha velocidade; verde: perde velocidade. O objetivo principal do jogo é coletar 10 objetos verdes enquanto o inimigo persegue o jogador.

Inicialmente, fizemos ‘imports’ a partir das bibliotecas “pygame”, “sys” e “random” para auxílio no desenvolvimento. Em seguida, demos início aos comandos de elaboração da tela do jogo, em resolução de 600px de largura x 600px de altura, assim como nomeamos a janela.

Determinamos as variáveis iniciais, como por exemplo, a velocidade do jogador e do inimigo (vilão), as fontes e suas configurações, além de outras.

Em seguida, foi criada a classe “Player”, onde estão todas as funções necessárias para o jogador, como andar nos eixos horizontal e vertical, a colisão, que seria a coleta dos objetos e a contagem dos mesmos e seu “desenho”/formato.

Também foi criada a classe “Coletaveis”, onde define o parâmetro das cores, seus raios e os desenha em círculos, além da classe “Vilao”, com todas as suas características e funções dentro do jogo..

Com um laço de repetição, foi criado o menu inicial com mensagem de “Bem-Vindo(a)”, informação de como iniciar o jogo e as habilidades que cada objeto possui.

Em seguida, temos a criação do jogador e inimigo em suas respectivas posições iniciais em relação a tela do jogo, assim como os objetos coletáveis, com suas aparições aleatórias de acordo com que são coletados.

Com outro laço de repetição, foram criados outros comandos enquanto o jogador está vivo, como o informativo de coletas de objetos, mensagem com quantos objetos verdes precisam ser coletados para ganhar. Assim como a função de apertar o botão “X” para sair do jogo ou para iniciar a andar, você deve apertar qualquer tecla de seta (para cima, para baixo, para esquerda ou para direita), que em seguida, o jogador mantém movimento constante, apenas com orientação a mudança de direção com as mesmas setas.

Foi criado também o comando de coleta de verdes para ganhar, em que se você coletar 10 objetos verdes, o jogador ganha (comando 'campeao') e surge a tela de ganhador com as opções de recomeçar ou sair. Caso o inimigo/vilão alcance o jogador, o mesmo perde (comendo 'not campeão') e surge a tela de 'Game Over' com os mesmos comandos de reiniciar ou sair.



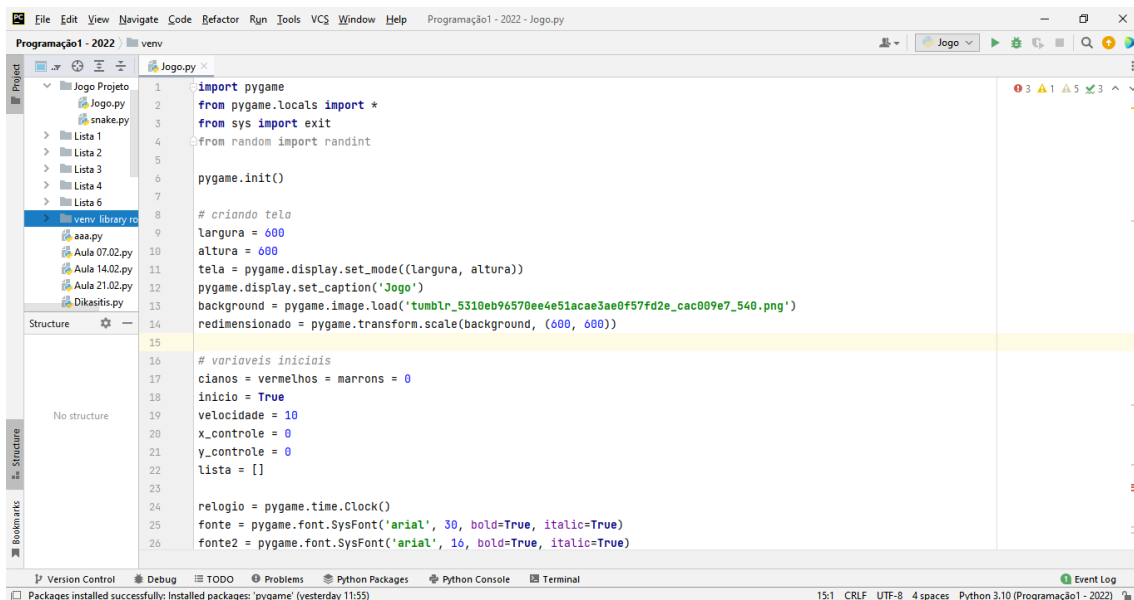
## Ferramentas, bibliotecas e framework

Para desenvolvimento do código do trabalho, usamos a ferramenta Pycharm Community, onde desenvolvemos e executamos o programa para testes antes de adicioná-lo ao repositório do GitHub, assim como após adição ao GH, ainda o utilizamos como meio de testes de execução.

Para um código mais completo e, assim, com maior eficiência para nossos objetivos, usamos algumas bibliotecas Python. A principal delas, para formulação do jogo, usamos a biblioteca *pygame*, que é um conjunto de módulos Python projetados para desenvolver jogos. Ela adiciona funcionalidade em cima da biblioteca SDL, assim permitindo que se crie jogos e programas multimídia completos na linguagem python.

Da biblioteca *sys*, que fornece funções e variáveis usadas para manipular diferentes partes do ambiente de tempo de execução do Python, importamos o comando *exit*, que nos permite sair do programa em execução.

Com a biblioteca *random*, que fornece ferramentas para gerar seleções pseudoaleatórias, importamos o comando *randint*, que retorna um valor aleatório dentro do range determinado.



## Divisão das tarefas

- Código-main: Gabriel Monteiro
- Colaboração no código (branch): Todos da equipe
- Check-point: Lucas de Castro e Gabriel Monteiro
- Relatório: Lucas de Castro e Lucas Gabriel
- Slide: Angel Willyan e João Lucas

## Conceitos

Dos conceitos vistos na disciplina, usamos as bibliotecas, que são um conjunto de módulos e funções úteis que reduzem o uso de código no programa, usamos comandos condicionais para determinarmos quais escolhas ou funções serão tomados, como por exemplo, na colisão de objetos do jogo, em que se o coletável for vermelho, ele ganha velocidade e contabiliza o objeto para ser mostrado em tela.

Usamos também laços de repetições, que permite que o código seja executado repetidamente com base em uma determinada condição booleana, como no nosso menu, que enquanto o jogador não pressionar a tecla única de inicialização do jogo “espaço”, o jogo não começa.

Além disso, também utilizamos das listas e tuplas como forma de reunir comandos ou informações em um único local.

Criamos as funções *def*, que são blocos de código que realizam determinadas tarefas que normalmente precisam ser executadas diversas vezes dentro de uma aplicação, como a função de andar em função dos eixos horizontal e vertical.

*Qual foi o maior erro cometido durante o projeto? Como vocês lidaram com ele?*

A falta de comunicação atrapalhou bastante a evolução do projeto, principalmente no início. Devido a isso, tinham mudanças que não eram bem definidas e tinham impacto no código do jogo, deixando ele algumas vezes desorganizado, fazendo com que a gente demorasse mais no andamento e nas modificações, visto que cada mudança poderia gerar um novo erro. Mas ao final do projeto, tudo foi alinhado.

*Qual foi o maior desafio enfrentado durante o projeto? Como vocês lidaram com ele?*

Conseguir conciliar o tempo para elaboração do projeto com projetos de outras disciplinas, tornando a reunião do grupo reduzida devido às demandas individuais. Decidimos realizar as nossas reuniões de maneira mais informal, no grupo da equipe no whatsapp, de uma forma que cada um avisava o que fez/alterou e sua sugestão para melhoria do código e do projeto, dessa forma conseguimos uma melhor interação entre os integrantes do grupo, mesmo que em breves momentos, deixando para as reuniões do meet as mudanças que teriam um maior impacto no escopo do projeto.

*Quais as lições aprendidas durante o projeto?*

Como a maioria do grupo era inexperiente com orientação a objetos e o conhecimento da plataforma GitHub, assim como a aplicação de conceitos do “Pygame”, esse projeto foi uma ótima oportunidade para aprendermos sobre tais conceitos. Todos os membros buscaram orientações em tutoriais, fóruns e ou auxílio de terceiros com conhecimento do assunto no intuito de buscar esse aprendizado e compartilhá-lo com os demais membros da equipe.

Ao final do projeto, a maioria dos membros já estavam familiarizados com alguns comandos do GitHub e uma execução automática do pygame. Isso permitiu com que, ao observar um erro ou possível melhoramento no código, fosse capaz de fazer tais alterações de forma independente.