Relatório:

A defesa do Ho-Ho

Realizado por: Daniel Silva 90120

23 de julho de 2022

Conteúdo

[Descrição do projeto: 3](#_Toc109426492)

[Proposta: 3](#_Toc109426493)

[Resposta à proposta: 3](#_Toc109426494)

[Realização: 3](#_Toc109426495)

[Técnicas utilizadas: 4](#_Toc109426496)

[Diagrama de Classes: 4](#_Toc109426497)

[Dificuldades: 5](#_Toc109426498)

# Descrição do projeto:

## Proposta:

Para a unidade curricular POOI, foi-nos proposto um trabalho final com o objetivo de demonstrar o conhecimento de cada aluno que foi adquirido durante a duração da unidade curricular.

### Resposta à proposta:

Para este efeito, decidi criar um jogo que contem uma base (Torre) do lado esquerdo do mapa, que vai ser atacada por *Pokemons* que virão do lado direito e o jogador irá defendê-la utilizando o seu *Pokemon*.

Vou utilizar classes para criar os inimigos, o jogador e as habilidades, o que vai permitir a utilização da matéria aprendida em aula.

## Realização:

Este jogo consiste na defesa de uma torre que irá ser atacada infinitamente.

Para a defender, temos controlo do *Bulbasaur*, o *Pokemon* nº 1 podemos andar com ele utilizando as setas da direita e esquerda, saltar com a barra de espaços e utilizar duas habilidades, o *Bullet Seed* no F e o *Hyper Beam* no D.

Dentro do jogo, um *Squirtle* é criado sempre que não exista mais nenhum, ele tem como o objetivo destruir a base do jogador, para esse efeito ele começa a sua vida do lado direito do mapa e vai progredindo para a esquerda até chegar à torre, se o fizer tira 1 de vida, no entanto o jogador pode matá-lo com as suas habilidades, mas se o *Squirtle* colidir com o jogador, o jogador também perde vida e pode morrer tendo este 2 vidas.

## Técnicas utilizadas:

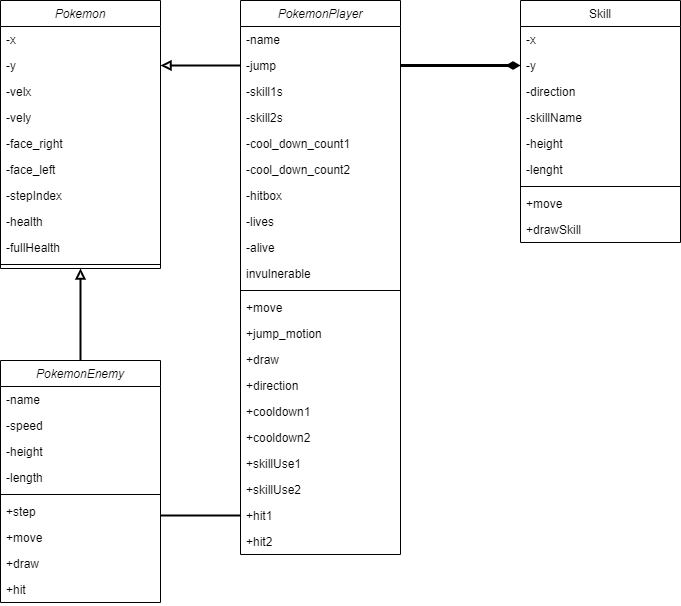
No desenvolvimento deste jogo foram utilizadas várias ferramentas e librarias do Python e também foram utilizados o *Affinity Photo* e Draw.io.

A libraria mais utilizada foi a *Pygame*, o qual disponibilizou as ferramentas necessárias para o desenvolvimento de um ambiente físico e gráfico. Com ela foi possível introduzir todo o ambiente do jogo, desde imagens como controlos das entidades presentes.

Outra das librarias usadas foi a *OS*. que proporcionou comandos de procura de imagens através de caminhos de sistema operativo.

Para alem das librarias mencionadas acima, foram também utilizadas ferramentas base do Python que permitiram que o jogo funcionasse. Das quais, as mais utilizadas foram as classes e as funções, a classes foram utilizadas para criar as habilidades e os *Pokemons* do jogador e inimigos, já as funções foram utilizadas para a execução de contas e comandos para as mecânicas do jogo. Foram também utilizadas várias variáveis e vetores para definir parâmetros para o jogo.

# Diagrama de Classes:



## Dificuldades:

Para mim a parte mais difícil deste projeto, foi dividir o código, para estar dividido em scripts diferentes, devido a ter seguido um tutorial para aprender o *Pygame* onde o instrutor programou as classes no script principal, e ao dividi-lo foi necessário alterar funções adicionando-lhes parâmetros e criar conexões entre classes.

Outra dificuldade foi a criação de outra habilidade, devido à primeira ter um hitbox pequeno, o código para tirar dano não funcionava muito bem para a segunda habilidade