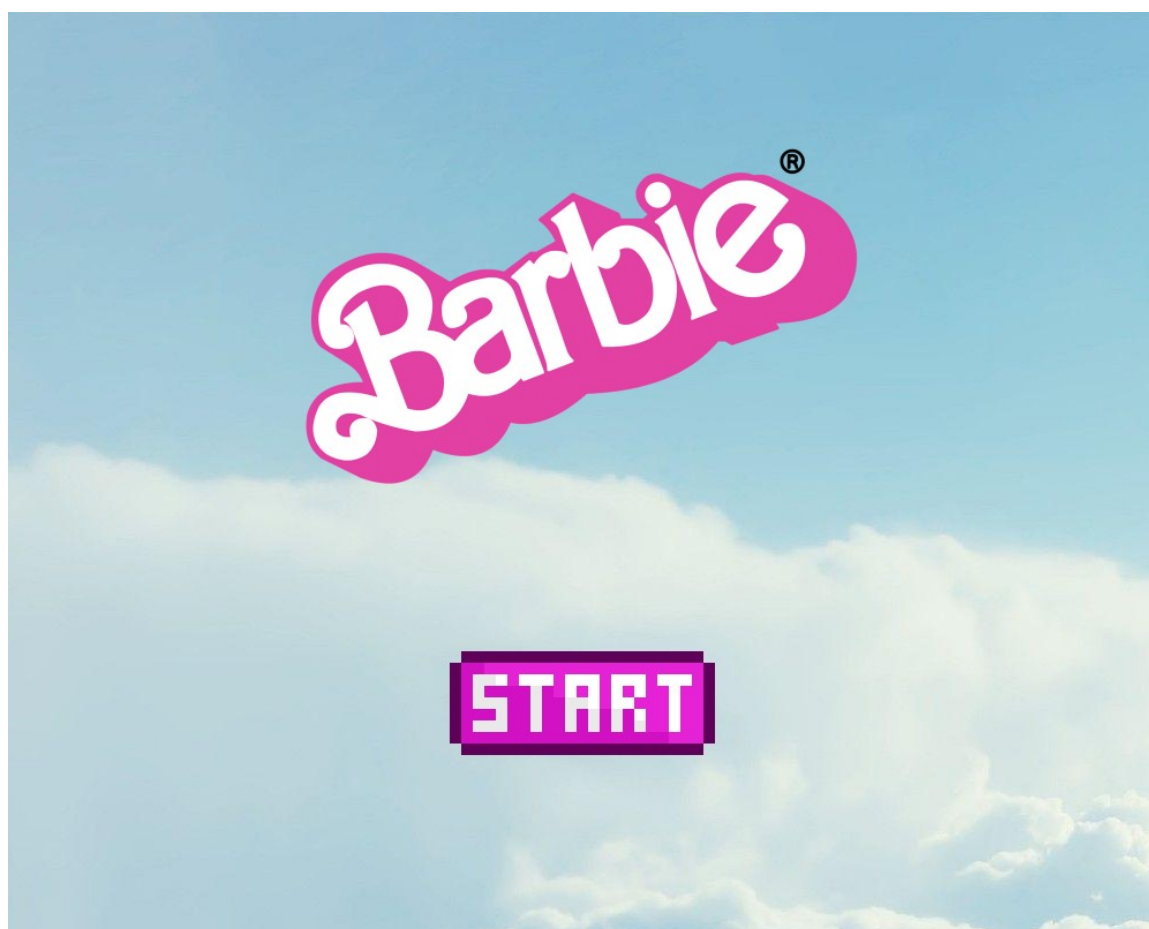




UNIVERSIDADE da MADEIRA

Programação Orientada por Objetos 2023/2024



Projeto realizado por:

Leandro Gonçalves - 2123522

Francisco Henriques – 2167322

Lucas Wenddell - 2117622

Introdução

No âmbito da disciplina de Programação Orientada por Objetos (POO), foi nos atribuído um desafio (Projeto), em forma de jogo, que tinha como principal objetivo testar os conhecimentos obtidos, até à data em Greenfoot, na disciplina de Programação Orientada por Objetos. O jogo tinha como tema principal o relativamente recente lançado filme “Barbie”, grande sucesso de bilheteria pelo mundo inteiro no dia da sua estreia. As personagens principais eram a Barbie e o Ken, que tinham como objetivo voltar para a BarbieLand.

Os conceitos que tínhamos de utilizar no jogo eram as seguintes: Inicialização de objetos usando os construtores; Herança de métodos com um mínimo de 2 níveis além de Actor; "Overriding" de métodos; "Overloading" de métodos; Encapsulamento. Para além disto, foi nos dado certos aspetos para o jogo, que tínhamos de seguir como: 2 ou mais jogadores que colaboram; Score com a pontuação atualizada em tempo real; Indicador de vida/energia/tempo ou semelhante para cada jogador; Modificação do aspeto do mundo e dos jogadores (por exemplo, imagens, posição de objetos, ou cores) durante o jogo; Indicação da pontuação obtida ao finalizar o jogo.

Com tudo isto que nos foi apresentado, começamos logo a pensar no que iria ser a nossa “história” para o nosso jogo. Tivemos diversas ideias, algumas delas geniais, mas relativamente difíceis de implementar em Greenfoot. Depois de dias de pensamento, e de reflexão, chegamos a um acordo de como o jogo iria funcionar e a como iria desenrolar a “história”.

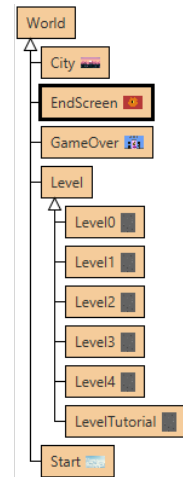
O jogo começa, num prédio, mas não é um prédio qualquer, mas sim o Prédio onde a própria Barbie e o Ken foram desenvolvidos. Num primeiro nível, temos a Barbie e o Ken a subirem, desesperadamente, por um conjunto de plataformas de pedra, de forma a conseguirem chegar ao topo, e passarem para o próximo nível. Contudo, ao passarem de nível, deparam-se com os empresários do Prédio da Barbie. De forma a chegarem ao topo desse andar, novamente, eles têm de desviá-los, não podendo tocar neles, pois isso significaria o “reset” do nível. Chegados ao terceiro nível, deparam-se com os seguranças que andam atrás deles, furiosamente, de forma a tentar expulsá-los do prédio, pois não é suposto eles estarem ali. Passados esse nível, chega um ainda mais desafiador, que une os seguranças com os empresários, que formam uma união, para tentarem apanhar a Barbie e o Ken. Sem sucesso, a Barbie e o Ken passam de nível e chegam ao último nível. Neste nível, pode-se observar uma enorme torre com uma bomba pendurada. Ao chegarem lá em cima, eles encontram-se com simplesmente único e destruidor, Oppenheimer, pai da bomba nuclear. Com muitos esforços do Ken e da Barbie, eles derrotam o Oppenheimer. Abre-se de seguida um portal, que os leva por fim para a BarbieLand, onde eles descansam por fim.

Com isto, terminamos a introdução e iremos agora passar para o que realmente interessa, que são o código por detrás desta história magnífica.

Classe World

A classe “World” é onde todos os Atores são associados com as diversas subclasses. Facilitando assim a busca de certos objetos da classe “Actor” A classe “World” tem 4 subclasses, sendo elas:

- City;
- GameOver;
- Level;
- Start;
- EndScreen



Subclasse City

Esta subclasse inicializa um novo mundo, caso os jogadores gastem todas as vidas no último nível do jogo, aparece este novo mundo, que consiste numa imagem de uma bomba nuclear e ao mesmo tempo um som de uma sirene.

Subclasse Endscreen

Esta subclasse nada mais faz que inicializar um novo mundo, em que caso o jogador passe o último nível do jogo, aparece este novo mundo, com uma imagem de vencedor e uma música.

Subclasse GameOver

Esta a subclasse só tem dois parâmetros no seu construtor, um “super” que criará um mundo de 1200 de comprimentos por 800 de altura. Para além disso temos a método `Greenfoot.stop()`, que irá parar o jogo, quando o jogador tiver gasto todas as vidas, aparecendo no ecrã uma imagem de “Game Over”, fazendo com que o jogador comece de novo do primeiro nível.

Subclasse Level

Esta subclasse é casa de todas as funções que iremos utilizar em cada subclasse de “Level”, sendo estas **Level0, Level1, Level2, Level3 , Level4 e LevelTutorial**. Lar de funções que irão fazer com que apareça a contagem decrescente em cada nível, os pontos e a continua contagem de pontos ao passar de um nível. Para estes feitos, utilizamos variáveis estáticas que reiniciam, toda a vez que o jogo é reiniciado. Utilizamos variáveis que guardam valores, de modo a obter a pontuação e criar contadores, contadores esses necessários para o temporizador. Usamos também ciclos `if's` e ciclos `for's`, como também fazemos “**overloading**” na subclasse Level, ou seja,

utilizamos dois construtores, sendo que um deles é utilizado especificamente para o Level de Tutorial, e o outro construtor é utilizado para os outros 5 níveis.

Nas subclasses de “Level”, podemos encontrar um conjunto de funções que irão fazer a decoração desse nível. Utilizamos imensas vezes o método addObject, método do Greenfoot que adiciona os objetos que pretendemos ao cenário do Greenfoot. Para além disto criamos um pequeno ciclo for que faz com que o mundo suba, periodicamente, fazendo também com que o jogo fique mais desafiante.

Subclasse Start

Esta subclasse nada mais faz do que adicionar um fundo para o nosso menu do jogo. Adicionar um botão “Start” e o “Icon” da Barbie. Já nesta subclasse são inicializadas as variáveis estáticas dos Níveis, de modo a facilitar erros de contagem.

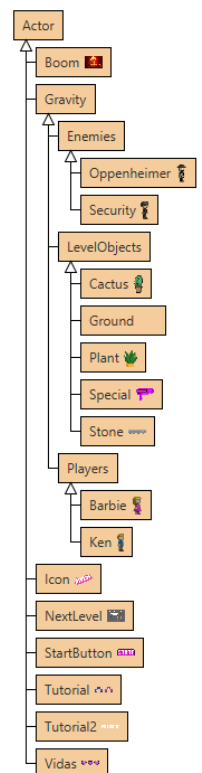
Classe Actor

A classe Actor é lar de uma infinidade de métodos que nos ajudariam a criar este projeto/jogo. Foi nos ensinado na disciplina de POO a utilizar diversos métodos, e a ir busca-los ao GreenFoot Documentation, onde estão os métodos todos.

Um dos métodos que utilizámos mais foi o **move ()**, sendo que na subclasse “Players”, damos **Overriding** desse método, ou seja, atribuímos outra função a esse método já existente.

A classe Actor, por sua vez, possui diversas subclasses como as subclasses:

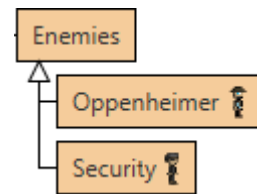
- Boom;
- Gravity;
- Icon;
- NextLevel;
- StartButton;
- Vidas;
- Tutorial;
- Tutorial2.



Subclasse Boom

Esta subclasse contém uma imagem de uma explosão. Este é um possível final de jogo caso os jogadores gastem todas as vidas no último nível do jogo.

Para além da imagem a cobrir o ecrã, ouvir-se-á também um som de uma sirene/alarme que irá estar-se sempre a ouvir, caso os jogadores não reiniciem o jogo.



Subclasse Gravity

Esta subclasse de Actor tem como função guardar todos os métodos utilizados para a existência de gravidade no jogo. Sendo que esta subclasse possui 3 subclasses que herdam todos esses métodos, sendo elas as subclasses:

- Enemies;
- LevelObjects;
- Players;

Enemies

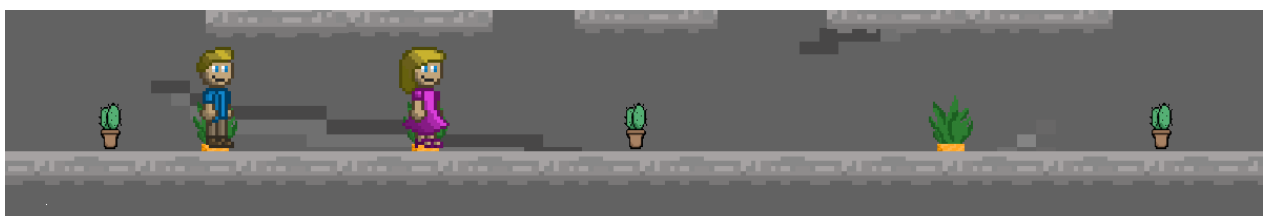
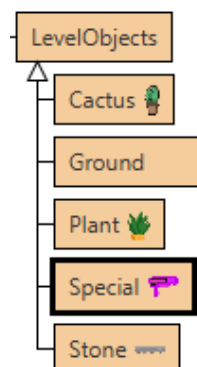
Esta subclasse, por sua vez, guarda alguns métodos que 2 subclasses de "Enemies" herdam. Sendo elas as subclasses:

- Security;
- Oppenheimer;

Estas duas subclasses representam personagens do jogo, que acrescentam um certo nível de dificuldade aos jogadores. O código implementado nestas duas subclasses é essencialmente, a mudança de imagem constante, enquanto eles estão em movimento, de forma a criar uma animação de andamento.

LevelObjects

Como o nome indica, esta subclasse detém todos os objetos que dão "spawn" no jogo. Contendo assim diversas subclasses que servem exclusivamente para a decoração do mundo, tirando a subclasse "Special" que é um item especial, que tem o código necessário para que os jogadores possam passar de



nível. Sendo todas essas subclasses designadas por: **Cactus; Ground; Plant; Special; Stone.**

Players

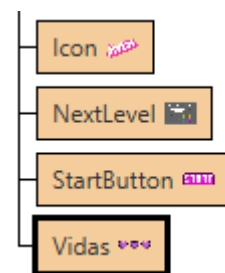
Esta contem todos os métodos necessários para fazer com que seja possível o jogador, controlar as principais, sendo elas o Ken e a Barbie. Elas são separadas subclasses, que herdam os métodos da classe superior ("Players"). Ambas as subclasses são controladas por key-binds, específicas para cada um dos 2 jogadores possíveis. O Ken é controlado pelas teclas "w", "a" e "d", enquanto que a Barbie é controlada pelas teclas "left", "up" e "right".

Subclasse Icon

Esta subclasse não tem código, simplesmente é um objeto utilizado na classe "Start", que representa um elemento de decoração do menu inicial do nosso jogo.

Subclasse NextLevel

Esta subclasse também não tem código, simplesmente é um objeto utilizado como Loading Screen, leva a que os jogadores esperem relativamente pouco para que o próximo nível seja inicializado. Nesta tela aparece também a pontuação dos jogadores.

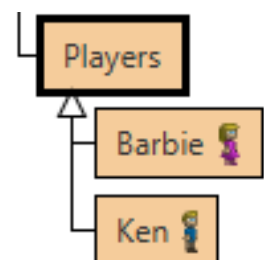


Subclasse StartButton

A subclasse "StartButton", é um botão que os jogadores ao clicarem nele, serão redirecionados para o primeiro nível do nosso jogo.

Subclasse Vidas

Esta subclasse é inicializada nos níveis, de modo a que os jogadores ao começarem o jogo, eles tenham 3 vidas. As vidas vão diminuindo para 2 vidas e sucessivamente para 1 vida. Quando está em 1 vida, se os jogadores tocarem num dos inimigos, o jogo terá de ser reiniciado, com 3 novas vidas e a pontuação obtida volta a estar 0.



Subclasses Tutorial e Tutorial2

Ambas estas duas subclasses não possuem código, servem apenas para que no nível Tutorial, apareça as indicações de como mover os personagens Ken e Barbie, de forma a ensinar os jogadores a jogarem o nosso jogo.

Conclusão

Em suma, achamos que o nosso projeto segue todas as indicações indicadas para uma boa realização do projeto, cumprindo todos os objetivos propostos pelos professores de Programação orientada por Objetos. Sendo que este projeto, foi bom para testar os nossos conhecimentos e habilidades na forma como construímos o código e o organizamos, de forma a ficar cada vez mais eficiente e com um bom aspeto visual.

