**OOP GAME REPORT**

***Members***

*Liêu Phong Sơn – ITITIU18200*

*Vũ Thành Nhân – ITITIU21267*

*Trần Quốc Nam- ITITIU21250*

1. **Introduction**

In this report, we introduce our remake version of the PlantsVersusZombies game. This is a well-know game, and it is challenging to create this game from scratch. We developed this game with the help of the JSwing framework and  [Slick2D](http://slick.ninjacave.com/) library of Java.

1. **Github repository**

This is our Github link: https://github.com/lieuphongson/Plant-vs-Zombie-OOP

1. **Game Rules**

**Game Rules:**

As we are developing this game as simple as possible, the game has three rounds. The initial sun of the round is 1000, after that we have 25 “sun” for random in the round. We need to collects “sun” with which defending plants can be bought. Zombies approach along several parallel lanes on the lawn, and the player must plan defenses in these lanes. If a zombie makes it to the house on any lane, the level is over.

1. **UML Diagram and Design Explanation:**

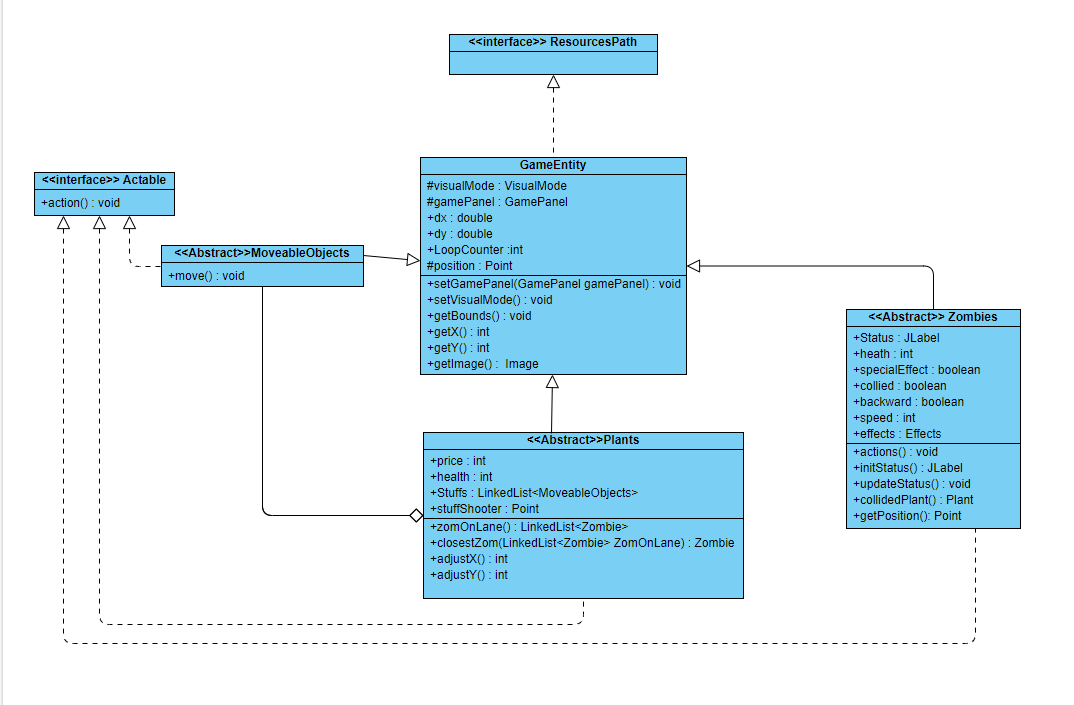
Firstly, we have an abstract class Character implement an interface class CharacterI.

Then, from this base class, we created 2 abstract subclasses to categorizes the entities further:

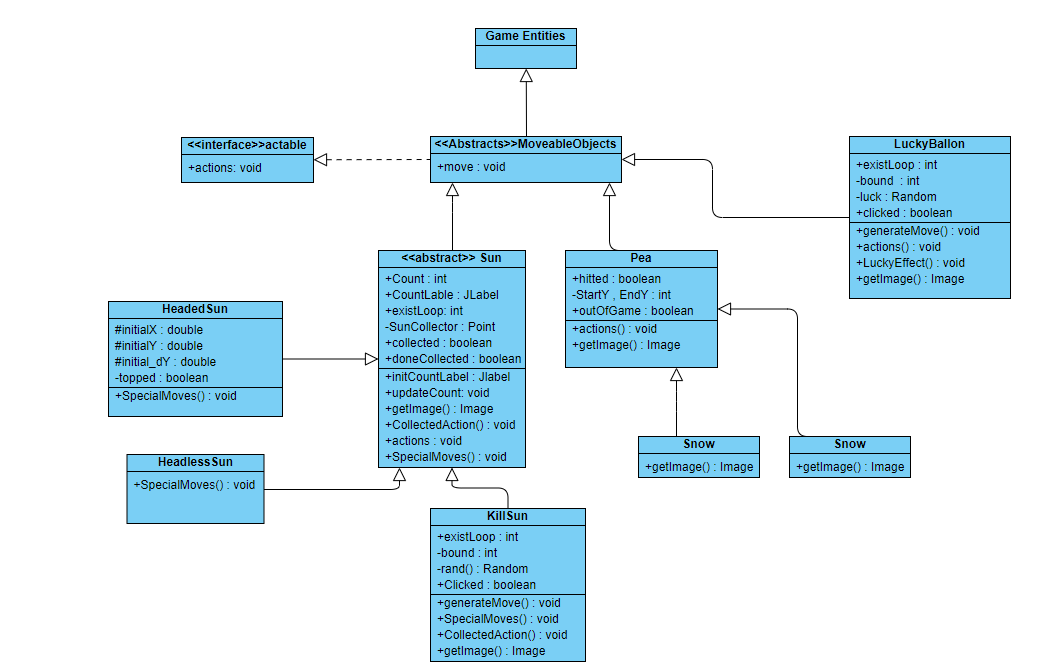
* Plant
* Zombies

Then, using polymorphism, we extend each abstract class above to create various concrete classes, which each class overwrite the common methods above:

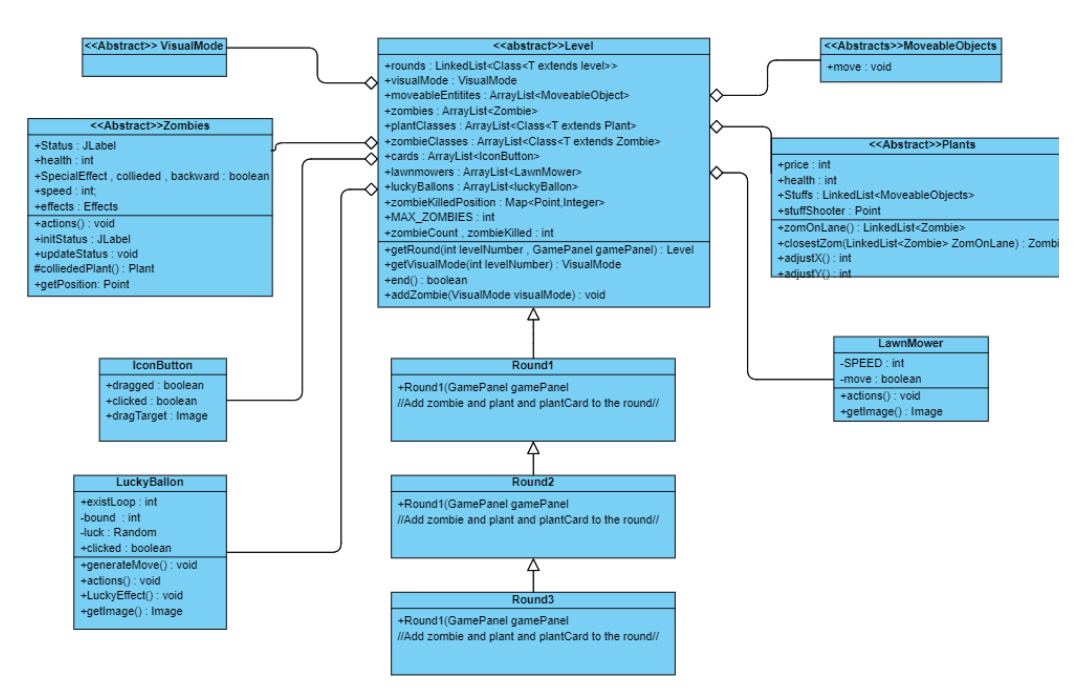
* move(): do the correct movement
* actable(): do correct action



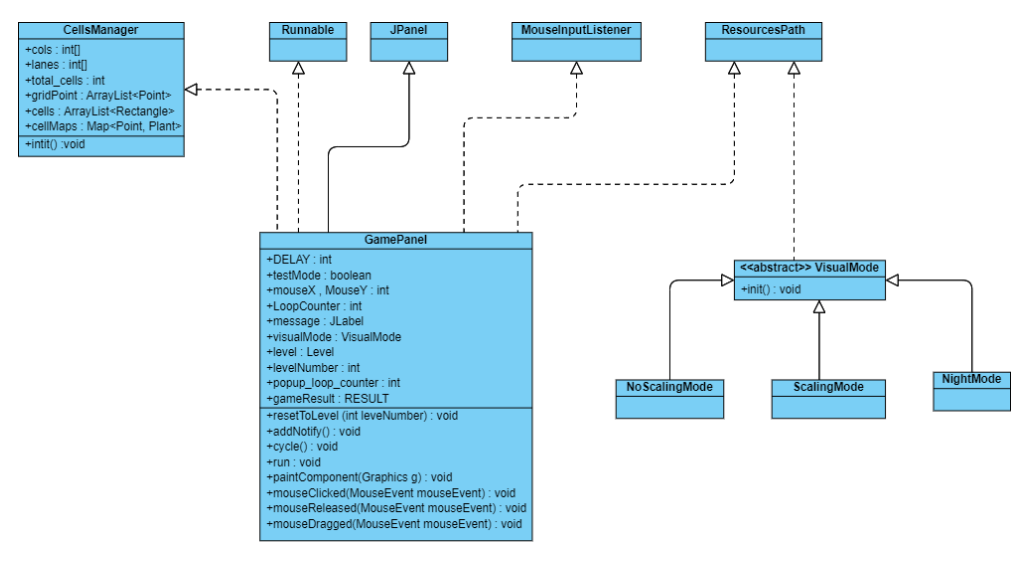
**Figure 1. Game Entities**



**Figure 2. Moveable (Sun & Bullet & Special object)**



**Figure 3. Game Mode**



**Figure 4. Game Panel**

1. **Evaluation**

**Overall evaluation**

The project makes good uses of basic Object Oriented programming ideas such as inheritance, interfaces, abstract classes to express the relationship of components of the game. By dividing the project into many levels of abstraction using Strategy design pattern, it is possible to keep the minimum change needed for any upcoming feature/optimization as well as flexibility in the game mechanism, dependencies, and resources. Nonetheless, the lack of a consistent set of principles is also displayed in the code and remains a barrier as the codebase grows larger.

1. **Plan for future improvements**

In the future, we will develop more function for the game due to the lackness of endgame like if player want to end the game without waiting the zombies get into the house, they can press the button endgame.