

BOĞAZİÇİ UNIVERSITY  
INDUSTRIAL ENGINEERING



*IE 201: Intermediate Programming*

---

*TERM PROJECT – “Plants vs Zombies Game”*

*Analysis Report*

*13.12.2020*

Group 4

Baran Kırkgöz      2018402048

Barış Arslan      2019402129

Eyyüp Taşçı      2019401108

## CONTENTS

---

I.	Description of the game .....	3
II.	Use Case Diagram .....	4
III.	Class Diagram .....	5
IV.	Explanation of the classes and relations .....	6

## DESCRIPTION OF THE GAME

---

Plant vs Zombies is a game that basically players try to kill zombies with seeds and keep them away from the left side of the screen. When zombies reach to left side of the screen, game ends.

There are 2 types of plants: sunflowers and peashooters. A sunflower allows a player to collect suns in a given time period, which can be used to plant more sunflowers and peashooters. A peashooter attacks zombies by sending their peas on them and when a pea reaches to a zombie, zombie's HP goes lower. If the number of attacks passes the HP of a zombie, that zombie dies.

Zombies can only attack to plants when they are directly in front of them. And when the attacks of zombies pass HP of plants, they also die. If there left no plants in zombie's tile, it can move to next tile and when it reaches the left side of the screen the game ends. Otherwise the game continues till forever.

## USE CASE DIAGRAM

---

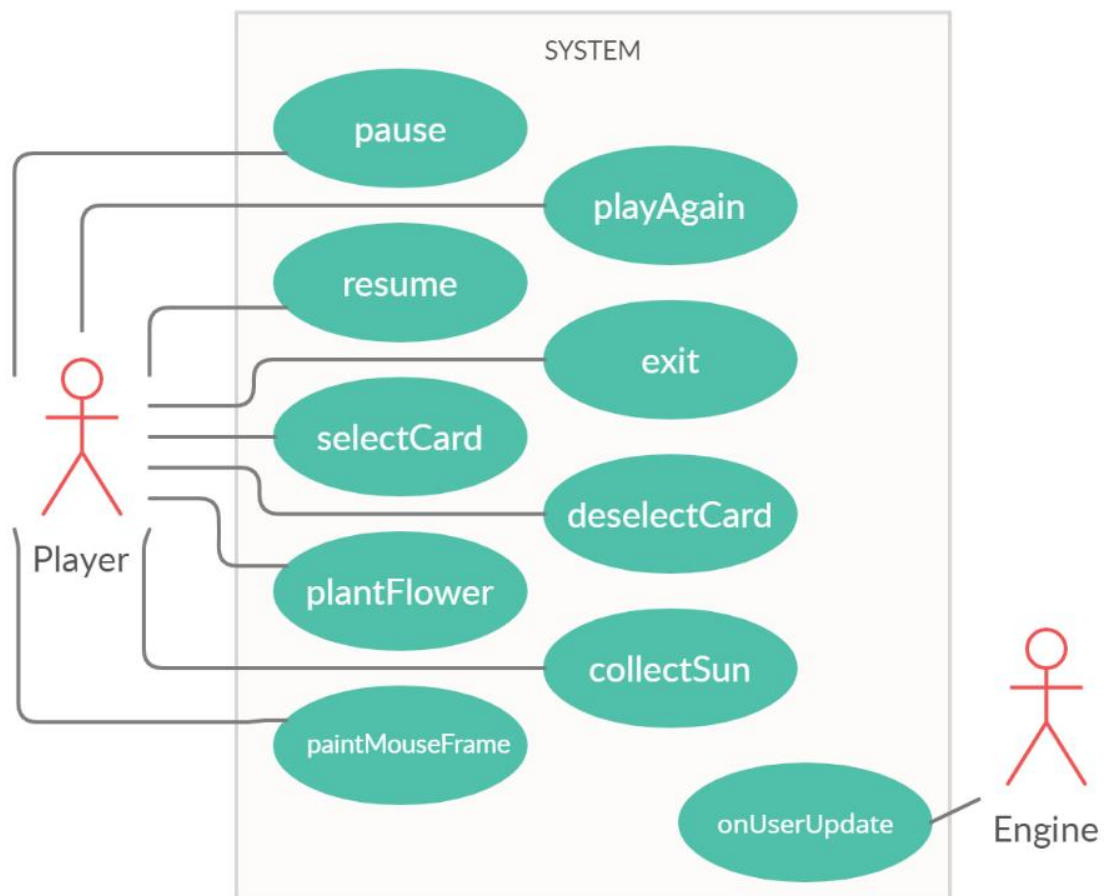


Figure 1.

Use case diagram simply describes the expected behaviors of the system from the perspective of the users. As can be seen from the figure 1, our use case diagram consists of ten use cases, nine for human player and one for engine itself.

Firstly 'pause' is the case when the player wants to temporarily stop the game, and 'resume' is the case to continue playing. When the game is over the player can either restart the game or quit the game. These behaviors correspond to 'playAgain' and 'exit' cases, respectively. While playing the game; the player can select a card, plant the selected card, deselect the selected card or collect the sun by clicking. 'selectCard', 'deselectCard', 'plantFlower' and 'collectSun' use cases represent these behaviors, respectively again. Lastly player can move the cursor upon the graphics of the game and this behavior represented by 'paintMouseFrame' use case. Lastly computer operates the game by running a loop, 'onUserUpdate' use case is the time when the computer runs this loop.

## CLASS DIAGRAM

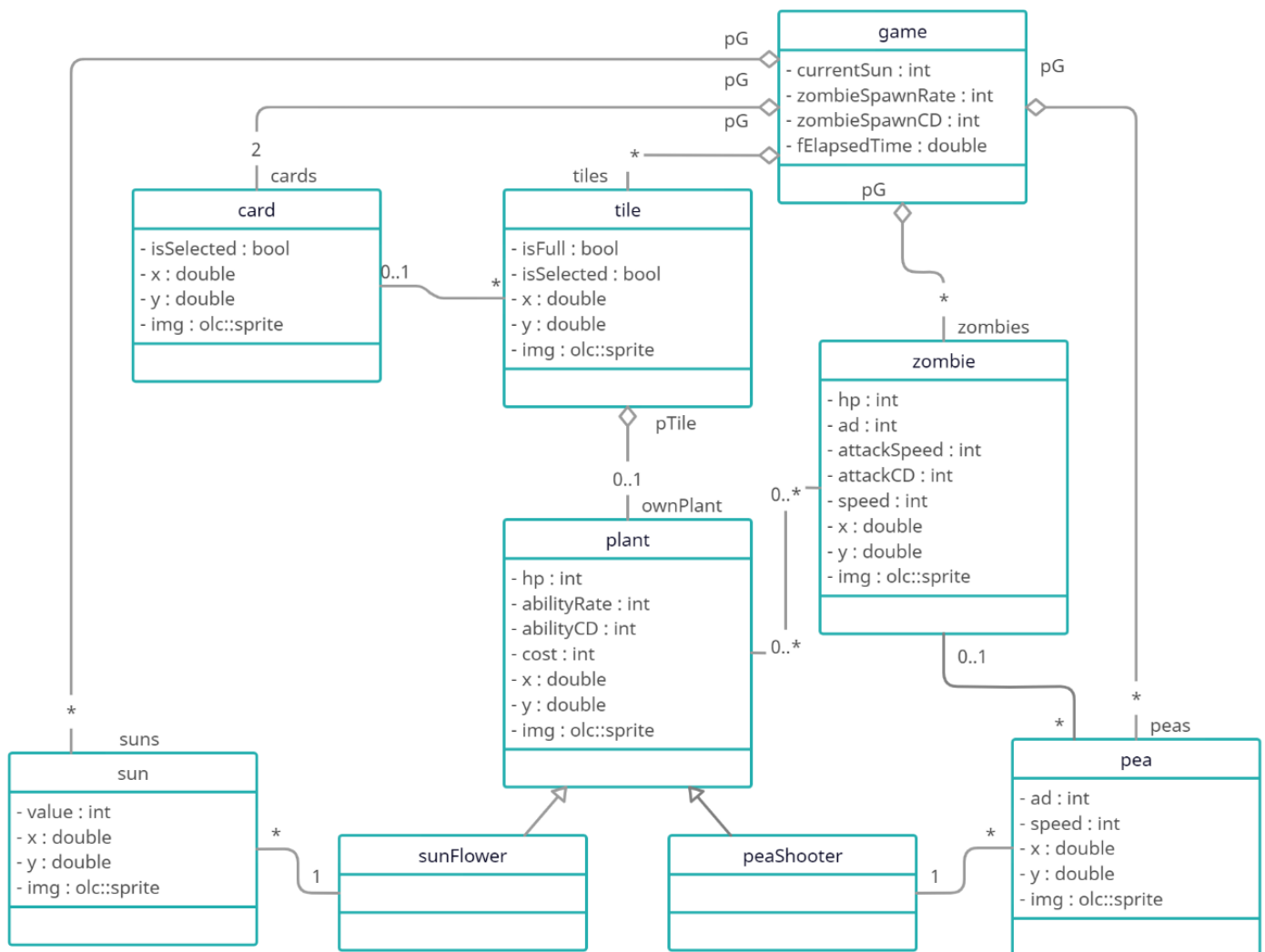


Figure 2.

## EXPLANATION OF THE CLASSES AND RELATIONS

---

Firstly, game object has two card objects within 'cards' variable, multiple tile objects within 'tiles' variable, zero or many zombie objects within 'zombies' variable, zero or many sun objects within 'suns' variable, and zero or many pea objects within 'peas' variable. Pointer to game object is stored as 'pG' in these variables. Game object also stores the current value of the sun, the time has been passed since the last use case of the 'onUserUpdate', time duration for zombie spawn rate together with the count down for zombie spawn.

Tile class represents the places that plants can be planted. Every tile object has the same image, but different coordinates that correspond to the place of the object on the game screen. Tile objects hold the data which are the existence of a plant in that object and the selection of the object. Every tile has at most one plant and tile object holds this plant within the 'ownPlant' variable. Also, every plant holds pointer to its tile with the name 'pTile'.

Card class has attributes for graphical appearance and selection. Card objects indicate which plant type will be planted. If any tile object is selected after a card object is selected, a new plant is created within the tile object.

Plant class has attributes for health point, cost, ability rate, and ability cooldown along with graphical attributes similar to other objects. As the name indicates health point holds the remaining health of the plant. Cost is the required value of sun for planting. Ability rate is the required time duration for the occurrence of the plant ability whereas ability cooldown is the remaining time for the occurrence of this plant ability. Plant class inherits two different classes: 'sunFlower' and 'peaShooter'. These two classes correspond to different plants with different abilities. sunFlower objects produce sun objects, peaShooter objects shoot pea objects.

Pea objects are the objects that peaShooter objects throw. Every pea object is shot by one peaShooter. However, one peaShooter object throws many peas. Each pea objects have speed and move along in the screen until it reaches the right end of the screen. If it encounters any zombie object, it gives attack damage to the zombie object with the same amount of 'ad' variable.

Sun objects are the objects that sunFlower objects create. Every sun object is created by one sunFlower object whereas one sunFlower object can create many suns. Each sun has location, image, and value which is the amount of the sun that will add up to the 'currentSun' quantity in the game object.

Zombie class has attributes for health point, speed, attack damage, attack speed, and attack cooldown. Health power holds the remaining health of the zombie object which is quite similar to the plant. Zombie objects move along in the screen with a speed until it reaches the left end of the screen. If it encounters any plant object, it gives attack damage to that plant object with the same amount of 'ad' variable within the intervals determined by attack speed. Attack cooldown holds the remaining time for the next attack.