IE 201 TERM PROJECT REPORT 1

GROUP 3

AHMET ÜNLER
BATUHAN SARITAŞ
MURAT ÖZTÜRK

INSTRUCTOR: ALİ TAMER ÜNAL

Content

Brief Explanation of the Game

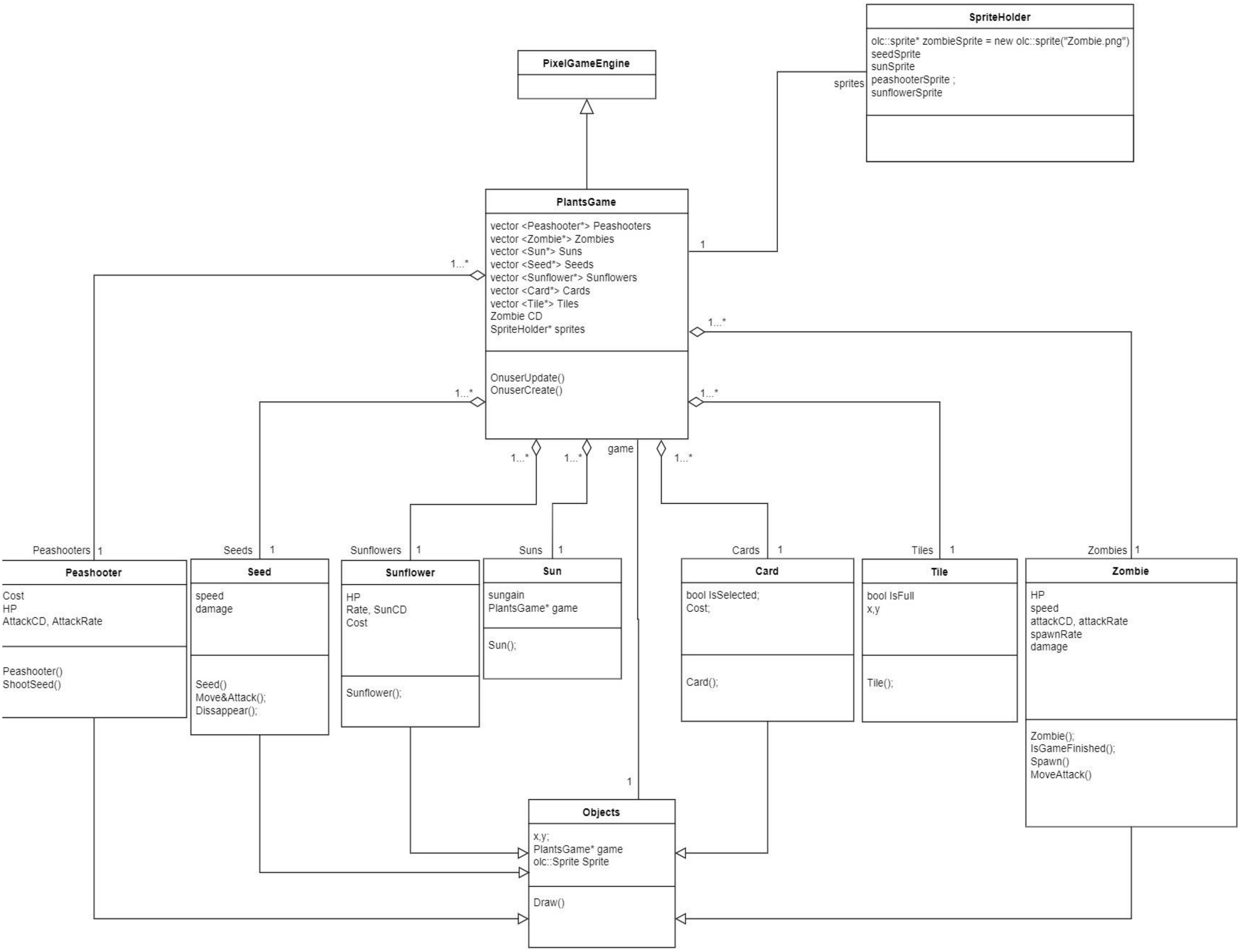
In this semester, as the registered students we were given the duty to develop and reconstruct a game in an object oriented point of view. In order to construct and redesign the game in a proper way, it is crucial to identify the specifications, attributes and the aims of the game.

The game consists of some different objects. Main idea is to keep on defending from the antagonists of the game. So it is important to understand that there Is not a well-defined objective that needs to be completed in order to “win” the game. Hence, it is important to develop an instinct in the player to keep on playing and try to develop his/her record for the game each and every play.

Main content of the game is consisted of sunflowers, peashooters and the zombies as antagonists. Zombies are spawned in some of the places in the map and as antagonists they aim to destroy the creatures (i.e. sunflowers and peashooters) that are spawned by the player, if they manage to pass the defense line of the player, they eventually defeat the player. However as it can be forecasted, the player needs a defense mechanism for the zombies. Sunflowers and peashooters eventually make up this whole defense mechanism. Although the sunflowers does not practically and physically defense the player and its territories, they grow up the suns, which are the monetary unit of the game. In other words, if a player wants to keep on playing and challenging the zombies, it is crucial for him to plant and harvest sunflowers in order to make a sustainable monetary cycle. In addition to a monetary cycle, a player’s first aim is to defend himself/herself from the attacking zombies as mentioned. That issue will be held by the peashooters and the seeds they fire. A peashooter is a defender equipped with seeds (i.e. bullets) that fire its seeds towards the zombies, aim to decrease and eventually finish the life points of the zombies and kill them. These two objects, peashooters and sunflowers can be selected from the cards of them by the player and they can be planted onto the tiles of the gaming map which are predefined.

The game primarily consist of the patterns described above. As it was mentioned, the players’ main aim is to defend themselves as long as possible versus the antagonist characters, zombies. The duration of the game, in other words success rate of the players, will be determined by the choices and the strategies of the players themselves.

Class Diagram



Explanation of Classes and Interrelations

1. PlantsGame

PlantsGame is the main class in the game. All the other classes aggregate into this class. It contains the functions which enables user and the CPU to update the gaming environment. It inherits from the PixelGame Engine, which contains the functions that enable the user to start, pause and stop the game. Lastly, Plants Game class has an association relation with the SpriteHolder class and Objects class.

2. SpriteHolder

SpriteHolder class is the class which holds specific sprites that conation the pictures of characters and lends these sprites to main class by association relation. It can be updated furthermore if new characters etc. are implemented into the game, this class would enable us to easily implement the images of these new implementations as well.

3. Objects

Objects class is the parent class. Its attributes, methods etc. are inherited by other classes such as Peashooter, Zombie, Seed, Card etc. In addition, Object has an association relation with PlantsGame class. All character classes including tile,card etc. which inherit the content of the Objects class use the attributes separately, freely and independently.

4. Peashooter

Peashooter is the main defense mechanism improved for the player to prevent the harms of the zombies. It shoots seeds to the zombies in order to kill them before they reach and either destruct peashooters or sunflowers, or directly defeat the player. This class inherits from the object and has an aggregation relation with the class PlantsGame.

5. Zombie

Zombie class is the main enemy of the gaming environment. They occur randomly with a distribution which will be decided by the game developers and their main focus is to harm the player or the gaming contents which are spawned by the player and make player lose the game. They have life points which are defined under this class, and their life would decrease by the seeds which are threw by the Peashooters. This class inherits from the object and has an aggregation relation with the class PlantsGame.

6. Tile

Tiles are the squares in which the player can deploy the sunflowers or peashooters in order to continue playing the game. Tiles would eventually be an important aspect for player to build up the strategy for the game. This class inherits from the object and has an aggregation relation with the class PlantsGame.

7. Seed

Seeds are the “bullets” of the Peashooters. They are fired by the Peashooters and every time they hit their targets, zombies, they decrease the life points of the zombies. For its properties, seed class can be seen as the other part of the defense mechanism for the player. This class inherits from the object and has an aggregation relation with the class PlantsGame.

8. Sunflower

Sunflowers are the contents which are deployed by the player to the tiles and spawn suns, the monetary unit of the game which will be held as a class below. In order a player to continue playing the game and earn monetary units of the game, sunflowers are essential. This class inherits from the object and has an aggregation relation with the class PlantsGame.

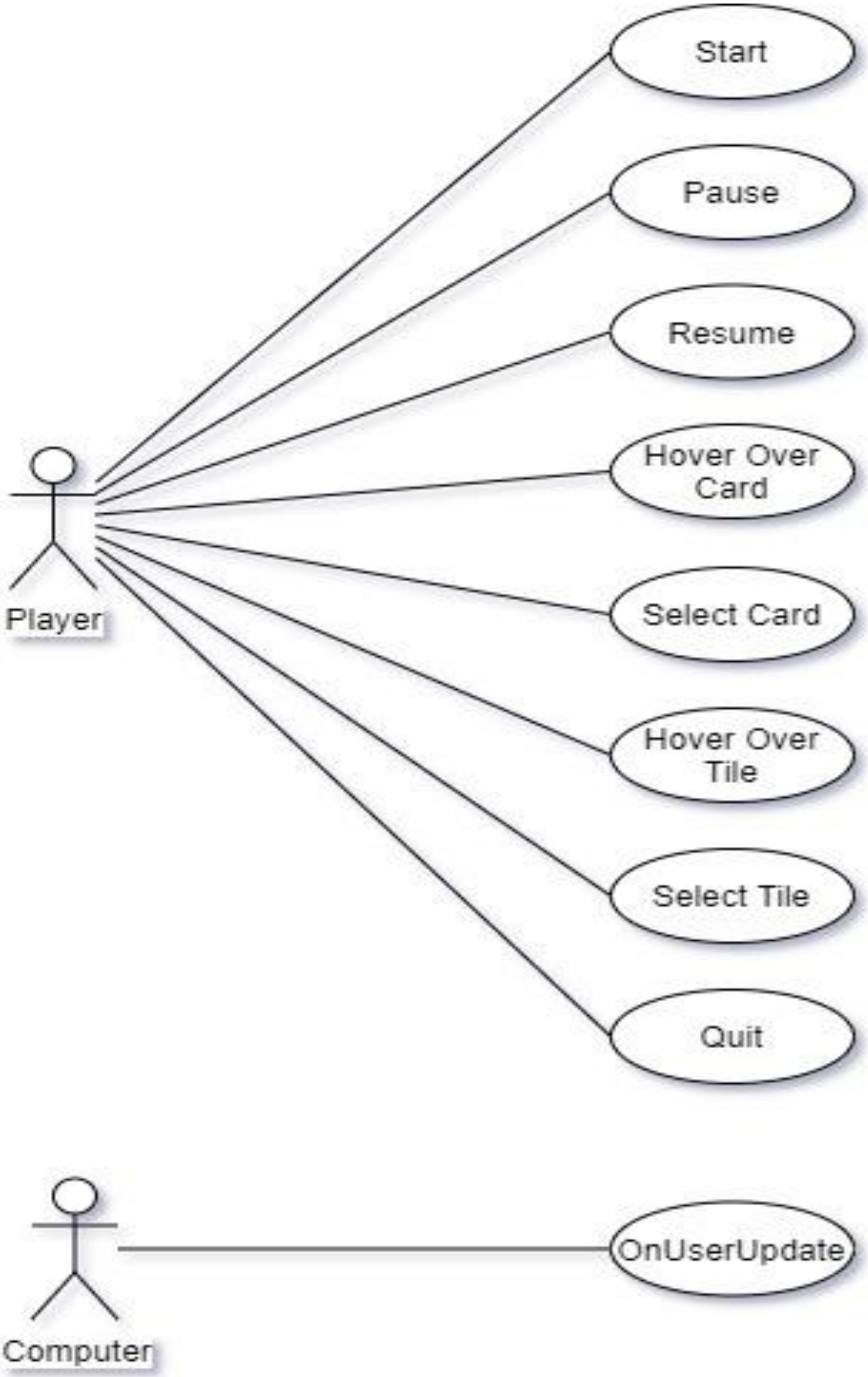
9. Sun

Suns are the monetary units of the game. They are spawned by the sunflowers, which are deployed by the player. In order to deploy new Peashooters or Sunflowers, Suns are essential and needed. Each of the Sunflowers and Peashooters will cost some amount of suns, and will be deployed in exchange of this amount. This class inherits from the object and has an aggregation relation with the class PlantsGame.

10. Card

Card class is the class in which the selection of the Sunflowers or Peashooters are made. Each of these class will have a card in the upper left of the screen and the player would select the one which he/she wants to deploy into the tiles. This class inherits from the object and has an aggregation relation with the class PlantsGame.

Use Case Diagram



Use Case Explanation

There are two external agents in this game, which are Computer and Player.

1. Start

Game starts from the beginning and game screen pops up when player clicks the button.

2. Pause

Game pauses when user clicks to “P” and a pause screen pops up.

3. Resume

Player can resume the game by clicking the related button it in the menu that popped up after his/her pause command.

4. Hover Over Card

When player hovers his/her mouse over the cards on the upper left of the game screen, game detects this movement and surrounds the frame of that specific card with a color of white.

5. Select Card

Player can select one of two cards that are shown on the upper left of the game screen. He/she can choose either a sunflower or a peashooter to deploy onto tiles.

6. Hover Over Tile

When player hovers his/her mouse over the green tiles inside the game screen, game detects this movement and surrounds the frame of that specific tile with a color of white.

7. Select Tile

Player can select a tile if he/she selected a card beforehand. After selection of the tile, game basically deploys the content of the player’s card onto that specific tile. For instance, game deploys a peashooter onto the tile selected if player has selected a peashooter card before this action.

8. Quit

Player can quit the game by clicking “E”. After quitting a menu pops up in which player can either leave the game or start again.

9. OnUserUpdate

This function is a function of internal agent/ computer. Computer keeps track of time and uses this data to picturize the game screen instantaneously. Each picturization means an iteration and thus, a call to onUserUpdate function.