Phase 2, CMPT 276 FALL 2021
Game Title: Pizza Time

Ali Alden 301302901
Francesco Guzzo 301360674
Manpreet Singh Sarna 301416796
Joel Tai - 301347412

Table of Contents

2.1 CLASSES

- THE MAIN GAME WILL CONSISTS OF 4 MAIN SUPERCLASS namely:
    - MAP CLASS
    - CHARACTER CLASS
    - REWARDS CLASS
    - MAIN CLASS

2.1.1 WORLD CLASS:

- The map class generates the map and borders.
- **SUBCLASSES FOR MAP CLASS:**
    - TILE CLASS (places tiles onto the 2d board)
    - TILE CHECKER CLASS (enumerate array values for tile class.)

2.1.2 CHARACTER CLASS:

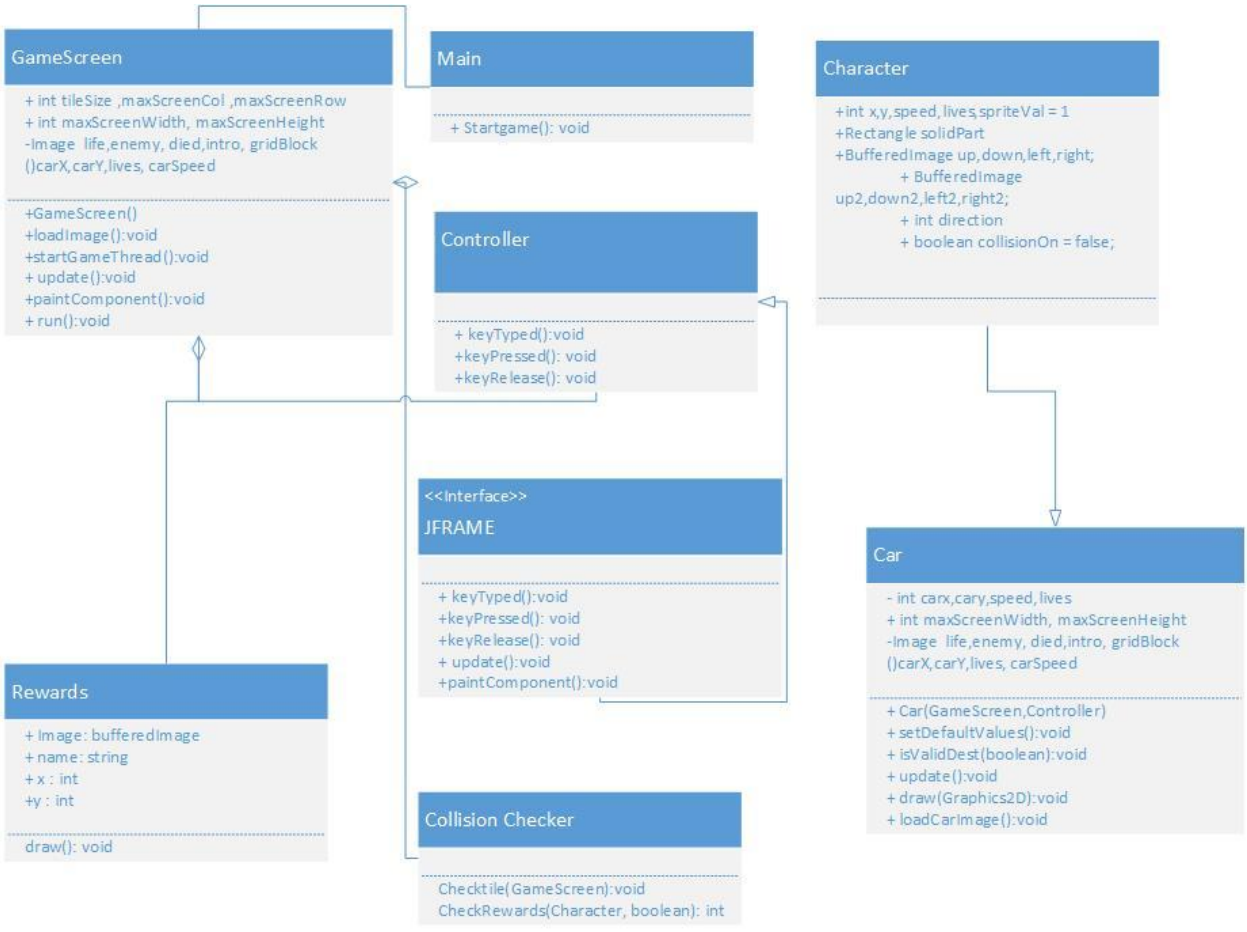
- The character class will generate car and police (player and enemy)
- SUBCLASSES FOR CHARACTER CLASS
    - CAR CLASS
    - POLICE CLASS

2.1.3 REWARDS PACKAGE:

- The reward package will take care of the items
- Subclasses for rewards
    - Pizza Class
    - Wrench class
    - Etc.

2.2 UMLS:

UML CLASSES FOR SYSTEM.:



2.3 **Overall Approach for the Game:**

Game title: Pizza Time
Description: It is a modified version of Pac Man with some changes that make the game more exciting.
Our overall approach of the game is to let the player be inside a confined space and then collect the pizza's as bonus points and along with that be safe from the police cars that are trying to catch the player while he is collecting the pizza and move from starting destination to the ending destination.

Libraries Used:
For our GUI we used java.awt to be the library that helped us in making our GUI in a way that we expected. There were several reasons to choose this library:

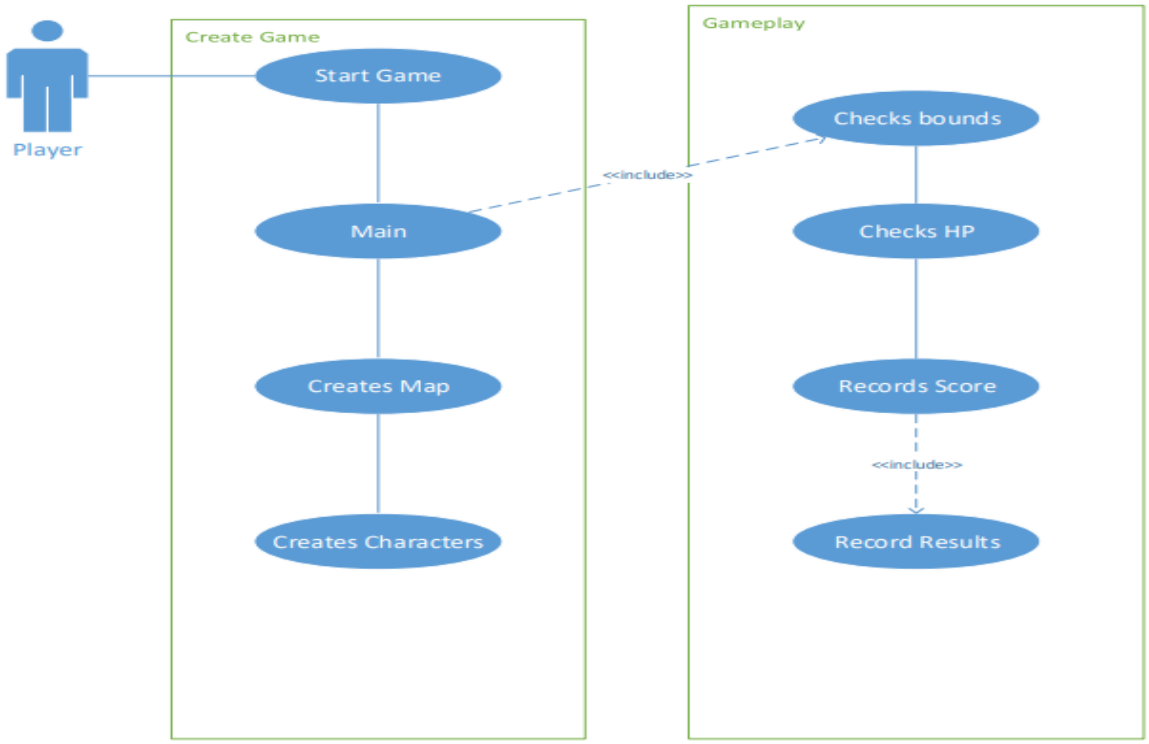
1. It provided us all the classes for creating our user interface for the images and graphics that are present in the game.
2. Java.awt.Color helped us in making our game screen where we used certain elements of the class like graphics for the UI of our main game screen where the operations come into place.
3. The event class allowed us to make the Controls of our game which are really easy to use that include upper arrow,down, left and right arrow for going up, down, left and right.
4. We also used the java.awt.Image class for our Character class that will help us in giving the modifications of the images in our Character Class like how it changes when moved up, down, left and right.
5. We also used Graphics2D class in it for creation of our tile checker whose main purpose is to make the blocks in which the car moves in the form of 18 rows and 24 columns. This will be the main playing area where all the tiles will mean 1 movement of the car which will go on tile by tile.

6. Rewards Class:
7. Enemy Class:

2.4 Use Case Changes

Use Cases Changes: For the player our main user case will be in the following steps:

1. Player will initiate the game where he can play.
2. Then when he runs the game the game starts and the game is created.
3. Then the main of the program will check the bounds of the game, Full HP, initial score being recorded.
4. Then the map is created
5. After that characters are created.
6. And then the game will continue to play until the player loses all its lives. Meaning that the gameplay will be in a loop where whenever the player loses life it will check lives, check the recorded score and as soon as player loses all its lives it will give out the results



2.5 Main Measurements taken to enhance the quality of the code:

1. One of the main measurements we took was to have a short and precise code that is easy to understand.
2. Secondly we added comments in all our logics in order to make the code legible and understandable for all of our team members when they type the code.

2.6 Team Meetings:

Team meetings were meant to get every group member up to speed with what was being done. If a group member had pushed some new code we took the time to go over it so people could understand the changes. The meetings were also used to delegate some tasks to make better use of our time. We would make a meeting for every wednesday from when phase 2 started.

After the first week of meeting one of our main challenges was to try to download maven and get in contact with each other in order to start working on the assignment. Another problem we had was making sure group members knew how to properly use an IDE.

For the second week when our group meeting happened, we are trying to fix the collision detection where there are certain bugs in the code which are kind of difficult to fix which has still resulted in less progress in the work and we are still trying to fix everything up. We are also having issues implementing the logic for the Police enemy so it chases the player using the shortest route.

We were also confused about the map length because of the bugs that are in the code right now and we are trying to fix them as soon as possible.

Coding Side:

A) Bug on collision checker, the whole map should not collide with data.
B) Bug on map tile checker, was squashed 11/05/2021.
C) Updated levels, there are 2 levels for the map now.