

CSE1242 Computer Programming II  
(Spring 2024)  
Term Project  
Due: 06.05.2024 23:59

# Table of Contents

Table of Contents .....	2
Traffic Control Simulator.....	3
Project Objectives.....	3
Project Breakdown .....	4
Game Environment .....	4
RoadTile .....	4
Building.....	5
TrafficLight .....	5
Level Parsing .....	6
Car Movement.....	7
Traffic Control:.....	8
Win and Lose Conditions: .....	9
Example Level Map and Input File .....	10
Level Maps .....	15
Level 1 .....	15
Level 2 .....	16
Level 3 .....	17
Level 4 .....	18
Level 5 .....	19
Bonus Points .....	20
Level Designer .....	20
Markers and Paths .....	21
Submission Instructions.....	22
Demo Sessions.....	22
Details about Project Report .....	22
Notes.....	23

# Traffic Control Simulator

This project will challenge you to build a traffic control simulator game using JavaFX. You will be responsible for creating a graphical user interface, parsing level data from text files, and implementing game logic for car movement and traffic control.

## Project Objectives

- Create a JavaFX traffic simulation game set within a city map, allowing players to manage traffic flow through traffic lights.
- Parse level data from text files to create the game map, including roads, buildings, and traffic lights.
- Implement car objects that spawn from buildings and follow designated paths.
- Implement car movement logic to stop at traffic lights or stop when the car ahead is stopping.
- Design a win condition where a specific number of cars reach their designated destinations.
- Develop traffic control mechanics by allowing players to click on traffic lights and change their colors.
- Implement car collision detection and a limited number of allowed crashes before losing the game.
- For bonus points, create a level designer mode to design maps with roads, buildings, traffic lights. Create road markers and draw paths using them in the level designer.

# Project Breakdown

The project can be divided into several key stages.

## Game Environment

To create the game map, you need to design classes to represent different map elements. Those elements are RoadTile, Building and TrafficLight. You can see the different kinds of map elements on the side menu of the level designer.

### RoadTile

There are four types of RoadTiles. These different road types are essential for creating city roads that can go in any direction and connect in any way. The main difference between road types are their shapes. Type with 0 index is a straight road and with index 1 is a curved road. Type 2 is a tile to put at the intersection of four different road whereas type 3 can be used at the intersection of three roads. RoadTile class takes the road type as an integer parameter and draw the road shape accordingly. It also takes rotation parameter too and draw the shapes considering the rotation as well. You can see the RoadTile types and their rotated versions. You also need to draw a centerline on the roads to distinguish left and right sides of the roads. That part is not visible on the images.

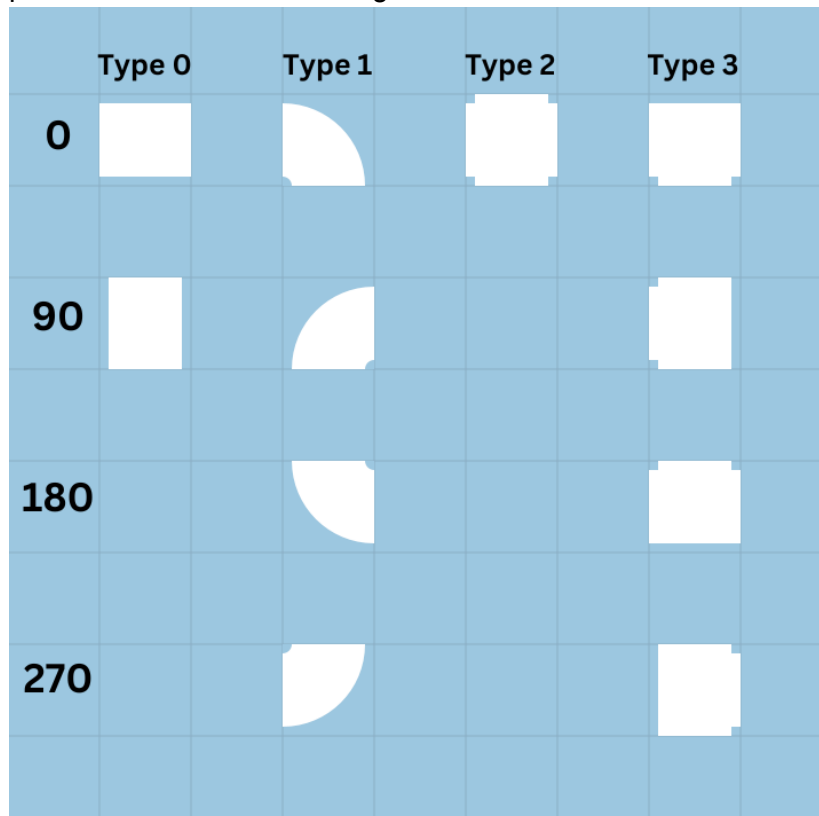


Figure 1: Road tiles

## Building

There are three types of buildings with four different colors. Type 0 and type 1 buildings are bigger buildings representing public buildings in the cities whereas type 3 is a smaller building representing houses. There are also four different colors to make the buildings more diverse. Building class takes type, rotation and color as parameters to draw the buildings. As you will see in the level parsing section, input file includes color index for buildings. It means you need to have an array of four colors and you will give the color at given index to the building. You can define your own colors for the buildings. Other than the shape differences, different type buildings has no difference. They act as starting points and destination points for the cars. You can see the Building types and their rotated versions in the image below.

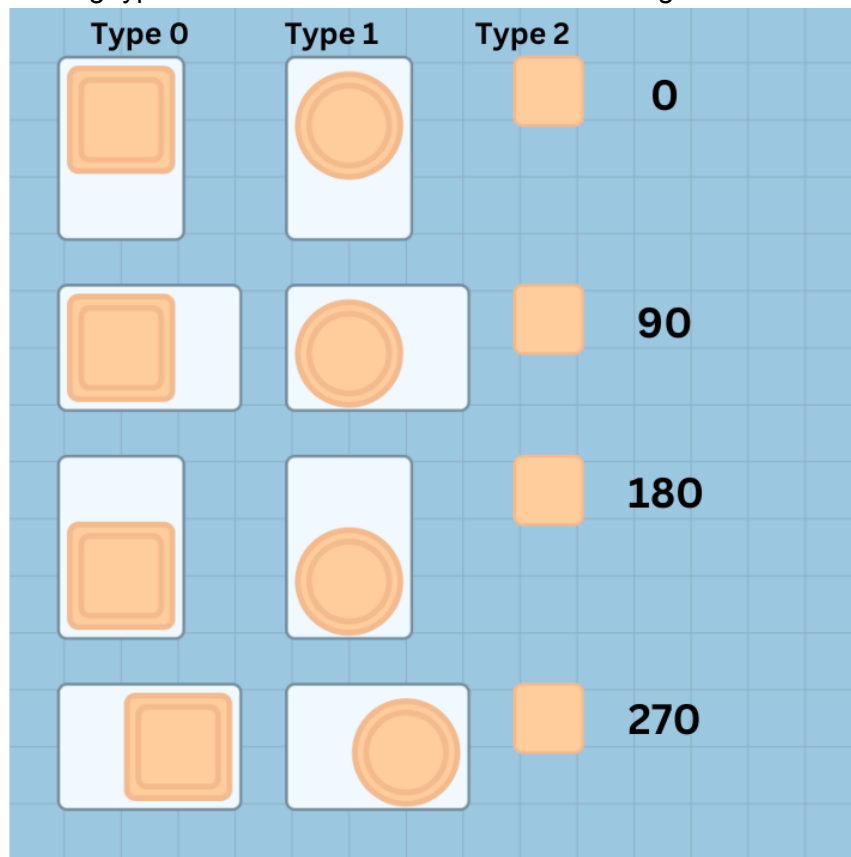


Figure 2: Buildings

## TrafficLight

TrafficLight is a shape that consists of a line and a circle. In the input file the start and end positions to draw the line will be given. The circle needs to be at the center point of the line. TrafficLight should also be able to change color when it's clicked. Red and green are the two color options.

## Level Parsing

To parse a level from an input file, you need to read the level data. Following is the input text format.

```
Metadata 800.0 800.0 15 15 2 50 5
Building 0 0 0 4 2
RoadTile 0 90 5 5
RoadTile 2 90 5 6
RoadTile 0 90 5 7
RoadTile 0 180 4 6
Building 2 180 0 5 8
Building 2 270 2 3 6
Building 1 270 2 9 6
RoadTile 0 0 7 6
RoadTile 0 0 8 6
RoadTile 0 180 6 6
TrafficLight 386.67 346.67 386.67 320.00
Path 0 MoveTo 506.67 346.67
Path 0 LineTo 453.33 336.00
Path 0 LineTo 400.00 336.00
Path 0 LineTo 346.67 336.00
Path 0 LineTo 346.67 336.00
Path 0 LineTo 304.00 336.00
Path 0 LineTo 282.67 336.00
Path 0 LineTo 240.00 336.00
Path 1 MoveTo 293.33 453.33
Path 1 LineTo 282.67 400.00
Path 1 LineTo 282.67 357.33
Path 1 LineTo 282.67 336.00
Path 1 LineTo 282.67 293.33
Path 1 LineTo 293.33 240.00
```

- `Metadata 800.0 800.0 15 15 2 50 5` : The file starts with a Metadata row representing the width, height, number of grid cells in x direction, number of grid cells in y direction and the number of paths this map includes. After that, the final two numbers show the number of cars to arrive at destinations to win the game (50) and number of allowed car accidents for the game to be over (5).
- `Building 0 0 0 4 2` : This line represents a building of type 0 with rotation 0 degrees. The third number represents the color index. You need to have an array of length 4 to keep your colors and you need to draw your buildings according to this color indices. Final two numbers represent the grid cell x and grid cell y indices. This is the upper left position for the element. To get actual positions, you need to calculate it by multiplying these indices with grid cell widths or heights.
- `RoadTile 0 90 5 5` : This line represents a RoadTile of type 0 with rotation 90 degrees. Final two numbers represent the grid cell x and grid cell y indices for the upper left position of the road tile. Again, you need to calculate positions by multiplying these indices with grid cell widths or heights.

- `TrafficLight 386.67 346.67 386.67 320.00` : This line represents a Traffic-Light between two points. The order is x1, y1, x2, y2.
- `Path 0 MoveTo 506.67 346.67` : This line represents a MoveTo path element. This element belongs to path 0, which is the index of the path. Finally there are two double numbers that represent the x and y position.
- `Path 0 LineTo 453.33 336.00` : This line represents a LineTo path element. This element belongs to path 0, which is the index of the path. Finally there are two double numbers that represent the x and y position.

In the game, you need to be able to parse five different level files provided to you. Additionally, you can export and import your own level files.

## Car Movement

You need to implement a method to handle the car spawn and movement logic. When the game starts, cars should be spawned from a random path starting point at random times. Additionally, when cars reach their end positions, they should be removed from the scene. You can use `AnimationTimer` to handle your car movement and traffic logic. It can look like the following code:

```
private void createTraffic() {
    AnimationTimer timer = new AnimationTimer() {
        @Override
        public void handle(long now) {
            update();
        }
    };
    timer.start();
}

private void update() {
    time += 0.16;
    //Implement the logic for cars checking other cars or
lights here
    if(time > 2) {
        if(Math.random() < 0.3) {
            spawnCar();
        }
        time = 0;
    }
}
```

Cars will travel along designated paths at a constant speed using the `PathTransition` class. As they move, they'll continuously check for traffic lights. If a car encounters a red light, it comes to

a complete stop. Additionally, cars will detect stopped cars ahead within a safe distance and adjust their speed accordingly, preventing collisions with cars that are stopped at red lights.

Car crashes can occur if two cars collide when they are both moving. After a crash, cars will remain at the scene for a brief period (0.5 seconds) before being removed. When these cars that had the accident are still in the scene, other moving cars shouldn't crash them. Instead, they should stop at a safe distance and then continue moving after crashed cars are removed from the scene. Also, it's important to note that a separate logic needs to be implemented to prevent crashes from happening at car spawn points.

### Traffic Control:

You need to create a system where players can influence traffic flow by clicking on traffic lights. Clicking a light would change its color (red and green). The cars who arrive near a traffic light stop, as mentioned in the previous section. You can see the following image to understand the traffic system better.



Figure 3: Traffic



## Win and Lose Conditions:

When reading the level file and creating the level, you also get the number of cars that need to reach their destination to win the game. When a car finishes the path, before removing the car from the scene, you need to increase the counter and complete the game successfully if it has enough cars. Same logic applies for car crashes. Before removing the cars, you need to increase the crash counter and end the game if it reaches the defined number. These counts should be shown at the upper right corner of the map. After a successful level, a new level should start when the next level button is clicked. We will give 5 different levels. After the final level, a success screen should be shown.

## Example Level Map and Input File

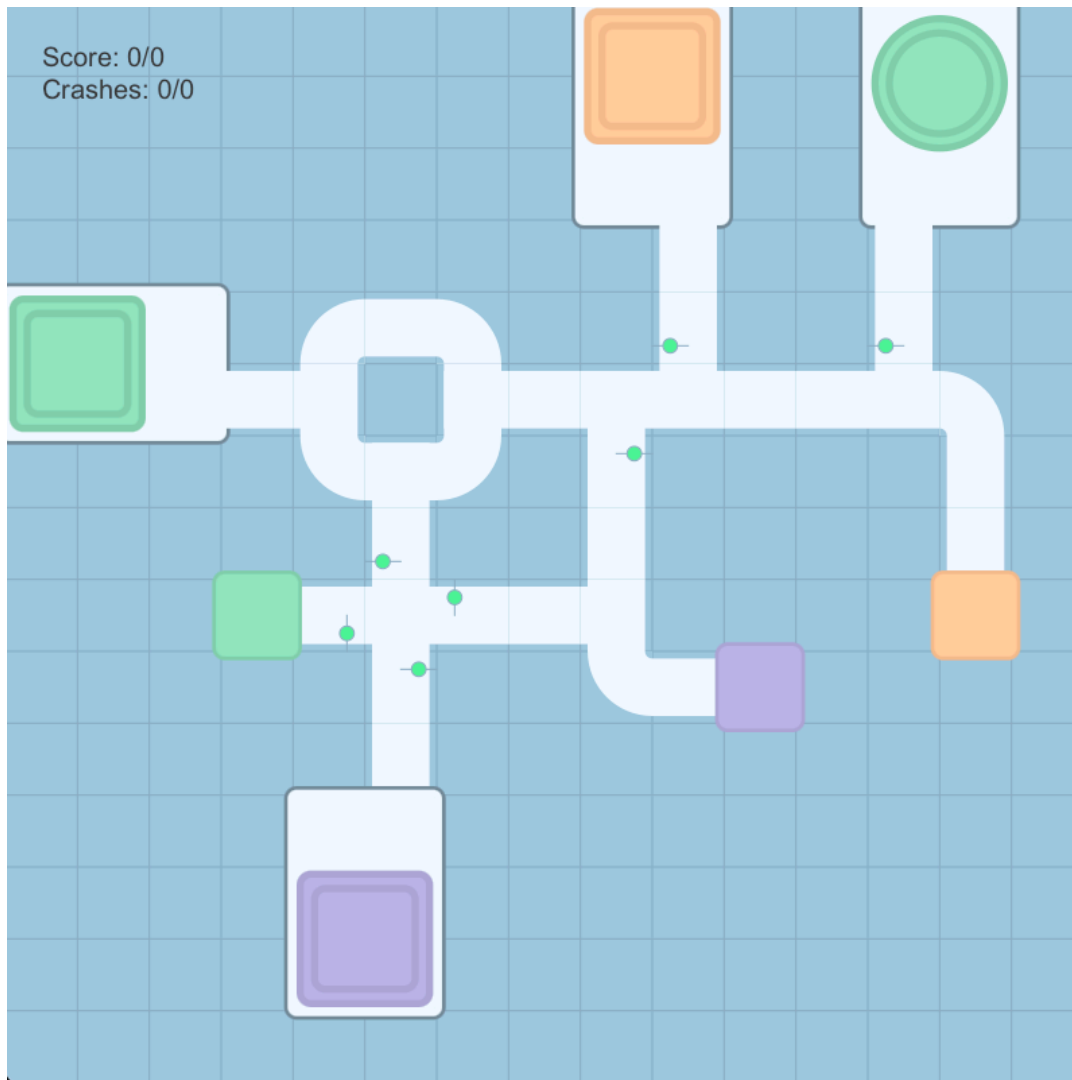


Figure 4 : Example Level

You can see the input file content below.

```
Metadata 800.0 800.0 15 15 7 100 10
Building 0 90 1 0 4
RoadTile 0 180 3 5
RoadTile 0 180 5 4
RoadTile 3 0 5 6
RoadTile 1 0 6 4
RoadTile 3 90 4 5
RoadTile 1 90 4 4
RoadTile 0 90 5 7
RoadTile 2 90 5 8
RoadTile 0 180 4 8
```

RoadTile 0 180 6 8  
RoadTile 1 180 4 6  
RoadTile 1 270 6 6  
RoadTile 0 270 5 9  
RoadTile 0 270 5 10  
Building 2 270 1 3 8  
Building 0 180 2 4 11  
RoadTile 3 270 6 5  
RoadTile 0 0 7 5  
RoadTile 3 0 8 5  
RoadTile 0 90 8 6  
RoadTile 0 90 8 7  
RoadTile 3 90 8 8  
RoadTile 0 180 7 8  
RoadTile 1 180 8 9  
RoadTile 0 180 9 9  
Building 2 180 2 10 9  
RoadTile 3 180 9 5  
RoadTile 0 270 9 4  
Building 0 0 0 8 0  
Building 1 0 1 12 0  
RoadTile 0 90 9 3  
RoadTile 0 90 12 3  
RoadTile 0 90 12 4  
RoadTile 3 180 12 5  
RoadTile 1 0 13 5  
RoadTile 0 0 10 5  
RoadTile 0 0 11 5  
RoadTile 0 90 13 6  
RoadTile 0 90 13 7  
Building 2 90 0 13 8  
TrafficLight 333.33 426.67 333.33 453.33  
TrafficLight 253.33 453.33 253.33 480.00  
TrafficLight 320.00 493.33 293.33 493.33  
TrafficLight 293.33 413.33 266.67 413.33  
TrafficLight 480.00 333.33 453.33 333.33  
TrafficLight 506.67 253.33 480.00 253.33  
TrafficLight 666.67 253.33 640.00 253.33  
Path 0 MoveTo 453.33 133.33  
Path 0 LineTo 496.00 186.67  
Path 0 LineTo 496.00 240.00  
Path 0 LineTo 496.00 282.67  
Path 0 LineTo 464.00 282.67  
Path 0 LineTo 442.67 282.67  
Path 0 LineTo 442.67 304.00  
Path 0 LineTo 442.67 346.67  
Path 0 LineTo 442.67 400.00

Path 0 LineTo 442.67 442.67  
Path 0 LineTo 442.67 464.00  
Path 0 LineTo 442.67 480.00  
Path 0 LineTo 447.67 498.67  
Path 0 LineTo 461.33 512.33  
Path 0 LineTo 480.00 517.33  
Path 0 LineTo 506.67 517.33  
Path 1 MoveTo 666.67 133.33  
Path 1 LineTo 656.00 186.67  
Path 1 LineTo 656.00 240.00  
Path 1 LineTo 656.00 282.67  
Path 1 LineTo 613.33 282.67  
Path 1 LineTo 560.00 282.67  
Path 1 LineTo 517.33 282.67  
Path 1 LineTo 496.00 282.67  
Path 1 LineTo 464.00 282.67  
Path 1 LineTo 442.67 282.67  
Path 1 LineTo 400.00 282.67  
Path 1 LineTo 357.33 282.67  
Path 1 LineTo 357.33 266.67  
Path 1 LineTo 352.33 248.00  
Path 1 LineTo 338.67 234.34  
Path 1 LineTo 320.00 229.33  
Path 1 LineTo 293.33 229.33  
Path 1 LineTo 266.67 229.33  
Path 1 LineTo 248.00 234.34  
Path 1 LineTo 234.34 248.00  
Path 1 LineTo 229.33 266.67  
Path 1 LineTo 229.33 282.67  
Path 1 LineTo 186.67 282.67  
Path 1 LineTo 133.33 240.00  
Path 2 MoveTo 133.33 293.33  
Path 2 LineTo 186.67 304.00  
Path 2 LineTo 229.33 304.00  
Path 2 LineTo 229.33 320.00  
Path 2 LineTo 234.34 338.67  
Path 2 LineTo 248.00 352.33  
Path 2 LineTo 266.67 357.33  
Path 2 LineTo 282.67 357.33  
Path 2 LineTo 304.00 357.33  
Path 2 LineTo 320.00 357.33  
Path 2 LineTo 338.67 352.33  
Path 2 LineTo 352.33 338.67  
Path 2 LineTo 357.33 320.00  
Path 2 LineTo 357.33 304.00  
Path 2 LineTo 400.00 304.00  
Path 2 LineTo 442.67 304.00

Path 2 LineTo 464.00 304.00  
Path 2 LineTo 496.00 304.00  
Path 2 LineTo 517.33 304.00  
Path 2 LineTo 560.00 304.00  
Path 2 LineTo 613.33 304.00  
Path 2 LineTo 656.00 304.00  
Path 2 LineTo 677.33 304.00  
Path 2 LineTo 693.33 304.00  
Path 2 LineTo 704.65 308.69  
Path 2 LineTo 709.33 320.00  
Path 2 LineTo 709.33 346.67  
Path 2 LineTo 709.33 400.00  
Path 3 MoveTo 720.00 453.33  
Path 3 LineTo 730.67 400.00  
Path 3 LineTo 730.67 346.67  
Path 3 LineTo 730.67 320.00  
Path 3 LineTo 725.66 301.33  
Path 3 LineTo 712.00 287.67  
Path 3 LineTo 693.33 282.67  
Path 3 LineTo 677.33 282.67  
Path 3 LineTo 656.00 282.67  
Path 3 LineTo 613.33 282.67  
Path 3 LineTo 560.00 282.67  
Path 3 LineTo 517.33 282.67  
Path 3 LineTo 517.33 240.00  
Path 3 LineTo 517.33 186.67  
Path 3 LineTo 506.67 133.33  
Path 4 MoveTo 293.33 613.33  
Path 4 LineTo 304.00 560.00  
Path 4 LineTo 304.00 506.67  
Path 4 LineTo 304.00 464.00  
Path 4 LineTo 304.00 442.67  
Path 4 LineTo 304.00 400.00  
Path 4 LineTo 304.00 357.33  
Path 4 LineTo 320.00 357.33  
Path 4 LineTo 338.67 352.33  
Path 4 LineTo 352.33 338.67  
Path 4 LineTo 357.33 320.00  
Path 4 LineTo 357.33 304.00  
Path 4 LineTo 400.00 304.00  
Path 4 LineTo 442.67 304.00  
Path 4 LineTo 464.00 304.00  
Path 4 LineTo 496.00 304.00  
Path 4 LineTo 517.33 304.00  
Path 4 LineTo 560.00 304.00  
Path 4 LineTo 613.33 304.00  
Path 4 LineTo 656.00 304.00

Path 4 LineTo 677.33 304.00  
Path 4 LineTo 677.33 282.67  
Path 4 LineTo 677.33 240.00  
Path 4 LineTo 677.33 186.67  
Path 4 LineTo 720.00 133.33  
Path 5 MoveTo 186.67 453.33  
Path 5 LineTo 240.00 464.00  
Path 5 LineTo 282.67 464.00  
Path 5 LineTo 304.00 464.00  
Path 5 LineTo 346.67 464.00  
Path 5 LineTo 400.00 464.00  
Path 5 LineTo 442.67 464.00  
Path 5 LineTo 442.67 480.00  
Path 5 LineTo 447.67 498.67  
Path 5 LineTo 461.33 512.33  
Path 5 LineTo 480.00 517.33  
Path 5 LineTo 506.67 517.33  
Path 6 MoveTo 560.00 506.67  
Path 6 LineTo 506.67 496.00  
Path 6 LineTo 480.00 496.00  
Path 6 LineTo 468.69 491.31  
Path 6 LineTo 464.00 480.00  
Path 6 LineTo 464.00 464.00  
Path 6 LineTo 464.00 442.67  
Path 6 LineTo 442.67 442.67  
Path 6 LineTo 400.00 442.67  
Path 6 LineTo 346.67 442.67  
Path 6 LineTo 304.00 442.67  
Path 6 LineTo 282.67 442.67  
Path 6 LineTo 240.00 442.67

## Level Maps

### Level 1

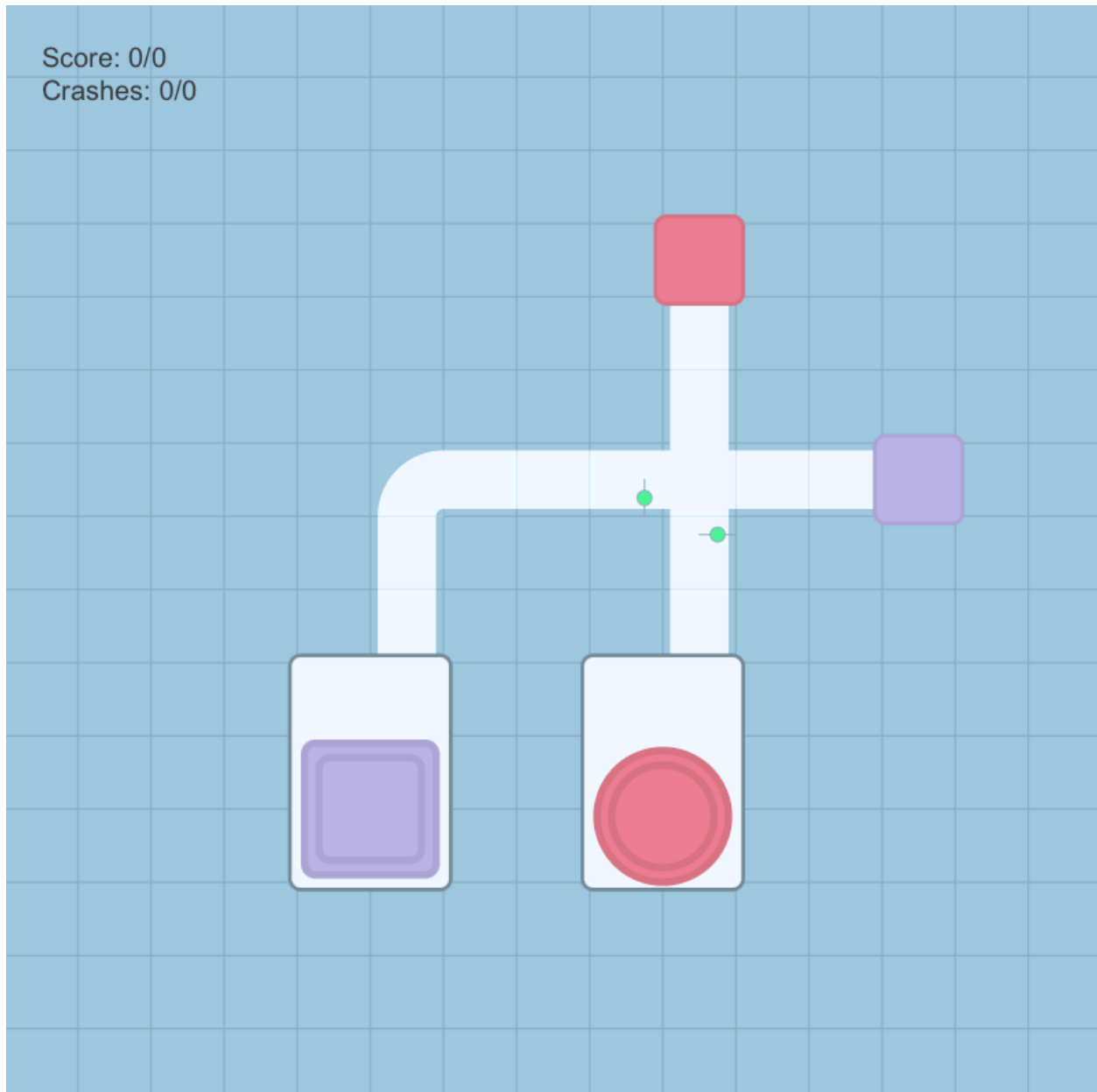


Figure 5: Level 1 Map

## Level 2

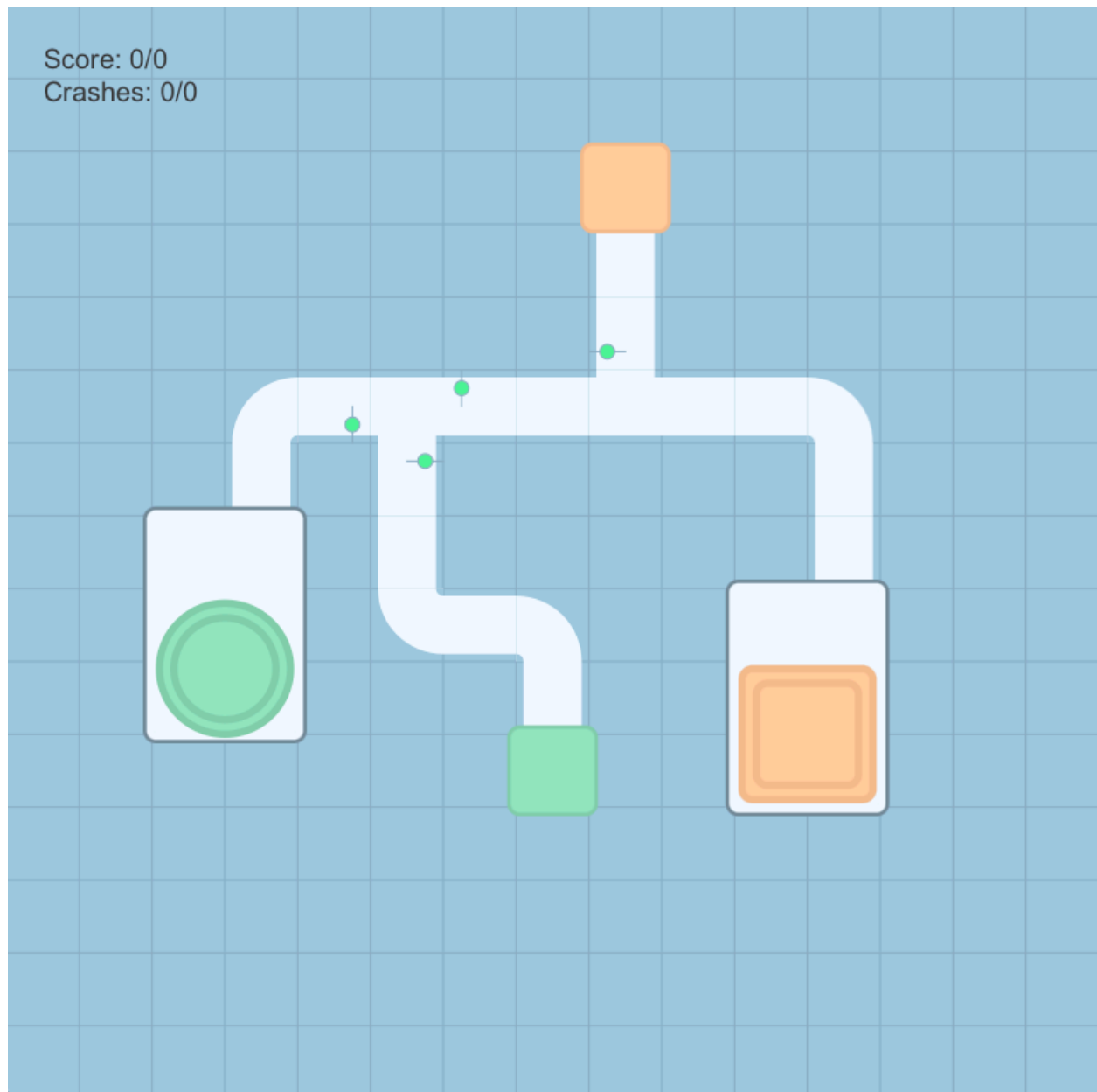


Figure 6: Level 2 Map



## Level 3

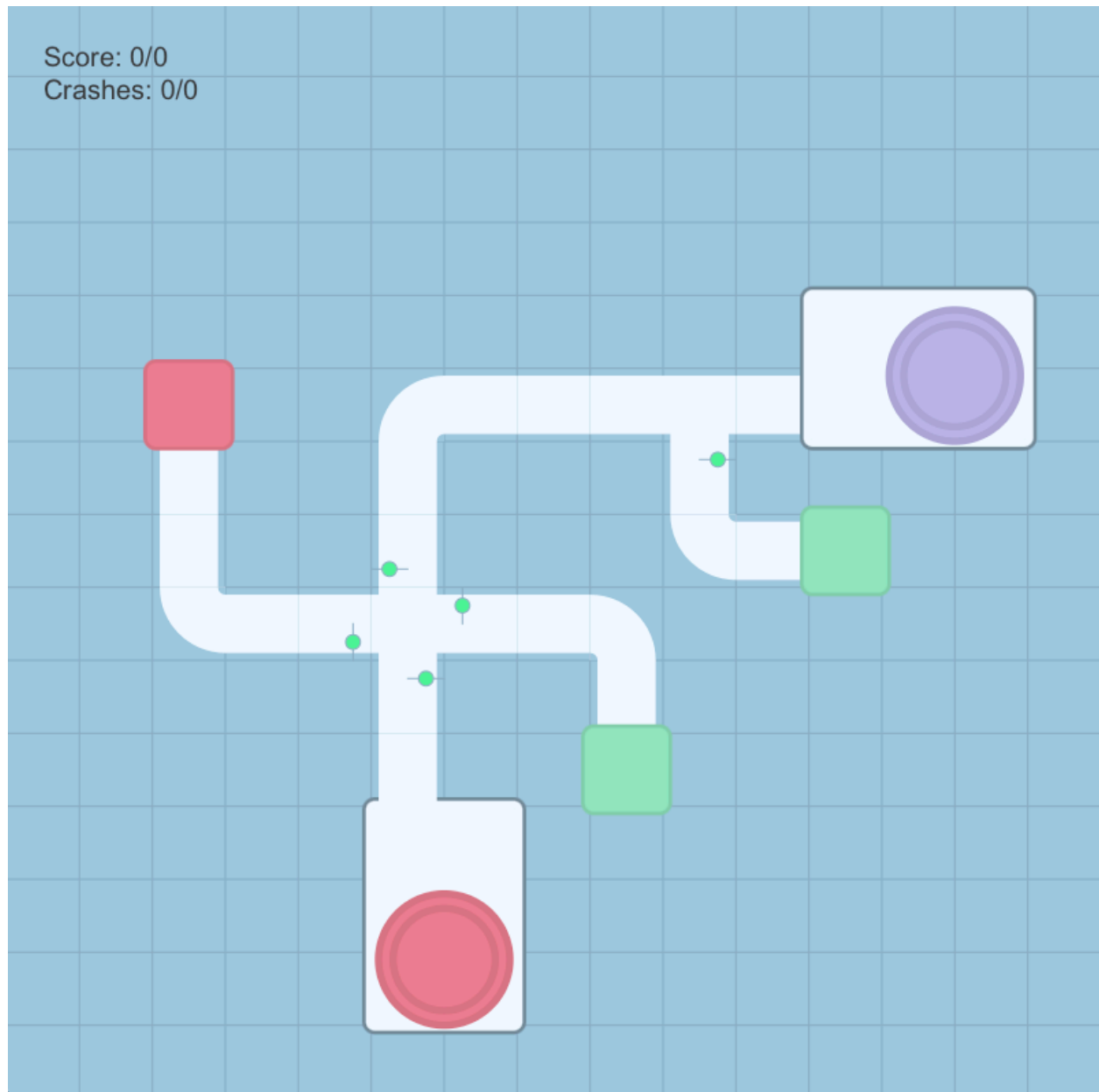


Figure 7: Level 3 Map

## Level 4

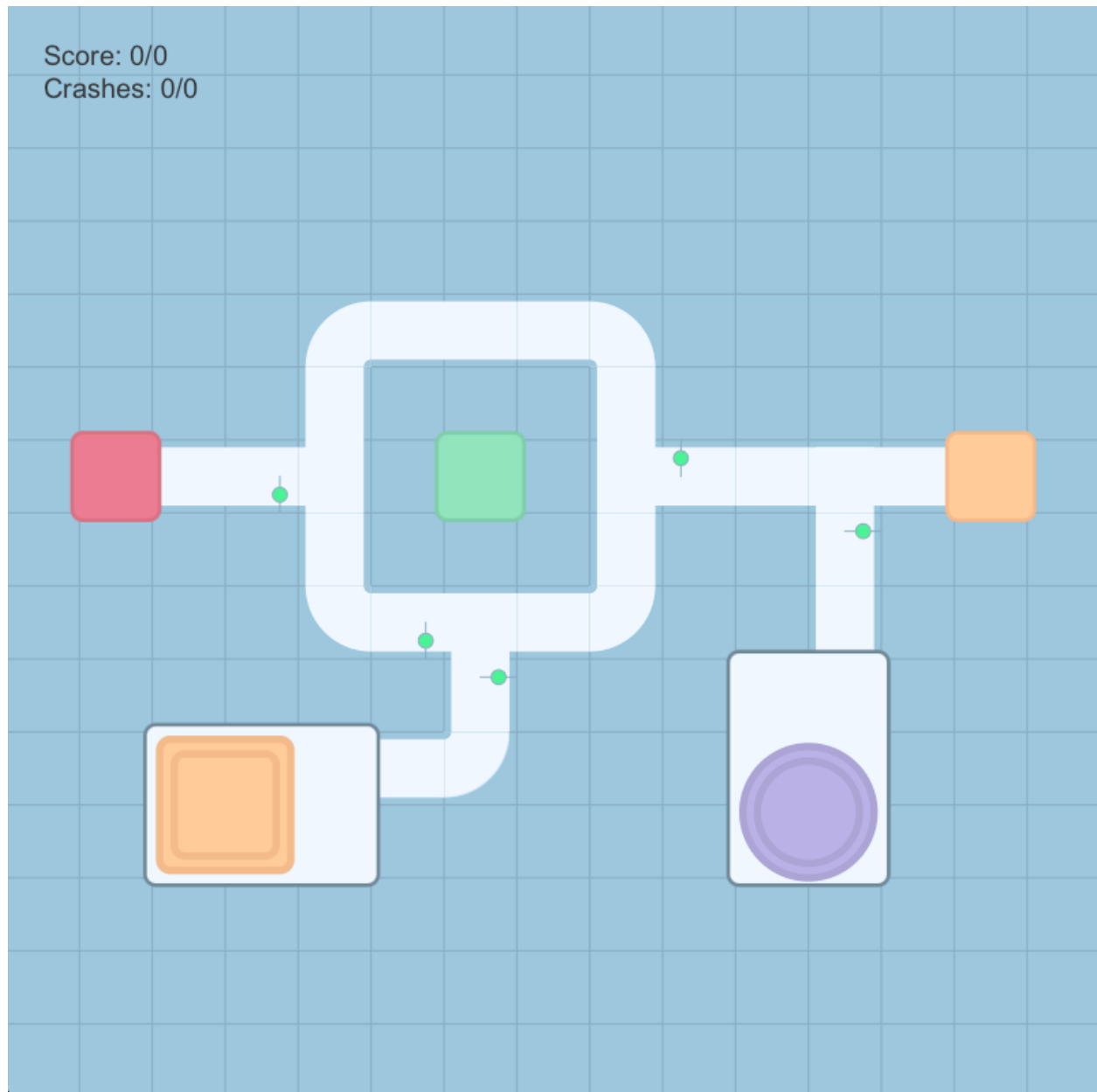


Figure 8: Level 4 Map

## Level 5



Figure 9 : Level 5 Map

## Bonus Points

### Level Designer

You can design a mechanism for players to create the level map by selecting a map element and clicking on the grid to place the elements. Elements need to be shown on the side of the scene as a menu. Additionally, there should be a rotate button and 4 color buttons in the side menu. You also need to implement a path creation mechanism for cars to follow. To do that, players click on the road markers and connect them to form a path. Finally, you need to develop functions to write level data to a text file containing map element types and positions. You should add path point position information to the text file as well.



Figure 10: Level Designer

## Markers and Paths

Markers are elements that allow the player to draw a path in the level designer. Building and RoadTile classes also define which positions to put markers on them. RoadTiles have gray markers whereas Buildings can either have gray or colored markers. Colored markers represent starting points for the paths. A map can have multiple paths. You can see the image below to understand the marker and path structure.



Figure 11: Markers and Paths

## Submission Instructions

- 1) The due date for the project is **06/05/2024**. You are supposed to work in groups of 3 people.
- 2) Please zip and submit your files using filename Student1Number\_Student2Number\_Student3Number\_Project.zip (ex: 150120042\_150118123\_150717015\_Project.zip) to Canvas system (under Assignments tab). Your zip file should contain the following files:
  - a) The commented source code of your project.
  - b) A 5-10 pages long project report that contains UML diagram of your project, implementation details, and screenshots.

## Demo Sessions

You will have demo sessions the week after submission. The exact time and date will be announced later.

You should demonstrate what you have done in 15-20 minutes. You should also answer some questions about your implementation. Each group member will be assessed separately.

## Details about Project Report

- 1) Your project reports are to be typed with normal sizes (Ex: Times New Roman 12pt.).
- 2) Your report must have a cover page with the following information:
  - a) Title
  - b) Project Name
  - c) Authors (IDs, Names, and Surnames)
  - d) CSE1242 Computer Programming II, Spring 2024
  - e) Date Submitted: **May 06, 2024**
- 3) Firstly, you should provide a section named as “Problem Definition” and briefly describe the problem or the game in your project in 1-2 paragraphs with your own words.
- 4) Then, you should add a section named as “Implementation Details” and provide the UML diagrams of your project. In this section, you should also describe how you design and implement the project in more detail.
  - a) Additionally, you may provide information about
    - i) which parts are complete/incomplete in your project?
    - ii) what are the difficulties you have encountered during the implementation?
    - iii) what are the additional functionalities of your project added by your team?
- 5) Then, you should add a section named as “Test Cases” and this section should contain the results of your testing phase. You should provide the screenshots of your project execution for the given test cases and add explanations about them. Filling this part only with screenshots is not a feasible solution.
- 6) These are the minimum requirements for your project report. You can add more.
- 7) The most important part about writing a project report is using your own words without copy-pasting from the Internet or the project document. Please show your own work.

## Notes

- 1) Write a comment at the beginning of each program to explain the purpose of the program.
- 2) Write your names and student IDs as a comment.
- 3) Include detailed comments to explain your actions. Since this is the term project, lack of comments will have an effect on your final grade.
- 4) Select meaningful names for your variables and class names.
- 5) In case of any form of copying and cheating on solutions, you will get FF grade from the course! You should submit your own work. In case of any forms of cheating or copying, both giver and receiver are equally culpable and suffer equal penalties. All types of plagiarism will result in FF grade from the course.
- 6) No late submission will be accepted.