CMPE 160: INTRODUCTION TO OBJECT ORIENTED PROGRAMMING
Project 1

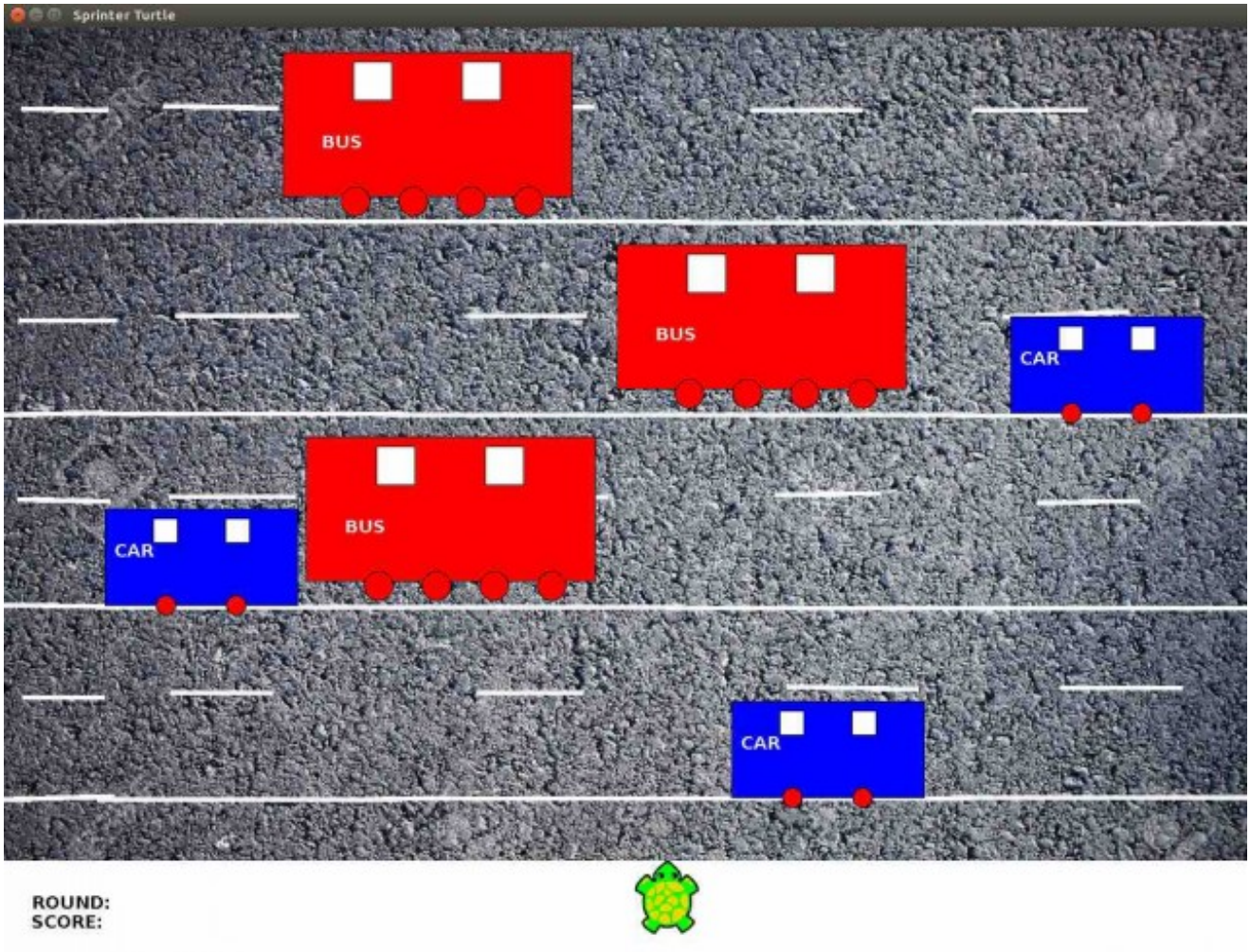
Deadline: 20/04/2015, 23:59.

In this project, you are expected to develop the game detailed below.

1 Description

You are expected to implement a game called ”Sprinter Turtle”. In the Sprinter Turtle game, the user controls a turtle by using arrow keys on the keyboard. The aim of the user is to get the turtle across the street over and over again, during that time, keep the turtle from getting hit by a car.

- The game starts with the turtle sitting in the safe zone.
- The vehicles are not controllable by the user. They travel within their own lanes. Furthermore, their creation, types, lanes and directions are decided randomly.
- The user controls the turtle using arrow keys. The turtle can move forward, backward, to its right and left.
- In every round that turtle completes, vehicles get faster. This means that starting from the safe zone, when turtle reaches across the street and comes back to the safe zone, vehicle instances will be moved with an increased margin.
- Whenever the turtle gets hit by a vehicle, the game finishes and the final score is calculated. The score is calculated based on the number of rounds that the user completes. Faster rounds must be more rewarding in terms of score.
- The leaderboard will keep the 10 highest scores. This requires you to get the user name before the game starts and keep the names and corresponding highest scores in a file that gets updated whenever a new high score is achieved.



2 Details

In the source code that is provided to you, there are two packages: visualization and runnable.

- The package visualization contains one abstract class Vehicle and an interface BoardIntf. You are expected to design and implement:
    1. The Board class, that implements the provided interface and serves as the board for the game, just as we did in PS5.
    2. Car and Bus classes, that must extend the provided abstract class. The instances of these classes will differ in size, color, and number of wheels they have.
- The package runnable contains Main.java and as usual, Main class contains the controller methods.
    - You are provided with the main loop of the game. Every iteration of this loop should be considered as a frame in your game. Therefore, in each frame you need to control vehicles, check for collisions, etc. In addition, if the game should proceed, then you need to update the objects, accordingly. Your main loop may look like as follows:

```
while true do
 collision ← checkForCollision()
 if collision then
 gameOver()
 else
 v ← createVehicle()
 allVehicles ← allVehicles.add(v)
 moveVehicles()
 removeVehicles() //Remove the ones that left screen
 updateScore()
 end if
end while
```

- As mentioned above, the creation, type, lane and direction of the vehicle must be decided randomly. The following code snippet, for example, creates a car object with 1% probability.

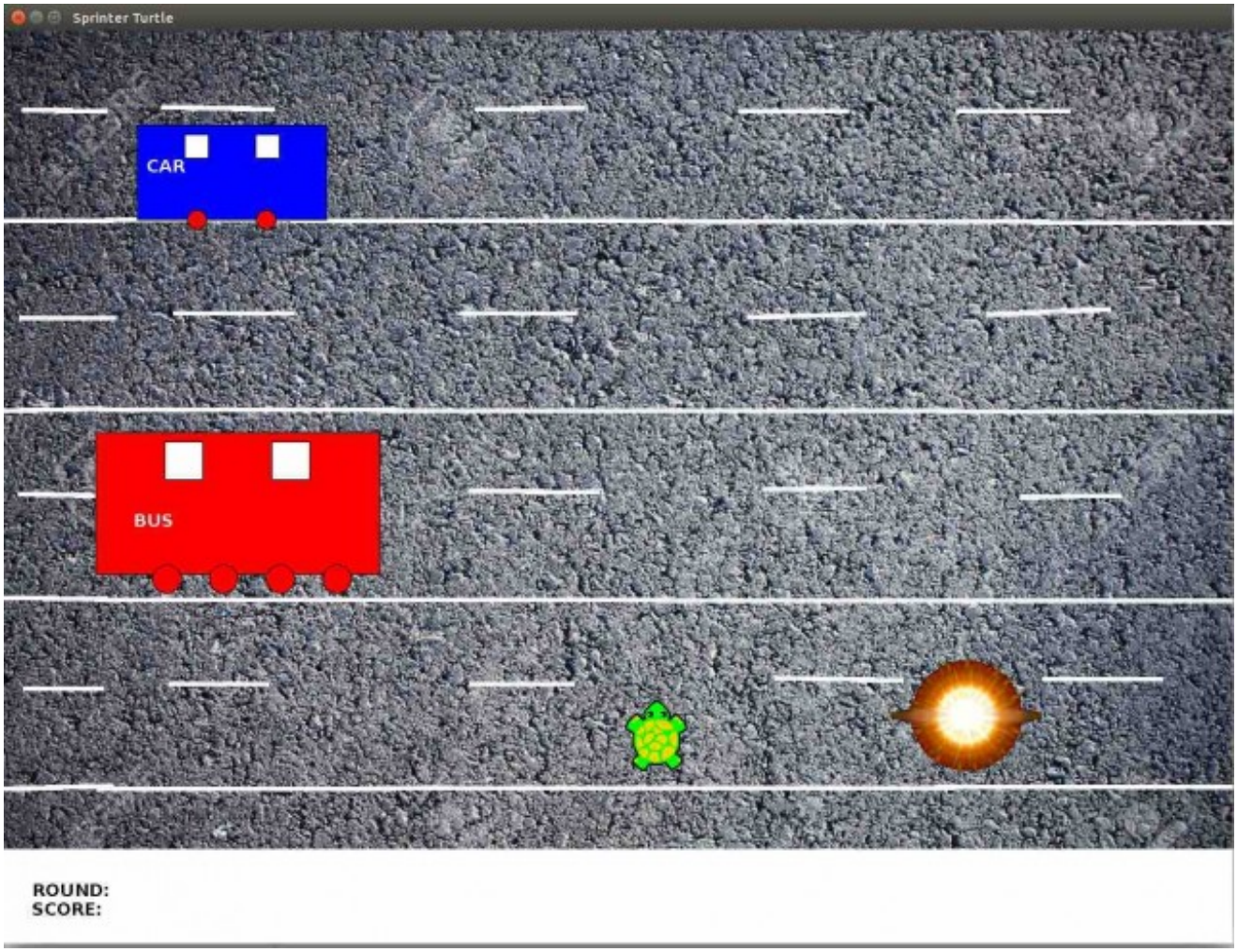
```
double createNew = Math.random();
if (createNew < 0.01){  //Create new vehicles with 1% probability
    Vehicle randomVehicle = new Car(0,0,0,0);
```

- IMPORTANT: Try to put as small code as possible inside the main method. Create other methods or classes, preferably, and call them in you main method. Main method should be quite comprehensible.

3 Bonus

Fully implemented, thoroughly tested and nicely documented additional features will receive bonus points. These features include:

- Implementing the ”Fire!” feature: This feature enables the turtle to fire its gun to eliminate a car using ”WASD” keys. The turtle may use its fire ability only 3 times during a game. For example, when the user hits D key on the keyboard, the car to the right of the turtle gets demolished, and a new car gets created from the starting position of the lane. An example screen shot is given in the following figure.



- Implementing the ”Meteor Hit” functionality: From time to time, a meteorite falls onto the road. The location that the meteorite will hit should be decided randomly. In order to enhance the gameplay, the users must be alerted 5 seconds before the meteorite hits. Therefore, you are expected to paint a

bright yellow dot on the screen where the meteorite will hit within the following 5 seconds. Every object, vehicles and turtle, within a radius of 10 units will disappear; so that, if the turtle gets hit, then the game finishes and if the vehicles get hit, they just explode and the game continues.

4 Notes

- You are going to use ACM's JTF Graphics library. A nice way to study the functionalities that this library provides might be to follow the tutorials here http://cs.stanford.edu/people/eroberts/jtf/.
- An initial implementation of the project is given at the end of the fifth PS. Use that code as your guide when building on top of the code we provide here.
- Pay attention to your indentation and and documentation of the source code. Write corresponding test methods.