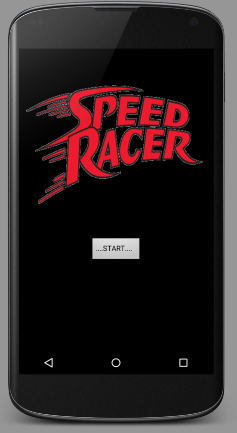
Speed Racer

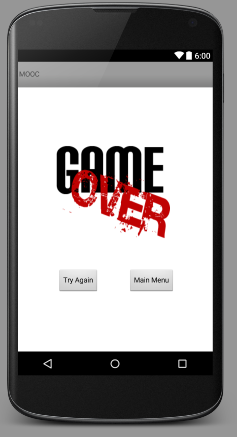
Speed Racer is a game targeted for android platform developed using only native android libraries. That means, it haven’t used any game engines or third party libraries for the development of game. In the development of Speed Racer, Object Oriented Programming (OOP) are thoroughly applied throughout the project in order to keep the architecture of the project extendible and scalable.



This is the main screen of the Street Racer. Once the player press the Start button, it will load the game. In the game, the player has to drive the car along the road while avoiding various obstacles he may encounter on the road. When the player exceed some specific distance, the next level would be loaded and the environment and other vehicles would be different compared to previous level. Following image shows the first level of the game. Player has to tap on left half or right half of the screen in order to maneuver the car.

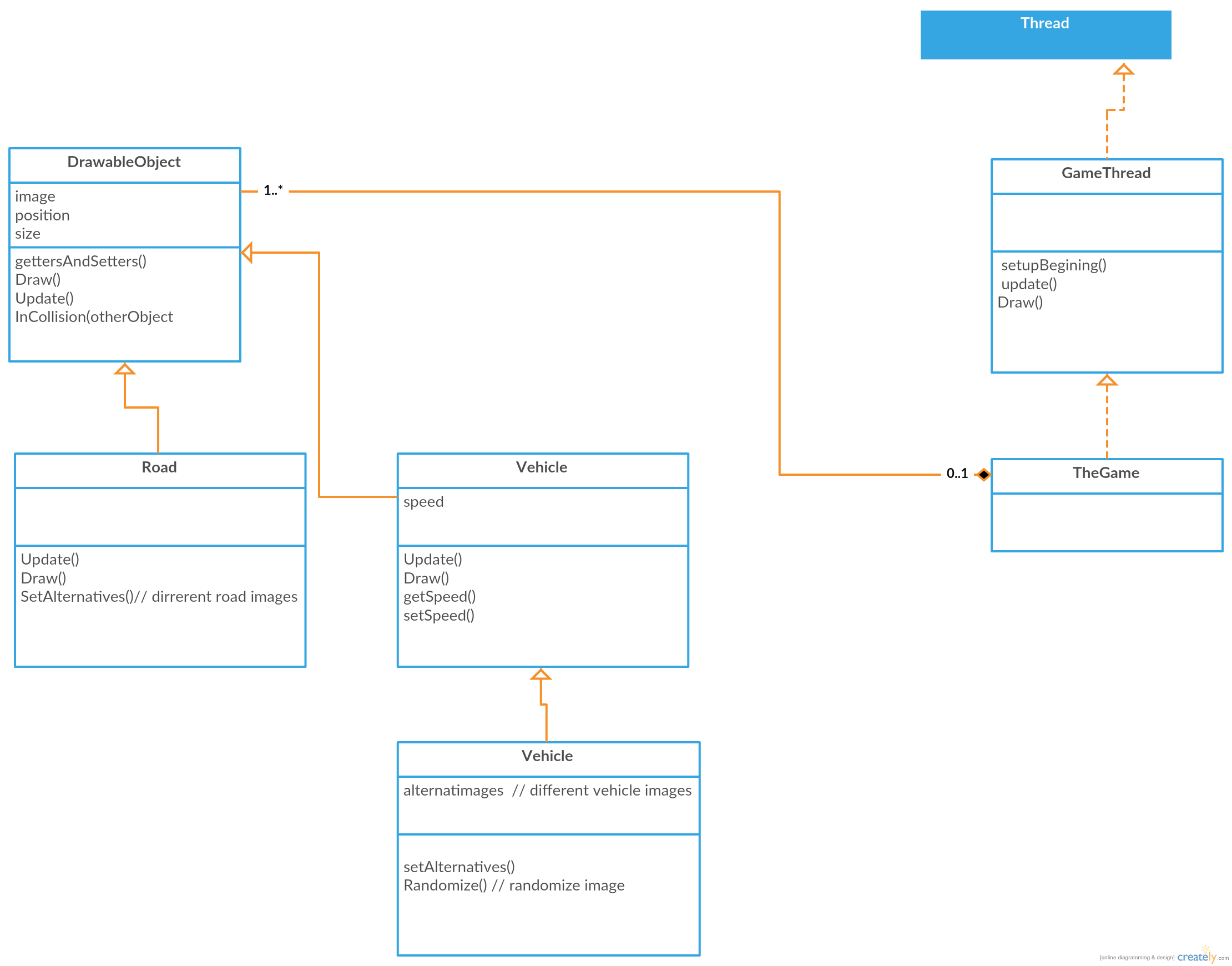


Whenever the player car collide with another object or another vehicle on the road, game over menu will be appeared. In the game over screen, there are two buttons to go to main menu or retry the game.



Architecture of the game

According to the requirement, Speed Racer have been extended from the given project. As roughly depicted in the following diagram, the main game thread is defined in TheGame class and all other major object types are extended from the DrawableObject class. All other classes which has a drawable component has been extended from DrawableObject class.



# TheGame.java

This class is the main class which handles the flow of the game. Update and Draw methods of this class is called per each frame of the game. All other class which has Update and Draw methods should be called from this class.

# DrawableObject.java

This is a custom class which did not included in the given source code. This is the base class for all classes which include some image drawing onto the screen. In this project, Vehicle, RoadCar, Road classes has been extended from this class as they need to draw some image on the screen. This class handles setting up the source image, position and scaling of the image to be drawn onto screen. By extending other draw able classes from this class, we do not have to think about drawing part of those subclasses. Which is a good thing we can achieve by the use of OOP concept Inheritance.

# Vehicle.java

This class has been extended from DrawableObject class and used to create the player car object. Update method of this class has been overridden from the base class because the behavior of player car is different from generic Drawable objects.

## Road.Java

This class is also extended from the DrawableObject class as it has to draw the Road texture onto the screen. But, there is a slight difference in drawing and updating this class. When drawing the image, it draws the texture twice as Road1, Road2 in following image because in that way, the illusion of seamless road can be achieved. In update method, the y position of the road object is moved down because, even though the player car should be moving forward and the road should be stationary in real world, the car image should keep fixed with the screen. So, in order to make the car look like moving forward, other objects in the environment are moved backward. Also, when the position of the Road object is out of the screen, it again replaces itself in a upper y position so then it could be drawn onto screen like a continuous infinite road. This is the code snippet which does replacing the road position

**if** (getPosY()>getHeight()/2+GameView.*screenHeight*){  
 setPosY(GameView.*screenHeight*/2);  
}



When player proceeds in the level system of the game, the texture of the road will be changed so the player can get a different feeling.

## RoadCar.java

This class defines other vehicles player meet in the road. The update method in this class works similar to the update method in the Road because it get replaced in a forward position once it leaves the screen. Also, this class contain several images of vehicles and as it get repositioned in in a forward position, texture also randomized using the available set of textures and the player can meet different vehicles on the road.