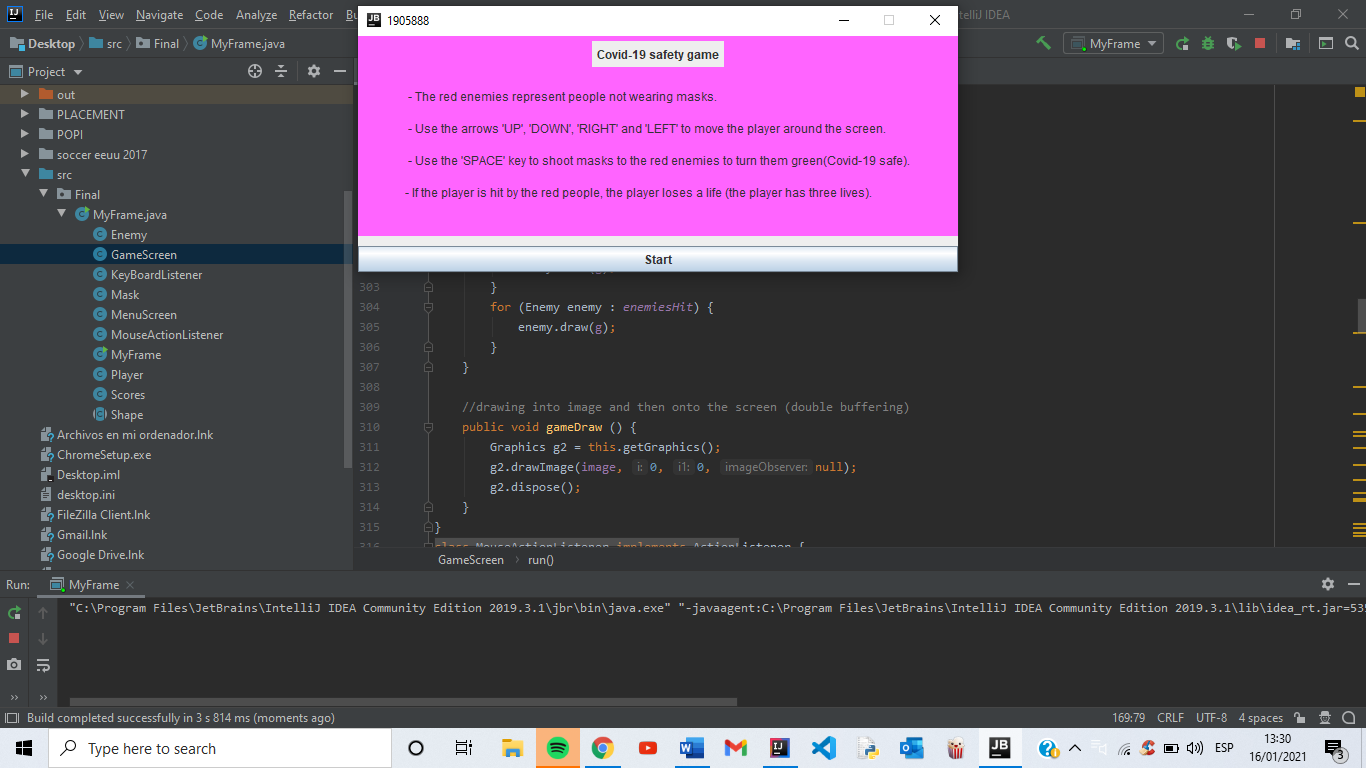
REPORT CE203 ASSIGNMENT 2

The game consists of a player shootings masks at red enemies who are not wearing masks. If the player hits an enemy with a mask, the enemy will turn green. Each time you give someone a mask is one point added to the score.

If the player is hit/touched by a red enemy, the player will lose a life and turn to a lighter colour (dark grey for 2 lives, grey for 1 life and white for the last life).

The player has three lives and the game does not end until the player dies. The more enemies you hit the harder it gets since it is harder to see the red enemies.

When the program is first run a menu screen pops up with some instructions and a “START” button:



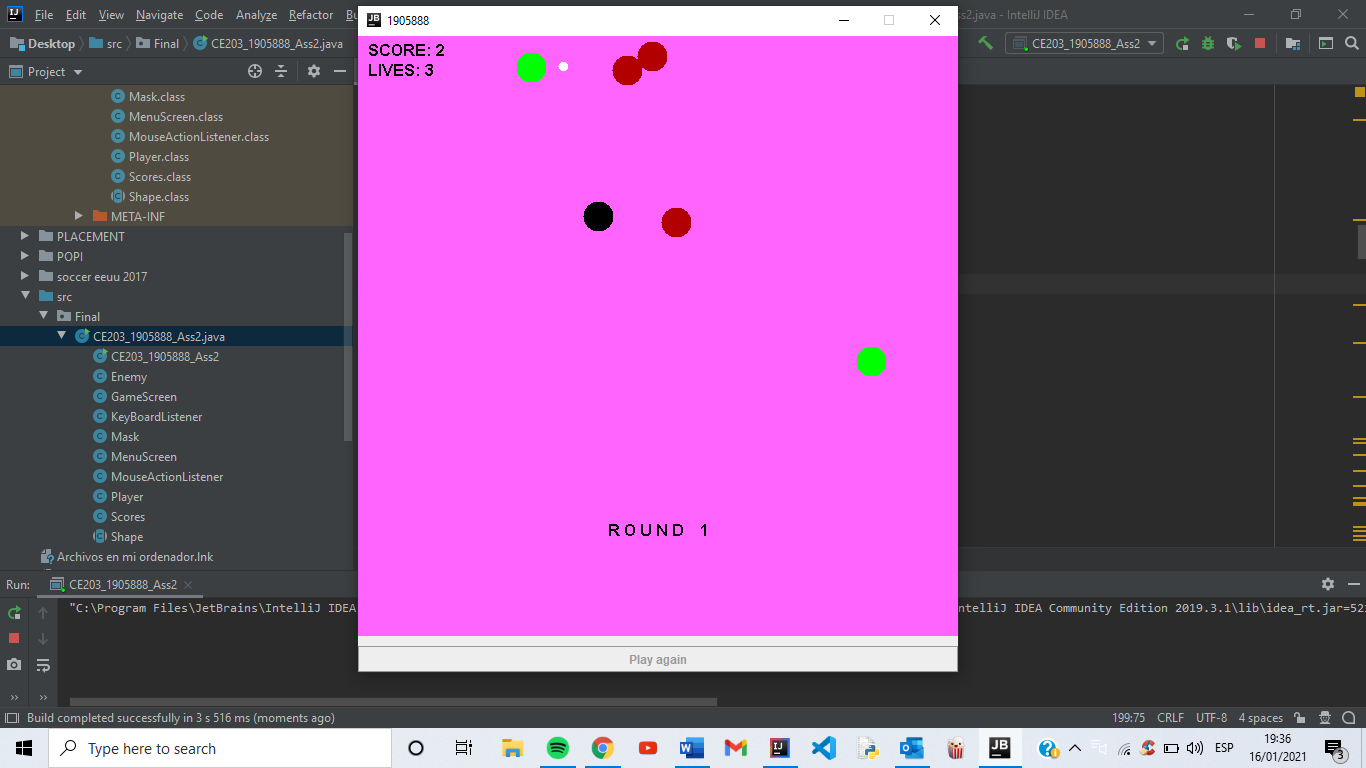
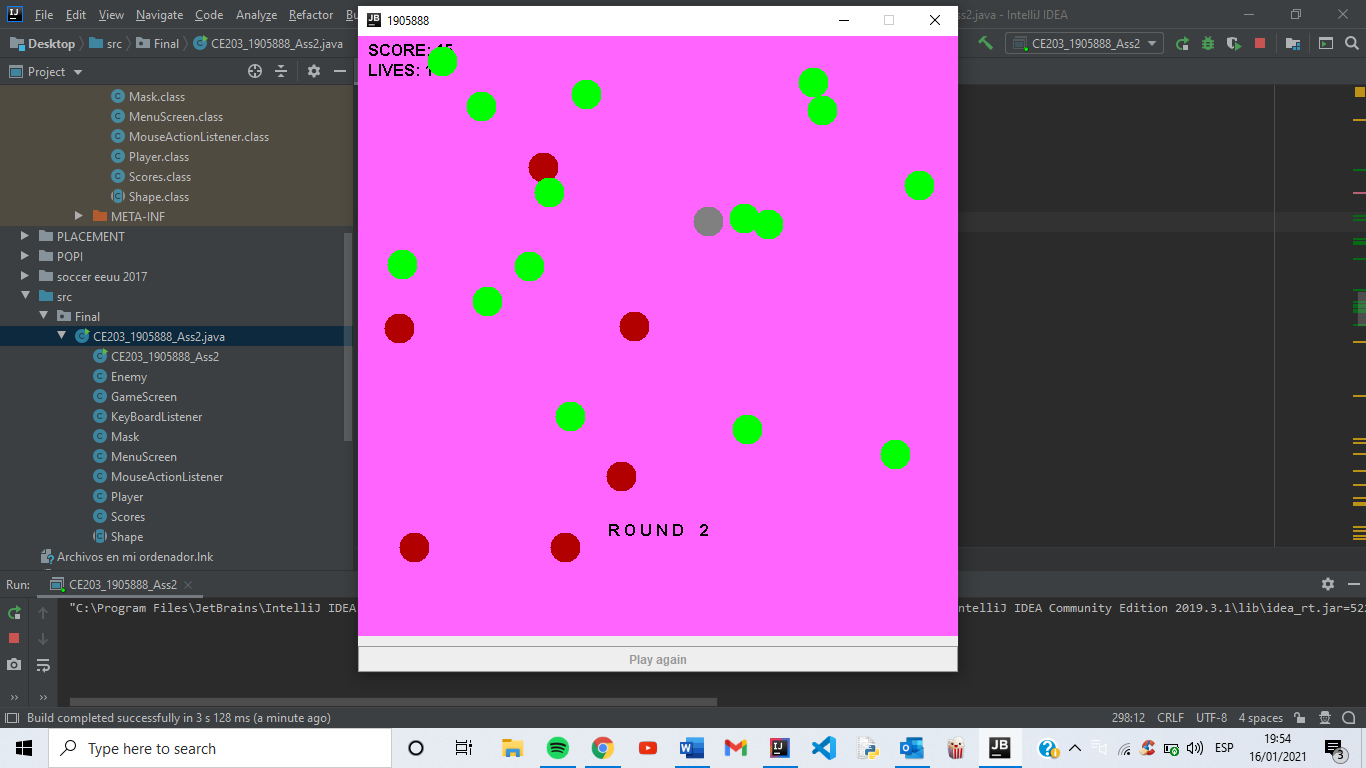
When the “START” button is clicked the game screen pops up with the player in the middle and some enemies coming from the top of the screen.

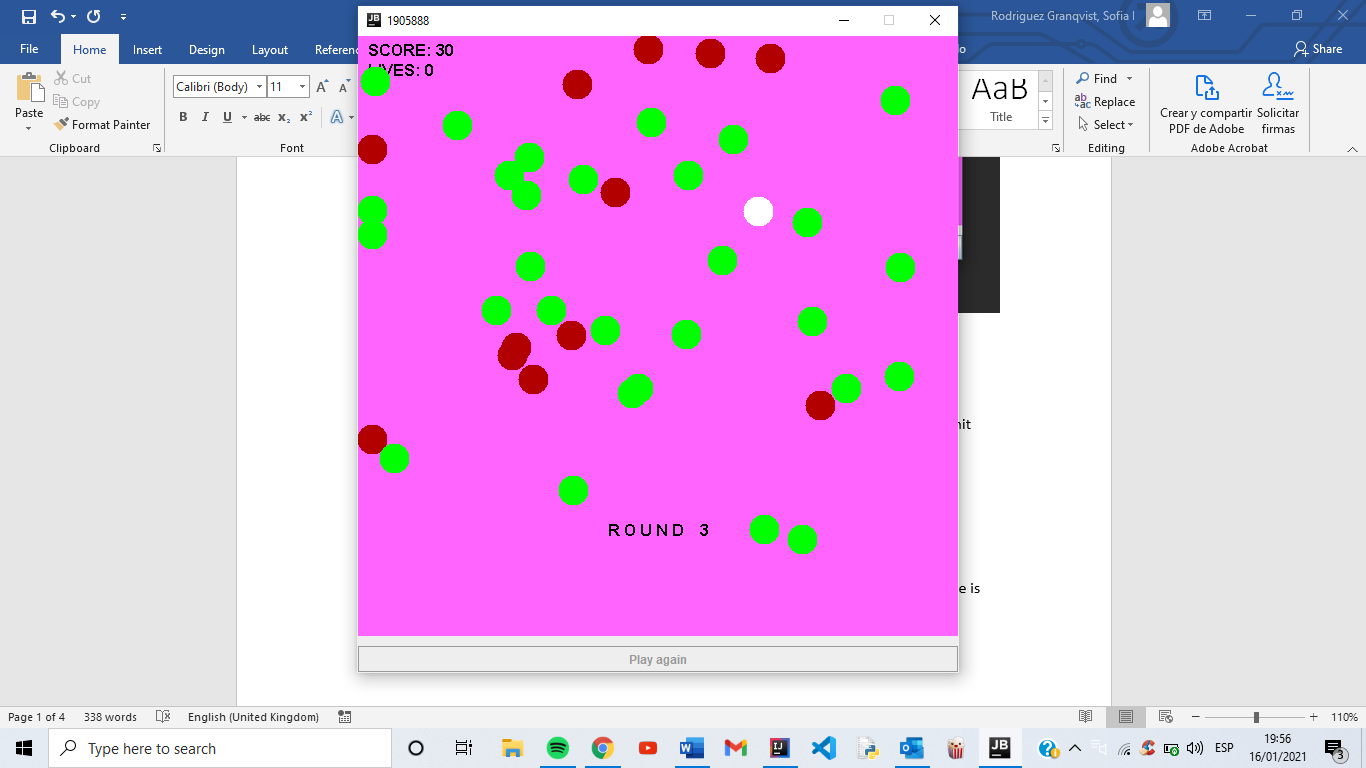
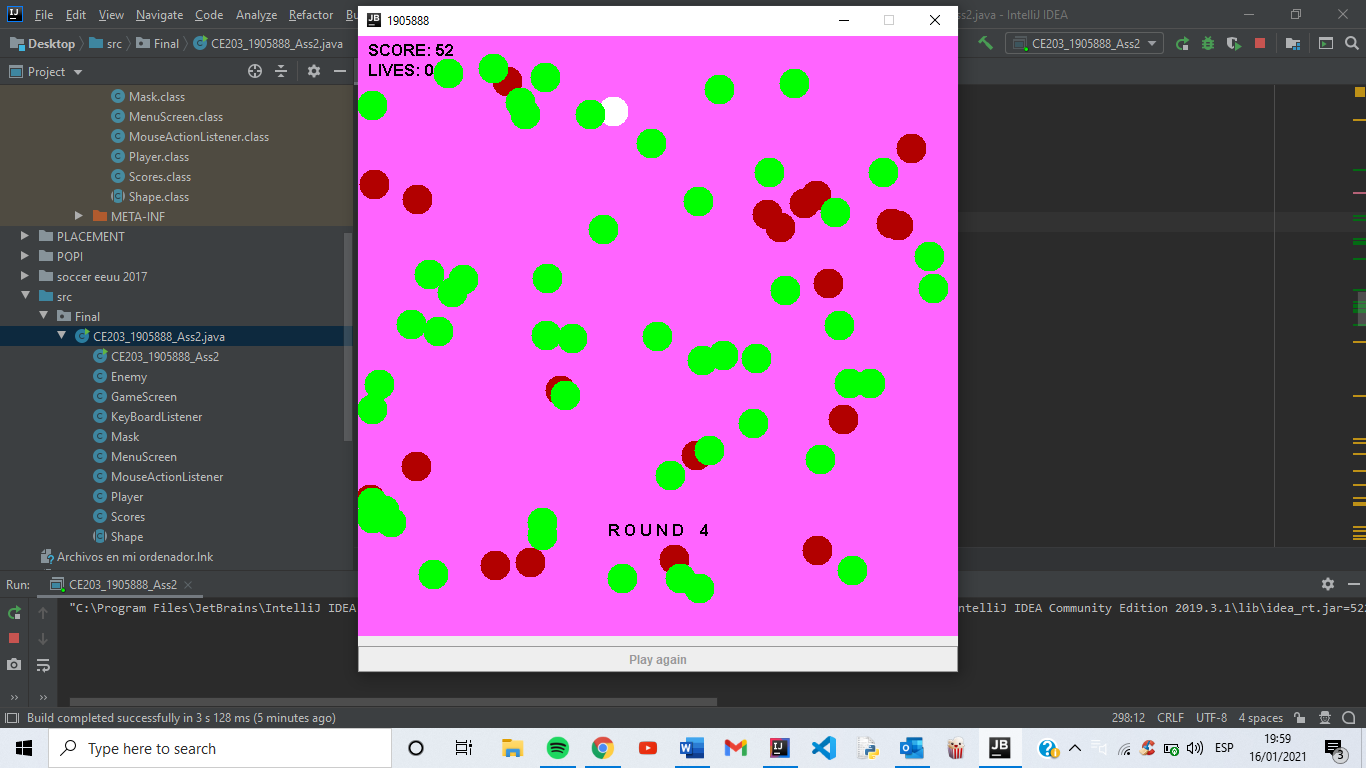
Depending on the round there is a fixed number of red enemies on screen so when one is hit another one appears. An enemy also is removed if it hits the player.

There are 4 rounds, each one more difficult with a higher number of enemies on screen.

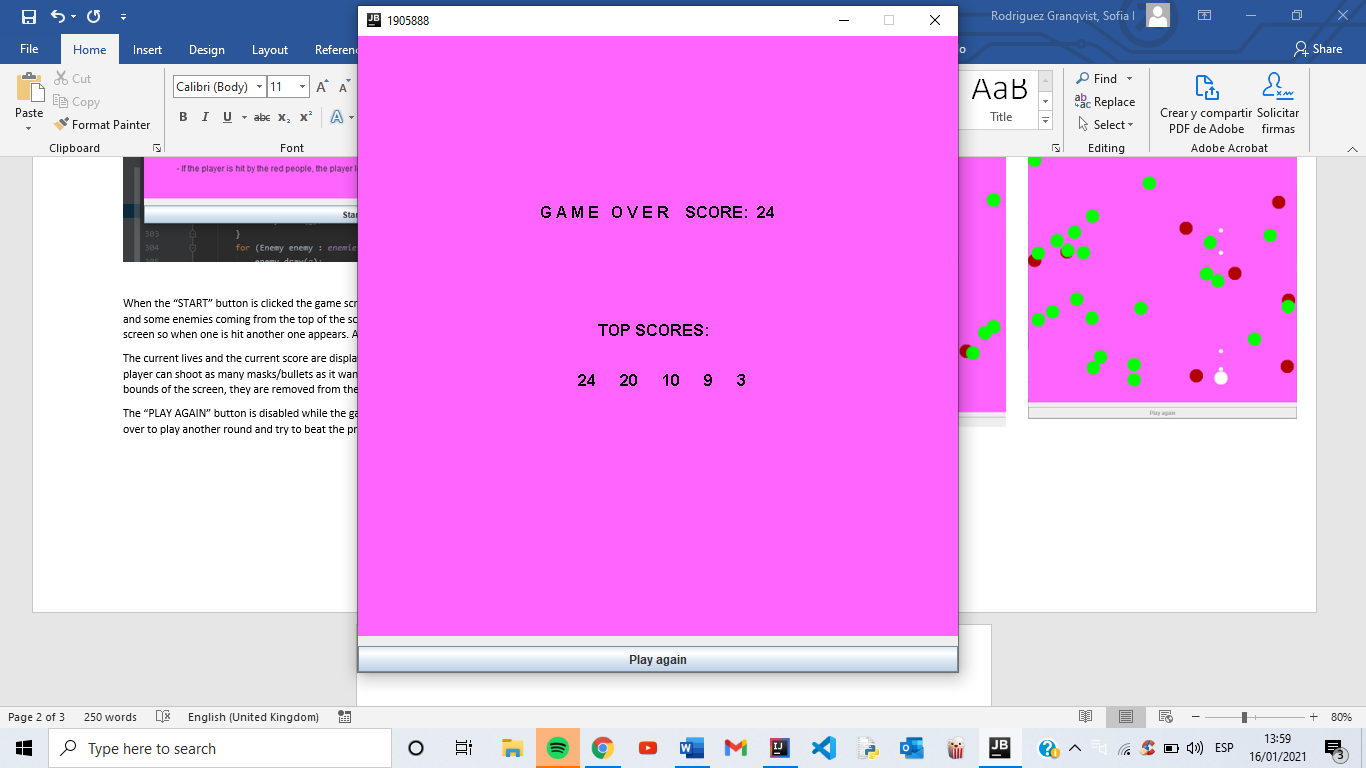
The current lives and the current score are displayed on the top left of the screen and the player can shoot as many masks/bullets as it wants. Once the masks/bullets are out of the bounds of the screen, they are removed from the array they are stored in.

The “PLAY AGAIN” button is disabled while the game is running and allowed when the game is over to play another round and try to beat the previous score.

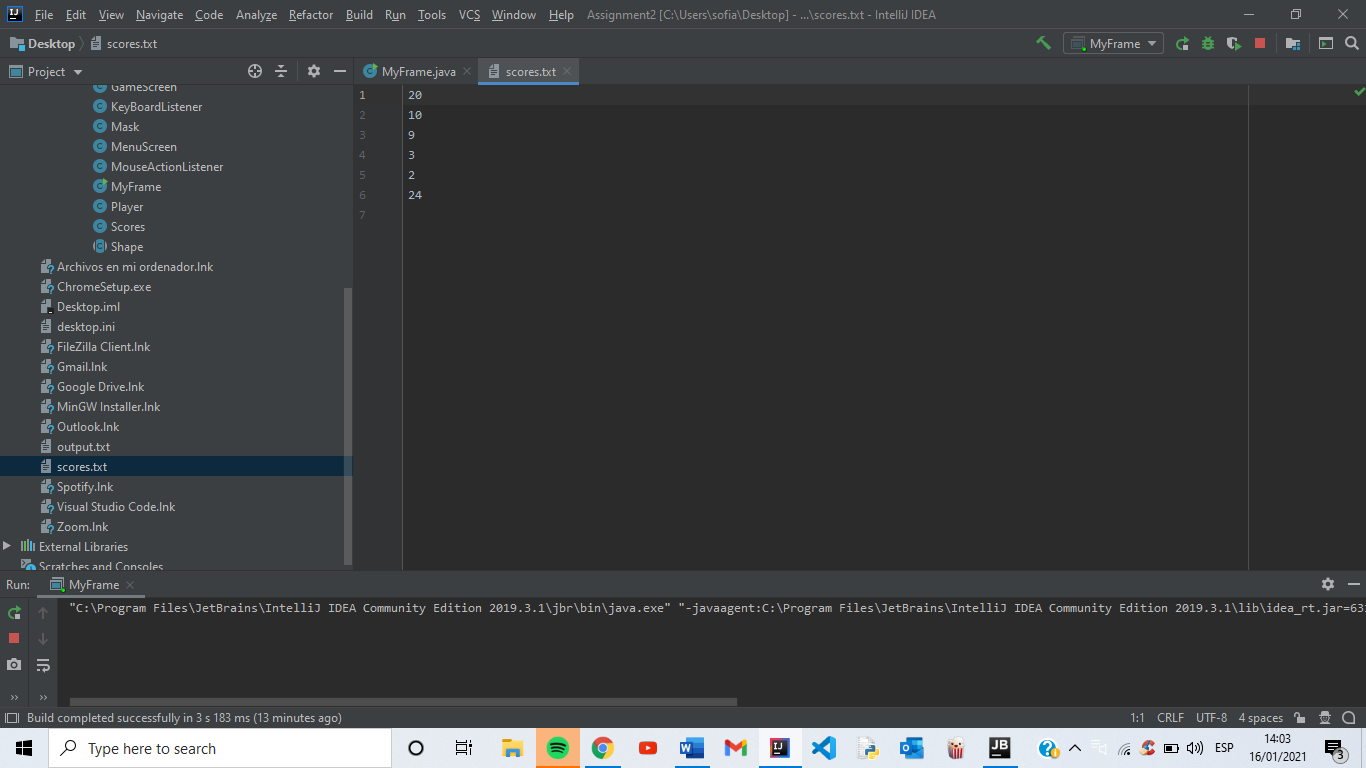




When the player is dead, a game over screen is shown with the score obtained in the current round, and the top 5 scores are shown on screen is descending order. The “PLAY AGAIN” button is enabled and another round starts:



Every score is written into a text file called “scores.txt”



POLYMORPHISM

All objects that interact in the game (Player, Enemy and Mask) extend the abstract class “Shape” based on the GRASP pattern “Polymorphism”. They all have in common that they have a posX, a posY, a radius and a color. They also override the method “draw” from the class Shape for the object to be drawn on the screen.

|  |
| --- |
| *Shape* |
| +posX : int  +posY : int  +radius:int  +color:Color |
| +Shape(int,int,Color,int)  +draw(Graphics2D) : void |

|  |
| --- |
| Player |
| -speed:int  -dirX:int  -dirY:int  +lives:int  +left:boolean  +right:boolean  +up:boolean  +down:boolean  +firing:Boolean |
| +Player(int,int,Color,int)  +setLeft(boolean):void  +setRight(boolean):void  +setUp(boolean):void  +setDown(boolean):void  +PlayerIsHit():void  +getPosX():int  +getPosY():int  +getRadius():int  +update():void <<override>>  +draw(Graphics2D):void<<override>> |

|  |
| --- |
| Mask |
| -speed:int  -distX:int  -distY:int  +left:boolean  +right:boolean  +up:boolean  +down:boolean |
| +Mask(int,int,Color,int)  +setLeft(boolean):void  +setRight(boolean):void  +setUp(boolean):void  +setDown(boolean):void  +getPosX():int  +getPosY():int  +getRadius():int  +update():void <<override>>  +draw(Graphics2D):void<<override>> |

|  |
| --- |
| Enemy |
| -speed:double  -dx:double  -dy:double  +rad:double  +onScreen:boolean  +hit:boolean |
| +Enemy(int,int,Color,int)  +isHit():void  +createEnemy(ArrayList<Enemy>):void  +getPosX():double  +getPosY():double  +getRadius():int  +update():void <<override>>  +draw(Graphics2D):void<<override>> |

STORE SCORES IN DATABASE

1. Make the necessary imports to the Scores class.
2. Create another method inside the Scores class.
3. Load a JDBC driver for the database system.
4. Establish a connection.
5. Create a Statement object which will be used to execute SQL statements.

**import java.sql.\*;**

//Create another method inside my Scores class called “InsertToDatabase”

**METHOD insertToDatabase(newScore){**

**Connection con = null;**

**Statement st = null;**

**TRY**

**Class.forName( "com.mysql.jdbc.Driver“ );**

**connection = DriverManager.getConnection(<<path>>);**

**st = con.createStatement();**

**m = st.executeQuery(<<sql query>>);**

**IF m is equal to 1**

**THEN**

**PRINT “successfull”**

**ELSE**

**PRINT “unsuccessfull”**

**CLOSE statement**

**CLOSE connection**

**}**

**CATCH (ClassNotFoundException e) {**

**PRINT some error line**

**}**

**CATCH (SQLException e) {**

**PRINT some error line**

**}**

**}**