DORMERS WELLS HIGH SCHOOL

ENRICHMENT WEEK

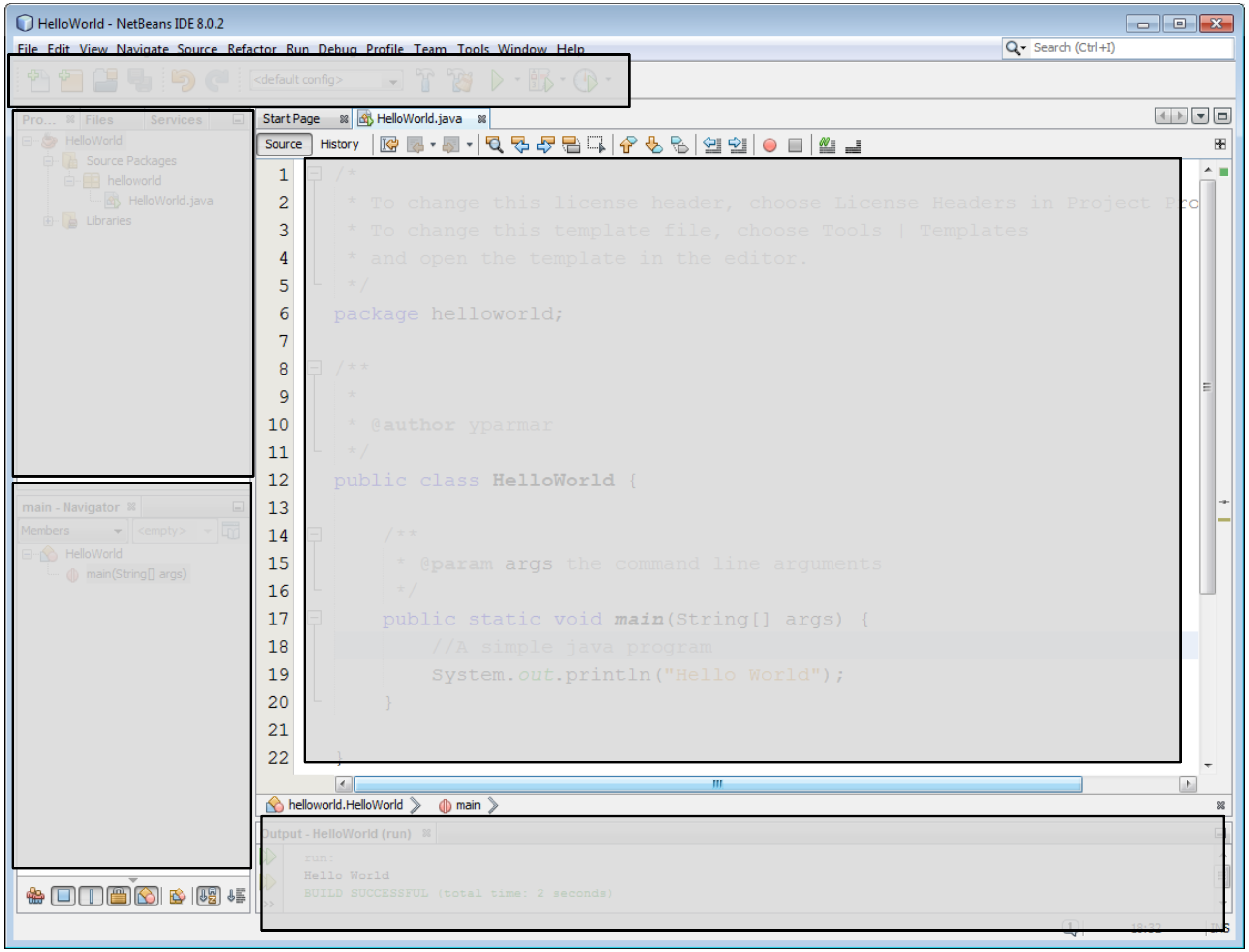
Game Design Concepts I

Name ....

Red

Introduction to Netbeans

Label the boxed of areas of the screenshot below, stating what the different parts of the interface are for.



```
/*
 * To change this license header, choose License Headers in Project Prc
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package helloworld;

/**
 *
 * @author yparmar
 */
public class HelloWorld {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        //A simple java program
        System.out.println("Hello World");
    }

}
```

```
run:
Hello World
BUILD SUCCESSFUL (total time: 2 seconds)
```

Lost something?
Click on
Window to get it
back.

DAY 1

A JavaFX window

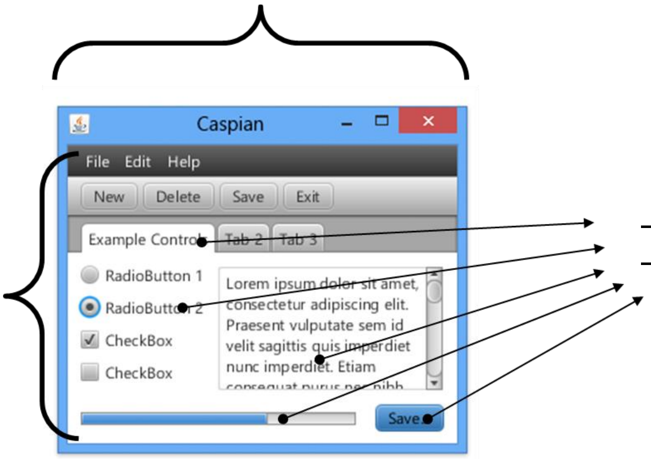
Label the diagram below with the following words:

Scene Window JavaFX Components Layout Stage

The _____________ is the _________.

_____________ contains a

____



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent vulputate sem id velit sagittis quis imperdiet nunc imperdiet. Etiam consequat purus nec nibh

____

____

Complete code for my basic window

```
package basicwindow;

import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.paint.Color;
import javafx.stage.Stage;

/**
 *
 * @author Yatish Parmar
 */

public class BasicWindow extends Application{


    public static void main(String[] args) {
        launch(args);
    }

    @Override
    public void start(Stage primaryStage){
        Group root  = new Group();
        Scene scene = new Scene(root, 640, 480, Color.BLACK);

        primaryStage.setTitle("A basic window");
        primaryStage.setScene(scene);
        primaryStage.show();


    }
}
```

Your tasks:

a) Change the size of the window to 800x600.
b) Change the background colour to one of your choice.
c) Give the window a different title.
d) Comment each LOC explaining what the purpose is.
e) Find out how to add a label to the group. Add a text title on the window itself.

Stage 1 – Making the Game Window

1. Create a new project called **Breakout**.

My window size: width = __________ *height =* ______________

Window background colour: ___________________

Window title: ________________________

2. Add shapes for a **player** and a **ball**

```
Class objectName;
objectName = new Class();


Rectangle ___________;
_______________ = new Rectangle( __ , __, _____________________)


Circle ___________;
_______________ = new circle( __ , _____________________)
```

3. Change the positions of the **player** and the **ball.**

```
playerX = ____;
_____________ .setX(playerX);



ballX = _____;
__________ .setCenterX(ballX);
```

Extension:
Set rounded corners on a rectangle by using the `.setArcWidth(value)` property.

Stage 2 – Moving the Player

What is an event listener?	What events can take place on a computer?

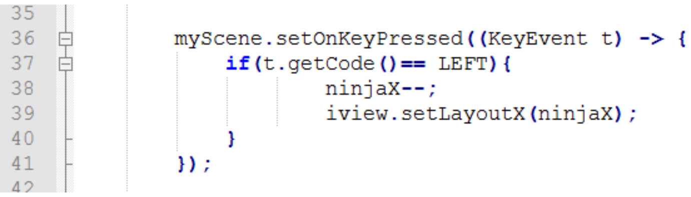
Annotate the code below explaining what is taking place

36:

37:

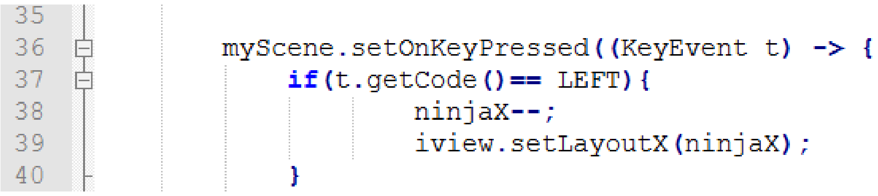
38:

39:

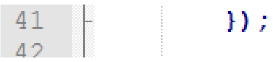


```
35
36        myScene.setOnKeyPressed((KeyEvent t) -> {
37            if(t.getCode()== LEFT){
38                ninjaX--;
39                iview.setLayoutX(ninjaX);
40            }
41        });
42
```

Modify the code below so the sprite moves right:



```
35
36        myScene.setOnKeyPressed((KeyEvent t) -> {
37            if(t.getCode()== LEFT){
38                ninjaX--;
39                iview.setLayoutX(ninjaX);
40            }
```



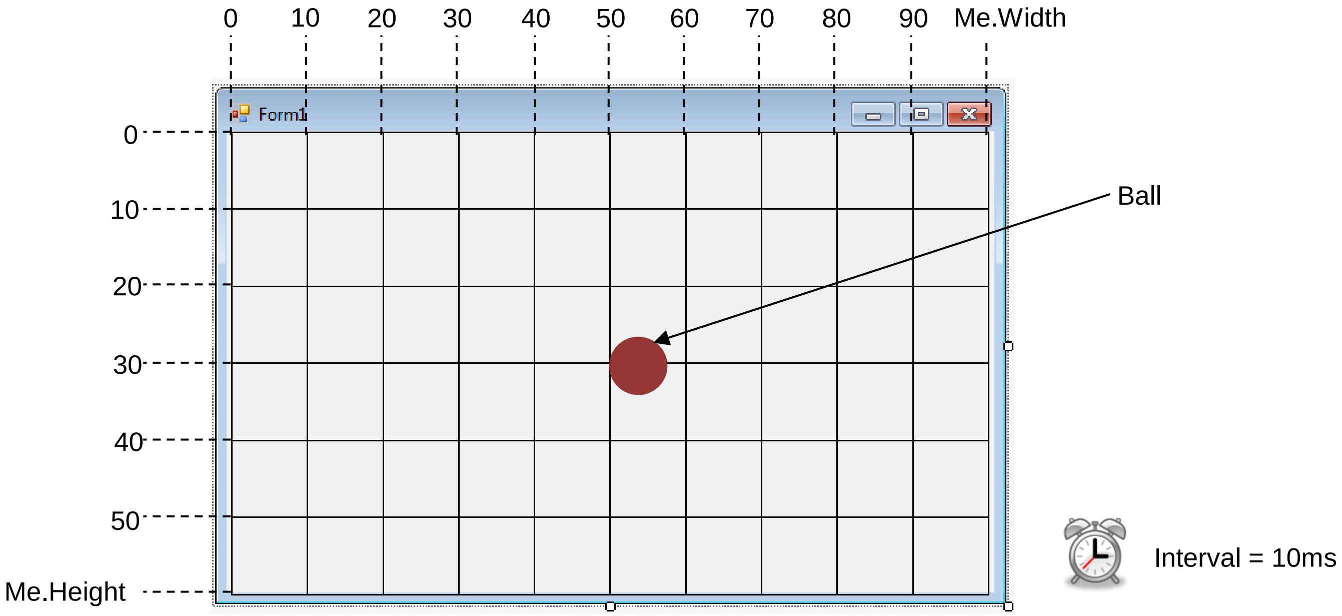
```
41        });
42
```

Explain what changes you would make to stop the player from going off the screen

Hint:

```
                                    <
                                    <=
if (t.getCode()==LEFT) && ballX     =     _______ ){
                                    >=
                                    >
```

Stage 3i - Co-ordinates and Variables



The timer ticks once every 10 milliseconds.

1) What is the *initial* value of ballX?

2) If the instruction

```
ballX = ballX – 10
```

is executed once each time the timer ticks, fill in the table to the right.

Timer Tick	Ball.Left
Initial Value:	
1	
2	
3	
4	
5	
6	
7	

3) Is there a problem with this? If so what?

4) This time I am going to use a *variable* called **xSpeed**. xSpeed has a value of 10. If the instruction :

ballX = ballX – xSpeed

is executed once each time the timer ticks, fill in the table to the right.

Timer Tick	Ball.Left
Initial Value:	
1	
2	
3	
4	
5	
6	
7	

5) This time, xSpeed has a value of -10. If the instruction:

Ball.Left = Ball.Left - xSpeed

is executed once each time the timer ticks, fill in the table to the right.

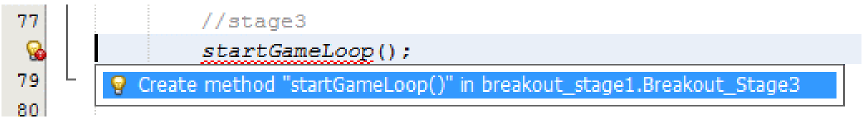
Timer Tick	Ball.Left
Initial Value:	
1	
2	
3	
4	
5	
6	
7	

6) Is there a problem with this? What would you like to happen instead? How could you fix this?

Hints: IF , > , < , What do you want to happen to xSpeed at the edge?

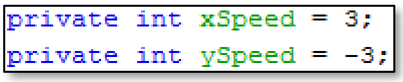
Stage 3ii – Using an AnimationTimer

A) Call a subroutine called gameLoop() at the end of start().



```
        //stage3
        startGameLoop();
```

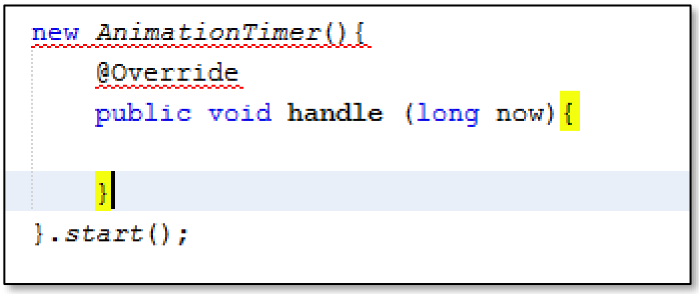
Create method "startGameLoop()" in breakout_stage1.Breakout_Stage3



```
private int xSpeed = 3;
private int ySpeed = -3;
```

B) Add two global variables for controlling the **x** and **y** speed of the ball.

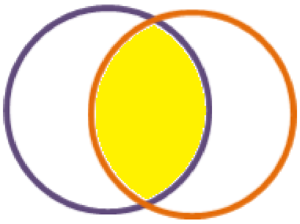
C) Add an animationTimer. This will run 60 times a second until stopped.



```
new AnimationTimer(){
    @Override
    public void handle (long now){

    }
}.start();
```

D) Look at your code from stage 2 that you wrote to move the player and your notes on stage 3. How could these be used to move the ball?

Stage 4 – Collision Detection



Collision detection involves:

- Drawing a polygon around a sprite
- Checking the co-ordinates of each point on a polygon to see if they intersect with the points on another
- Reporting back to the main program that a collision has taken place.

A) **When** does the collision detection need to take place?

❒ Every time the ball hits the bat

❒ Every timer tick to see if the ball has hit the bat?

B) Use the code below to test for a collision. Remember to replace the objects below with your own player and ball.

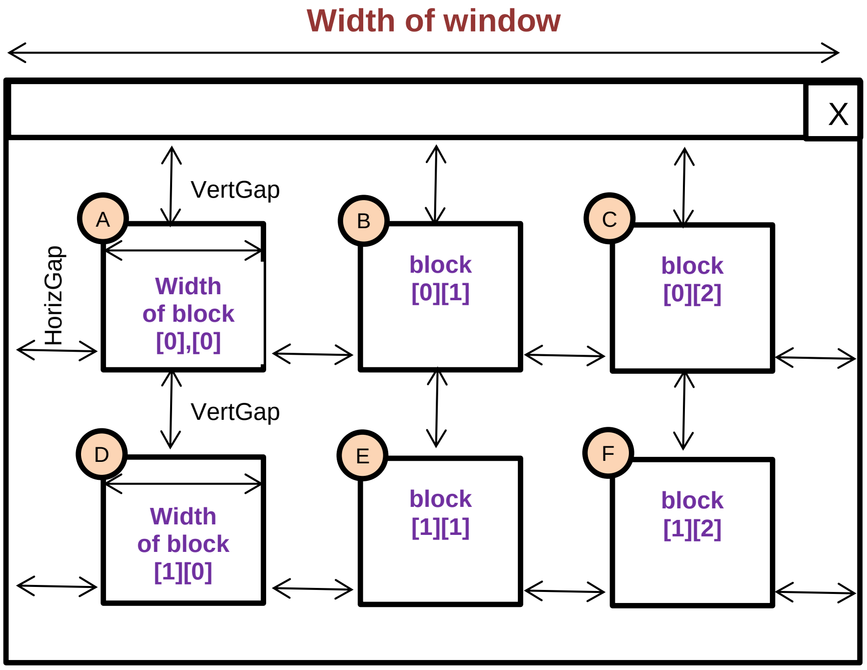
```
if(ball.getBoundsInParent().intersects(player.getBoundsInParent())){
            // What do you want to happen if the ball & player intersect?
            // e.g. bounce ball, increase points
            // will go in here
}
```

Stage 5 – Blocks

So far we have dealt with single objects (e.g. player and ball, xSpeed, ySpeed). What happens if we want to store multiple values of the same type? We need to use an array. You can imagine an array to be like a table of objects, all of the same type.

```
Rectangle[][] block = new Rectangle[2][3];
```

Look at the diagram of the window below, this shows each `Rectangle` from the array above positioned in a window. `VertGap = 15` and `HorizGap = 20`. The `width` of a block is 40 and the `height` is 30. Calculate the x and y co-ordinates for the origin (top left corner) of each block.



Block	x	y
A		
B		
C		
D		
E		
F		

Write a formula that would draw each Rectangle into the correct x,y position on screen

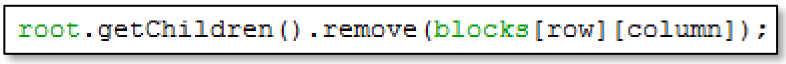
X ....

....

Y...

....

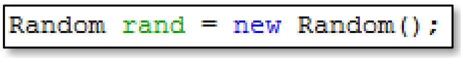
A) Create a global array of type `Rectangle` to hold the blocks.
B) Use a loop to draw all the blocks on the screen.
C) Think about your collision detection from stage 4 and your drawing loop from B. Add collision detection so that if the ball intersects with a block, it removes the block from the (root) layout.



```
root.getChildren().remove(blocks[row][column]);
```

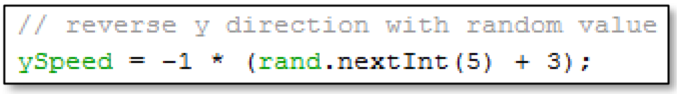
Stage 6 – The Uncharted Realm

Set the bounce angle to a random value



```
Random rand = new Random();
```

Create a global random number generator. You can then change the speed by a random value.

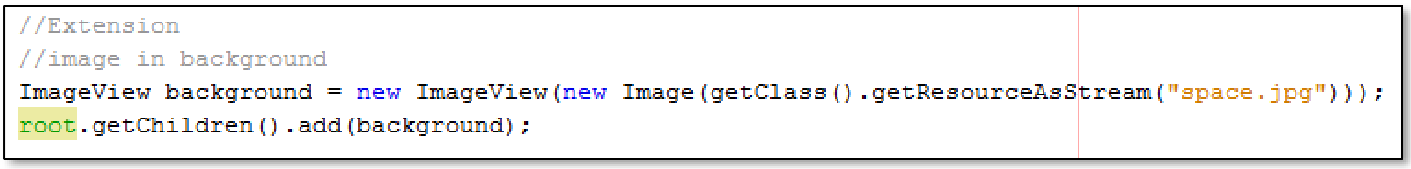


```
// reverse y direction with random value
ySpeed = -1 * (rand.nextInt(5) + 3);
```

Count the number of bounces and blocks broken

- Use a global ***variable*** to store the number of bounces/blocks broken
- When does the value need to go up?
- Add a Label to the window
- Update the label using `.setText("Score = "+ variableName)`

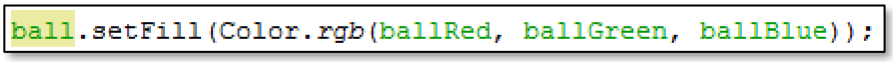
Add an image in the background



```
//Extension
//image in background
ImageView background = new ImageView(new Image(getClass().getResourceAsStream("space.jpg")));
root.getChildren().add(background);
```

Changing ball colour

The line of code below will change the ball colour to the RGB values of ballRed, ballGreen, ballBlue.

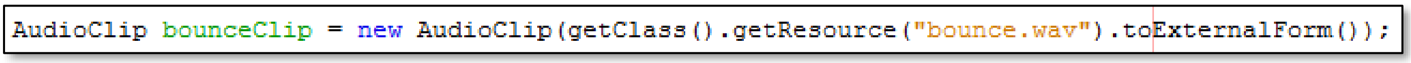


```
ball.setFill(Color.rgb(ballRed, ballGreen, ballBlue));
```

How can you cycle the values of the colours from 0-255 while the game is playing?

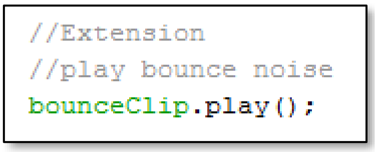
Bounce sound effect

1) Download a bounce sound effect from the web.
2) Use this code to help you create a global AudioClip



```
AudioClip bounceClip = new AudioClip(getClass().getResource("bounce.wav").toExternalForm());
```

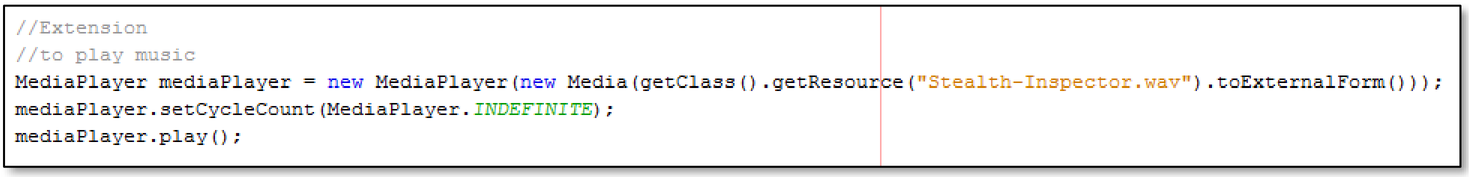
3) Play the AudioClip when the ball bounces off the player.



```
//Extension
//play bounce noise
bounceClip.play();
```

Background music

1) Download some music from audio.lgfl.org.uk and add the **.wav** file to the project resources.
2) Create a MediaPlayer and start playing music when you initialise the window.



```
//Extension
//to play music
MediaPlayer mediaPlayer = new MediaPlayer(new Media(getClass().getResource("Stealth-Inspector.wav").toExternalForm()));
mediaPlayer.setCycleCount(MediaPlayer.INDEFINITE);
mediaPlayer.play();
```

DAY 2

Stage 1 – Creating the window and importing sprites

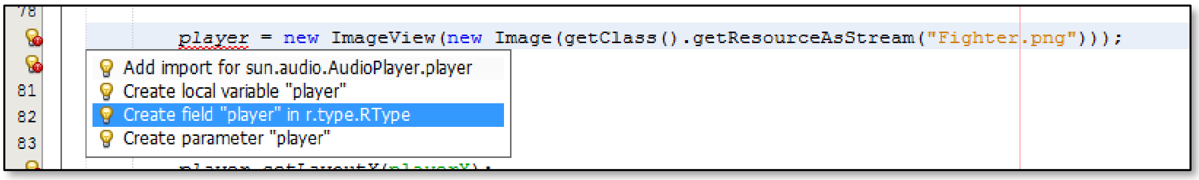
We are going to be creating a version of the classic side-scroller R-Type. The player controls a space ship which can be moved around on screen and shoots missiles. The enemies launch in waves from right to left.

A) To begin create a new project called `RType`. Create a JavaFX application + background colour (see sample code on page 5). Make the window width 1000px and the height 800px, set the background colour to black. Make sure the layout is a `Group`.

B) Download your sprites from Sharepoint and save them in your project folder. You need a:

1. Ship for your player
2. Fireball
3. Enemy

An `ImageView` is a way to look at an `Image` and add it to a JavaFX layout. This code can be used to create an image that uses the image Fighter.png.

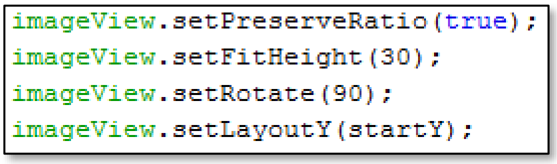


```
        player = new ImageView(new Image(getClass().getResourceAsStream("Fighter.png")));
        player.setLayoutX(playerX);
```

Add import for sun.audio.AudioPlayer.player
Create local variable "player"
Create field "player" in r.type.RType
Create parameter "player"

C) Create an `ImageView` for the player's ship. Don't forget to add it to your `Group` layout.

Here's some code to help you rotate and resize your `ImageView`. Remember, the green text below should be the name of your `ImageView`.

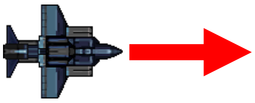
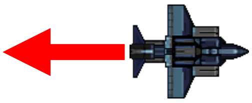


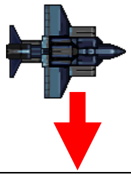
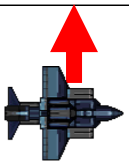
```
imageView.setPreserveRatio(true);
imageView.setFitHeight(30);
imageView.setRotate(90);
imageView.setLayoutY(startY);
```

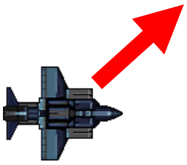
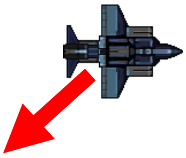
D) Add action listeners so you can move the ship in all 4 directions.

Stage 2i - Moving Diagonally

If the player is moving in a particular direction, that direction is said to be true, else it is false. For each of the indicated directions of travel, state whether the direction is true or false.

left = __________
right = _________
up = _____________
down = __________
		left = __________
		right = _________
		up = _____________
		down = __________
			

left = __________
right = _________
up = _____________
down = __________
		left = __________
		right = _________
		up = _____________
		down = __________
			

left = __________
right = _________
up = _____________
down = __________
		left = __________
		right = _________
		up = _____________
		down = __________
			

The player is moving as indicated in image 1 below. They then start moving in the direction indicated by image 2.

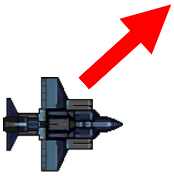


Image 1

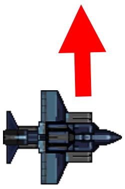


Image 2

What changed in order to change the player's direction of travel?

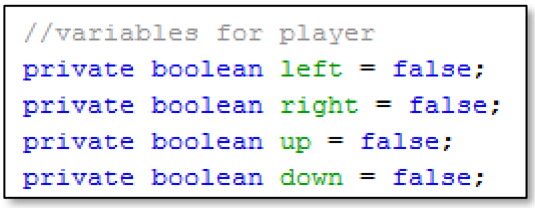
....

....

....

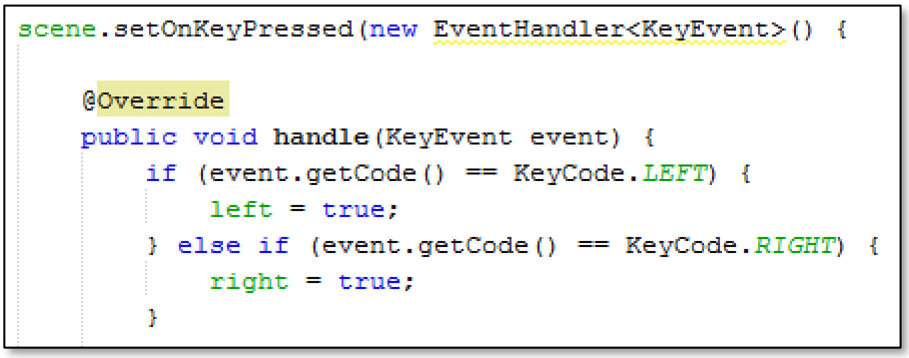
Stage 2ii – Moving diagonally using keyReleased

Here you can see that I have made global variables for each direction of travel.



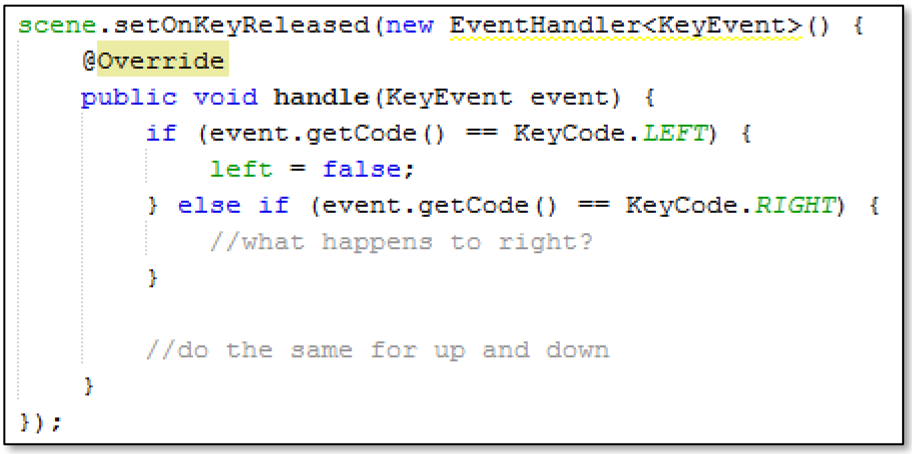
```
//variables for player
private boolean left = false;
private boolean right = false;
private boolean up = false;
private boolean down = false;
```

My old event listener updated the x co-ordinate directly. I have changed my if statement so that it now sticks the direction of travel. You need to do the same for up and down.



```
scene.setOnKeyPressed(new EventHandler<KeyEvent>() {

    @Override
    public void handle(KeyEvent event) {
        if (event.getCode() == KeyCode.LEFT) {
            left = true;
        } else if (event.getCode() == KeyCode.RIGHT) {
            right = true;
        }
```

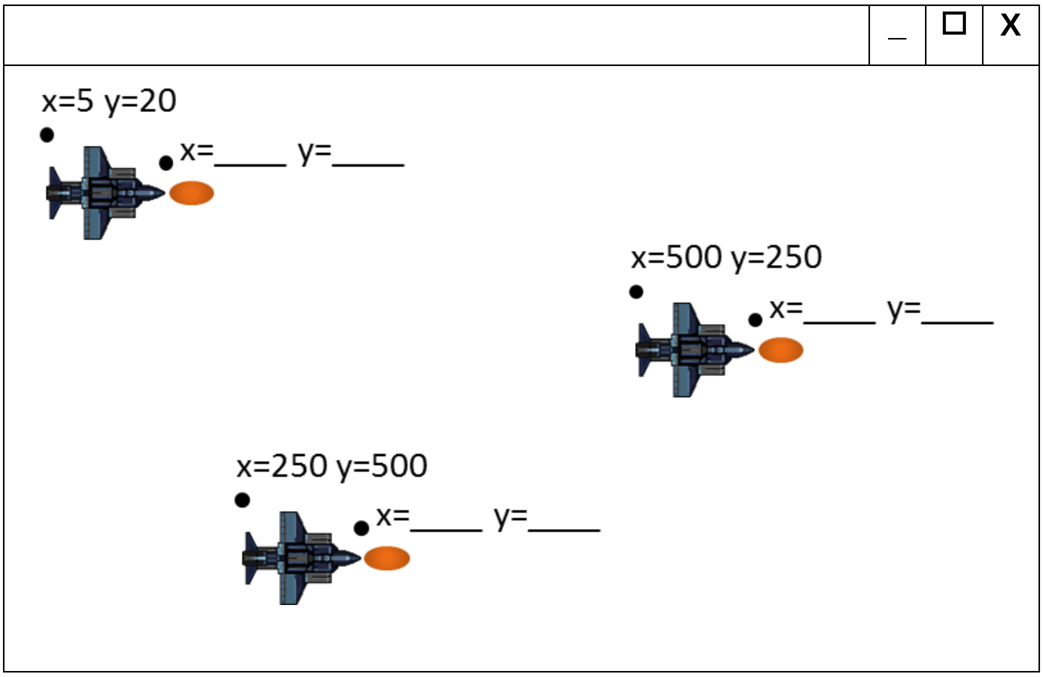
Here I have started a new event listener for when a **key is released**. This is time I unstick the direction of travel by setting it to false. I have only completed the case for left. What about right, up and down?



```
scene.setOnKeyReleased(new EventHandler<KeyEvent>() {
    @Override
    public void handle(KeyEvent event) {
        if (event.getCode() == KeyCode.LEFT) {
            left = false;
        } else if (event.getCode() == KeyCode.RIGHT) {
            //what happens to right?
        }

        //do the same for up and down
    }
});
```

Now I need to use an `AnimationTimer` to keep changing the values of my player's x and y co-ordinates. Look at your code from yesterday for moving the ball. Use this to help you create a subroutine with an `AnimationTimer` that updates the `xSpeed` and `ySpeed` of the `player`. Hint: look at page 10.

Stage 3 - Firing missiles



My spaceship is 104 pixels wide and 82 pixels high, and my fireball is 40 pixels wide and 20 pixels high.

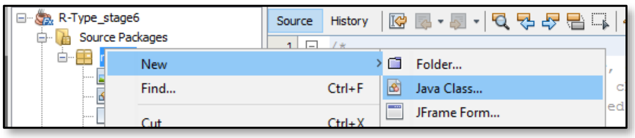
I want the fireball to appear from the nose of the ship.

For each of the spaceships on the left, work out the starting X and Y co-ordinates of the fireball.

Now use this concept to work out how you can make your fireball appear from your player's nose.

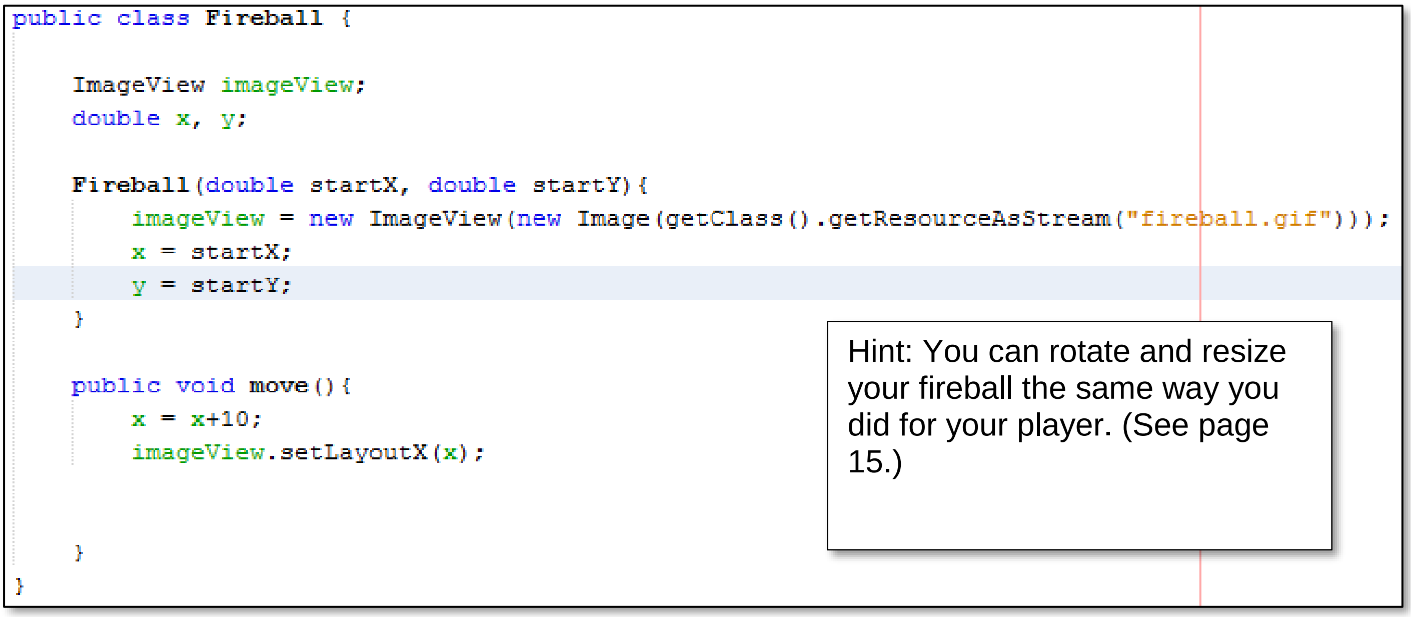
Width of my player sprite: ___________ Value I have to add to my playerX : ________

Height of my player sprite: ___________ Value I have to add to my playerY : ________



A) Create a new class for a `FireBall`.

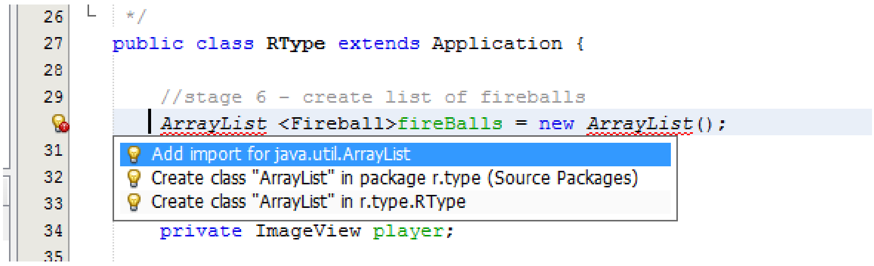
B) Use the template below to create code for a new `FireBall`.



```
public class Fireball {

    ImageView imageView;
    double x, y;

    Fireball(double startX, double startY){
        imageView = new ImageView(new Image(getClass().getResourceAsStream("fireball.gif")));
        x = startX;
        y = startY;
    }

    public void move(){
        x = x+10;
        imageView.setLayoutX(x);


    }
}
```

Hint: You can rotate and resize your fireball the same way you did for your player. (See page 15.)

C) At the top of my main class, I have created a global `ArrayList` to hold all the fireballs currently being shot. An ArrayList can hold multiple items (in this case multiple fireballs).

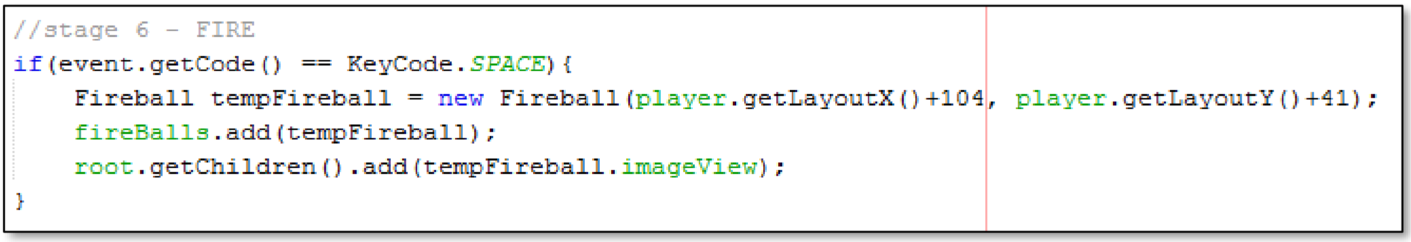


```
 */
public class RType extends Application {

    //stage 6 - create list of fireballs
    ArrayList <Fireball>fireBalls = new ArrayList();
```

Add import for java.util.ArrayList
Create class "ArrayList" in package r.type (Source Packages)
Create class "ArrayList" in r.type.RType

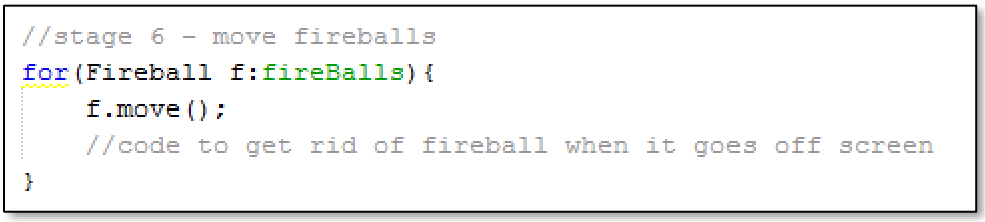
```
    private ImageView player;
```

D) Look at your event listener for key presses for controlling the player movement. Add an event listener so when the fire key is pressed three things happen.
1) A new Fireball created at the correct X and Y co-ordinates.
2) The fireball is added to the list.
3) The ImageView belonging to the fireball is added to the layout.



```
//stage 6 - FIRE
if(event.getCode() == KeyCode.SPACE){
    Fireball tempFireball = new Fireball(player.getLayoutX()+104, player.getLayoutY()+41);
    fireBalls.add(tempFireball);
    root.getChildren().add(tempFireball.imageView);
}
```

E) Add code to your animationTimer so that every fireball in the list if moved along.

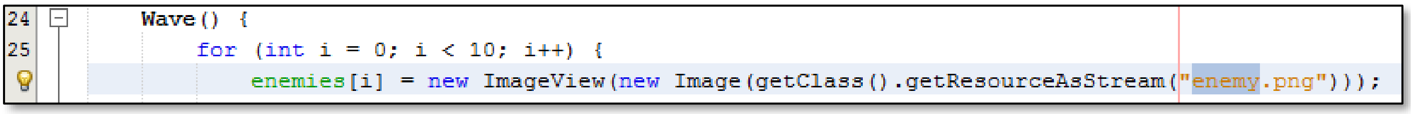


```
//stage 6 - move fireballs
for(Fireball f:fireBalls){
    f.move();
    //code to get rid of fireball when it goes off screen
}
```

E) What happens when the fireball goes off screen? Add some code to remove the fireball from the layout and the ArrayList when it is no longer needed.

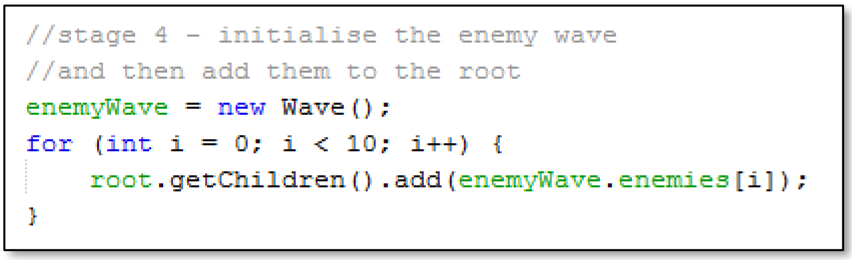
Stage 4 – Animating the enemy wave

A) Download the wave class from SharePoint and add it to your project. Change line 26 so that it has the correct name of your enemy sprite.



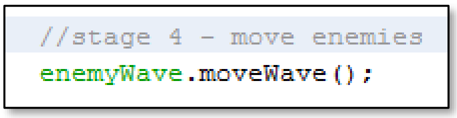
```
24      Wave() {
25          for (int i = 0; i < 10; i++) {
                enemies[i] = new ImageView(new Image(getClass().getResourceAsStream("enemy.png")));
```

B) In my RType class, where I initialised my player, I also need to initialise the enemy wave.



```
//stage 4 - initialise the enemy wave
//and then add them to the root
enemyWave = new Wave();
for (int i = 0; i < 10; i++) {
    root.getChildren().add(enemyWave.enemies[i]);
}
```

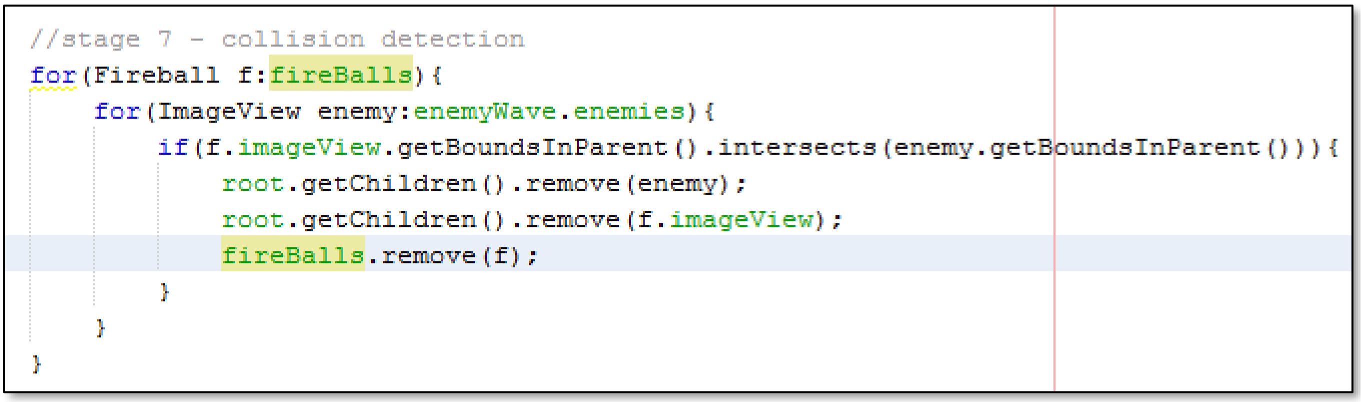
D) Finally, I need to move the enemy wave every time the timer ticks (i.e. add the code to your AnimationTimer).



```
//stage 4 - move enemies
enemyWave.moveWave();
```

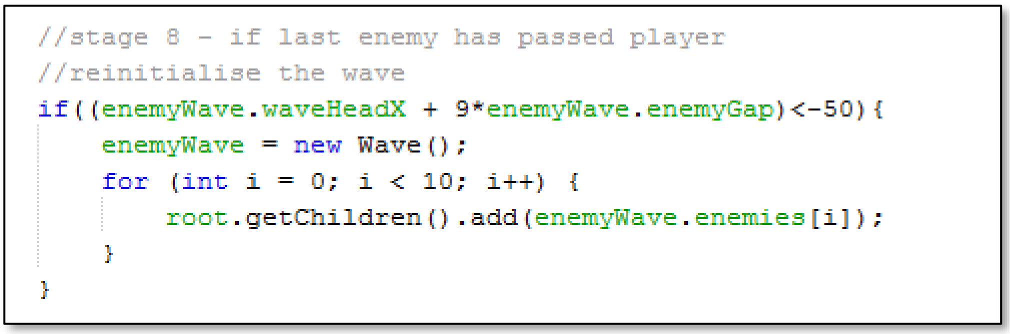
E) You can use the idea of collision detection from Block Breaker here. We checked to see if the ball intersected with each block. R-Type is a bit trickier though. We have to check if each fireball has intersected with each enemy. To do this we start with the first fireball and check it with all the enemies, then the next fireball with all the enemies, then the sent fireball and so forth. This means we need two loops:

```
Loop for each fireball
    Loop for each enemy
        Has the current fireball intersected with the current enemy?
            If yes remove the fireball and the player
```



```
//stage 7 - collision detection
for(Fireball f:fireBalls){
    for(ImageView enemy:enemyWave.enemies){
        if(f.imageView.getBoundsInParent().intersects(enemy.getBoundsInParent())){
            root.getChildren().remove(enemy);
            root.getChildren().remove(f.imageView);
            fireBalls.remove(f);
        }
    }
}
```

F) Finally, if the last enemy has passed the player, then respawn the wave.



```
//stage 8 - if last enemy has passed player
//reinitialise the wave
if((enemyWave.waveHeadX + 9*enemyWave.enemyGap)<-50){
    enemyWave = new Wave();
    for (int i = 0; i < 10; i++) {
        root.getChildren().add(enemyWave.enemies[i]);
    }
}
```

Extensions:

- Use Random to make the wave appear at different y co-ordinates.
- Modify the translation so the wave is bigger.
- Use Random to vary the height of the translation so the waves are of different sizes.

Stage 5 – The Uncharted Realm

What can you do to extend your game now? Here are some ideas:

- Enemies fire fireballs at the player
- Power ups so you can increase the number of fireballs in a single shot (e.g. double shot, then 3 fireballs at angles)
- Hold down fire for 5 seconds to launch a power shot
- End of level boss if you kill 50 enemy waves
- As you pass multiple waves, the number of simultaneous waves increases. E.g. destroy twenty waves, two waves now come at you on screen at once.
- Obstacles on screen e.g. random asteroids and comets
- Enemies randomly drop small mines which drift towards player