



# COMP-4478-WA - Game Programming Assignment I

## I. Controllers of the Game

	Player A Move Left
	Player A Move Right
	Player B Move Left
	Player B Move Right

## II. How to Run

1. Add the folder to sublime text project
2. Press Ctrl+Shift+B
3. Select Neko Test
4. Press Ctrl+Enter

## III. Program Explanation

The program represents a squash game. It is a 2-player game in which each player hit the ball turn by turn. If a player miss a hit then the opponent player gets a point. The player whose score reaches 3 first wins the game. The additional twist in the game is that the speed of the ball increases with each hit making it difficult as the game progresses.

## Step1: Defining all the Objects (Sprites) on the screen

The squash game consists of 5 different objects

1. Player A: Player A is represented by the orange paddle and can be moved using the left and right keys. Every time the left key or right key is pressed the player moves 300 pixels in left or right direction respectively.

```
if(FlxG.keys.anyPressed(["LEFT"]) && playerA.x>10){  
    playerA.velocity.x = -paddleSpeed;  
}
```

Figure 1: Player A Left Movement

```
if(FlxG.keys.anyPressed(["RIGHT"]) && playerA.x<540){  
    playerA.velocity.x = paddleSpeed;  
}
```

Figure 2: Player A Right Movement

Here the negative sign in the Figure 1 indicates the direction of motion and the variable paddleSpeed is globally initialized to 300;

2. Player B: Player B is represented by the blue paddle and can be moved using the A and D keys. Every time the A key or D key is pressed the player moves 300 pixels in left or right direction respectively.

```
if(FlxG.keys.anyPressed(["A"]) && playerB.x>10){  
    playerB.velocity.x = -paddleSpeed;  
}
```

Figure 3: Player B Left Movement

3. Score Board: The score board is of type FlxText and is placed at the center of the screen. The default text is set to "0|0", when ever a player scores the text is updated.

```
scoreBoard=new FlxText(0,0,FlxG.width,"0|0");  
scoreBoard.setFormat(null,24,FlxColor.GREEN,"center");  
add(scoreBoard);
```

Figure 4: Score Board

4. Ball: The game ball is currently a 10\*10 box of green color and having a elasticity of 1. The initial position of the ball is set just above player A paddle and at the start of game it moves towards the player A with velocity 100.

```
ball=new flixel.FlxCircle(120,400);
ball.makeGraphic(10,10,FlxColor.GREEN);
ball.elasticity=1;
ball.maxVelocity.set(10000,10000);
ball.velocity.y=100;
add(ball);
```

Figure 5: Game Ball

5. Walls: A squash game has 3 walls that the ball can hit front wall, left wall and right wall. Each of which is defined using FlxCircle and given dimensions along with additional feature like color.

```
frontwall = new FlxCircle();
frontwall.makeGraphic(640,5,FlxColor.GRAY);
frontwall.immovable=true;
add(frontwall);

leftwall = new FlxCircle();
leftwall.makeGraphic(5,480,FlxColor.GRAY);
leftwall.immovable=true;
add(leftwall);

rightwall = new FlxCircle(635,10);
rightwall.makeGraphic(5,485,FlxColor.GRAY);
rightwall.immovable = true;
add(rightwall);
```

Figure 6: Game wall

## Step2: Defining rules of the game

**Rule1:** If it is Player A turn Player B cannot hit the ball and vice versa.

This condition is achieved by making the Player B transparent so the ball as well as the Player A can pass through Player B. Here we have defined 2 flags one for each player

```
private var playerAturn=1;
private var playerBturn=0;
```

Figure 7: Player Turns

The 1 for player A indicates the next turn is of player A and 0 indicates it is not player B turn.

The below figure 8 represents that only if it is player A turn then only Player A can collide with the ball and on collision the turns are reversed. Same goes for Player B.

```

if( playerAturn==1){
    if(FlxG.collide(ball,playerA,ballVelocity)){
        playerAturn=0;
        playerBturn=1;
    }
}

```

Figure 8: Player A Collision

```

if(playerBturn==1){
    if(FlxG.collide(ball,playerB,ballVelocity)){
        playerAturn=1;
        playerBturn=0;
    }
}

```

Figure 9: Player B Collision

Here *ballvelocity* is the function to increase the velocity of the ball on each collision to increase the difficulty of the game.

```

private function ballVelocity (Ball:FlxObject, Player:FlxObject){
    if(Ball.velocity.x>0 && Ball.velocity.y>0){
        Ball.velocity.x+=200;
        Ball.velocity.y+=200;
    }
    else {
        Ball.velocity.x-=125;
        Ball.velocity.y-=125;
    }
}

```

Figure 10: Ball Velocity increment

Rule 2 &3: If player misses the hit then the opponent gets a point and also the opponent the next serve (ball).

The below code (Figure 10) shows the complete logic for Player A (Same logic applies for Player B in reverse sense).

```

if( playerAturn==1){
    if(FlxG.collide(ball,playerA,ballVelocity)){
        playerAturn=0;
        playerBturn=1;
    }
    else{
        if(ball.y>480){
            // FlxG.sound.play("assets/sounds/boo.wav");
            playerBscore++;
            playerAturn=0;
            playerBturn=1;
            scoreBoard.text=playerAScore+"|"+playerBscore;
            if(playerBscore==maxScore){
                scoreBoard.text="PLAYER B WON";
                FlxG.sound.play("assets/sounds/applause_y.wav");
                ball.velocity.set();
                new FlxTimer().start(2,function(timer){FlxG.resetGame();});
            }
            resetBall(playerB.x+playerB.width/2,playerB.y-50);
        }
    }
}
}

```

Figure 11: Player A Game Logic

Here the steps are

1. If it is player A turn then only player A can collide with the ball and the turns are reversed (Rule 1)
2. However if the player A misses the hit and ball goes beyond the screen (screen size 640\*480) then player B gets a point, the turns are reversed (as per rule 2 &3) and the score board is updated.
3. If the player B new score is greater the 3 then player B wins, a sound clip of applause plays and the game resets
4. However if the player B score is less then 3 the position of the ball is reset and the ball is given to player B for the next serve. This is done using the reset ball function which modifies ball x and y co-ordinates.