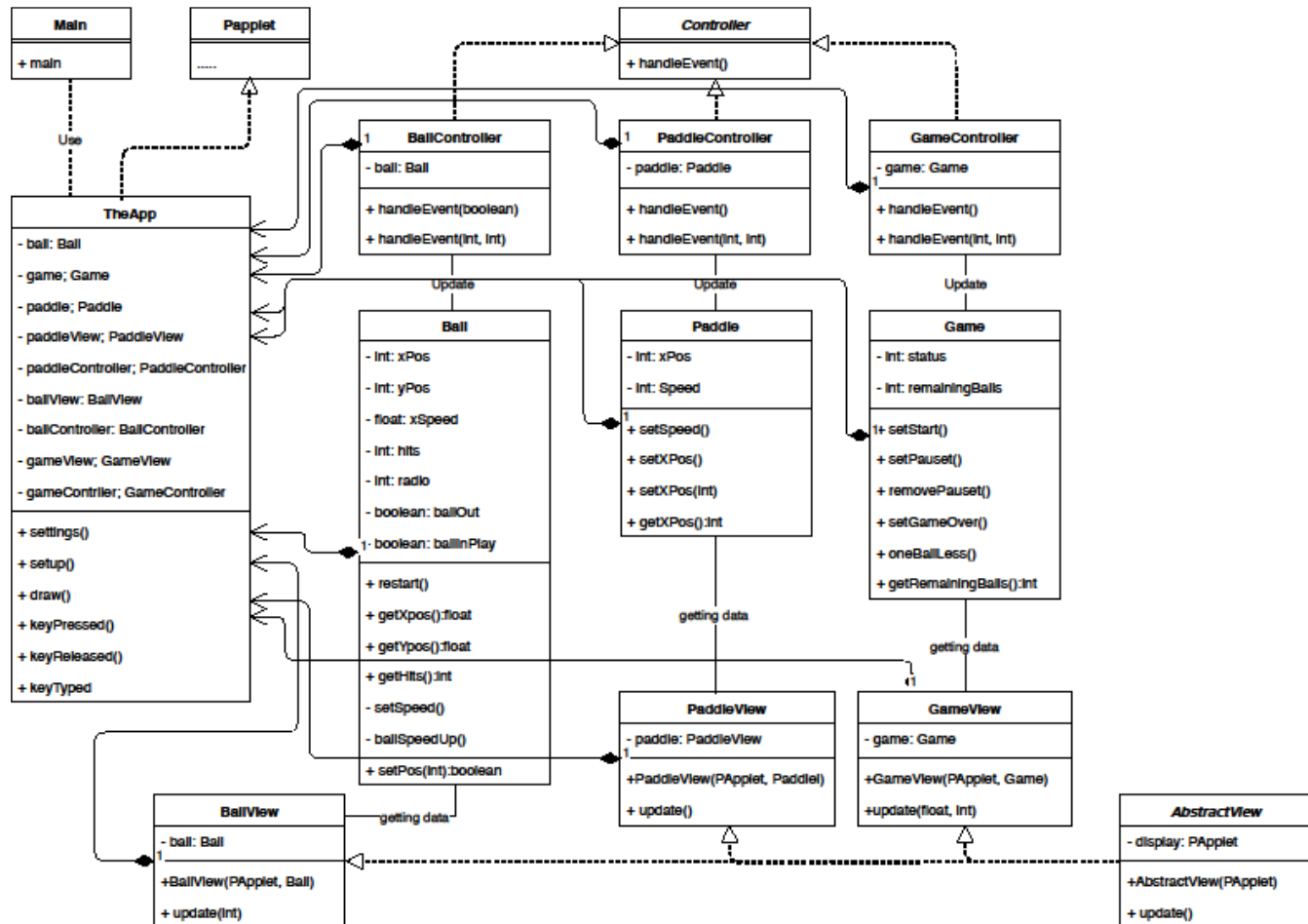


SQUASH GAME

As I'm new with Java I decided to go for the basic SQUASH game, when some of the deluxe functionalities like Number of Balls, Score and Speed up

Here the UML (also a new thing for me)

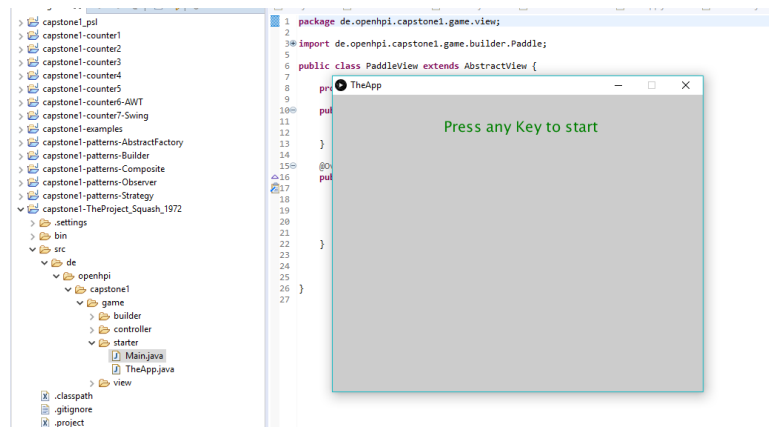


The game can have 4 different status all stored in the Game Object

```

// Game can have several status
// 0 -> Game not yet started
// 1 -> Game started
// 2 -> Game paused
// 3 -> Game over
private int gameStatus = 0;
  
```

The game starts after any key is pressed



That is controlled by the GameController

```
public void keyTyped() {  
    // Game starts when a key is pressed  
    if (game.getGameStatus() == 0) {  
        gameController.handleEvent();  
    }  
}
```

Then the ball starts to move, with random speed on X and Y axis.

```
// Ball speed is randomly set  
public void setSpeed(){  
    Random rn = new Random();  
    // Initial random speed for X axis  
    while (xSpeed == 0) {  
        xSpeed = rn.nextInt(7)-3;  
    }  
  
    // Initial random speed for Y axis  
    while (ySpeed == 0) {  
        ySpeed = rn.nextInt(7)-3;  
    }  
}
```

The paddle is controlled with the left and the right arrows.

It moves while the keys are pressed, and stops when they are released

```
public void handleEvent(int keyCode, int gameStatus) {  
    // TODO Auto-generated method stub  
    if (gameStatus == 1) {  
        // Set paddle direction towards left when LEFT arrow is pressed  
        if (keyCode == KeyEvent.VK_LEFT) {  
            paddle.setSpeed(-1);  
        }  
  
        // Set paddle direction towards right when RIGHT arrow is  
        pressed  
        } else if (keyCode == KeyEvent.VK_RIGHT) {  
            paddle.setSpeed(1);  
        }  
    }  
    // Move paddle while left or right keys are pressed  
    paddle.setXPos();  
}
```

```

    } else if (gameStatus == 2) {
        paddle.setXpos(220);
    }

```

Any time the ball hit the Paddle, the number of hits is increased by one, and the score by ten

The game shows the number of remaining balls on the right upper corner, and the current score in the left upper corner of the screen.



After 10 hits of the ball with the paddle, the game increases the speed of the ball

```

if (ballHits % 10 == 0) { // Each ten hits, speed up the ball
    ballSpeedUp();
}

```

```

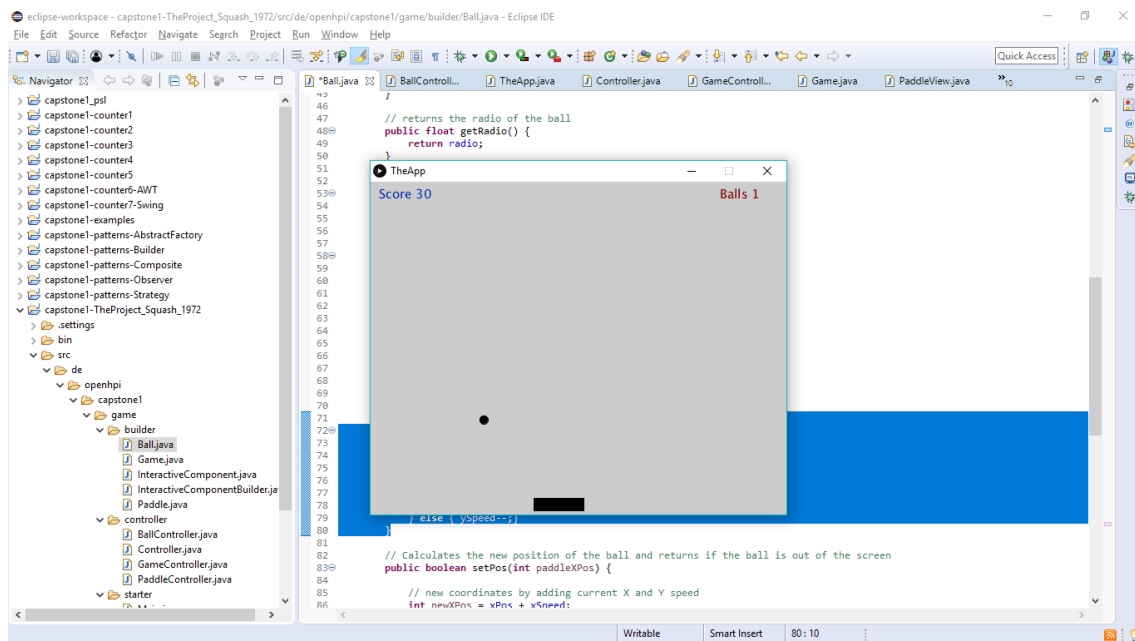
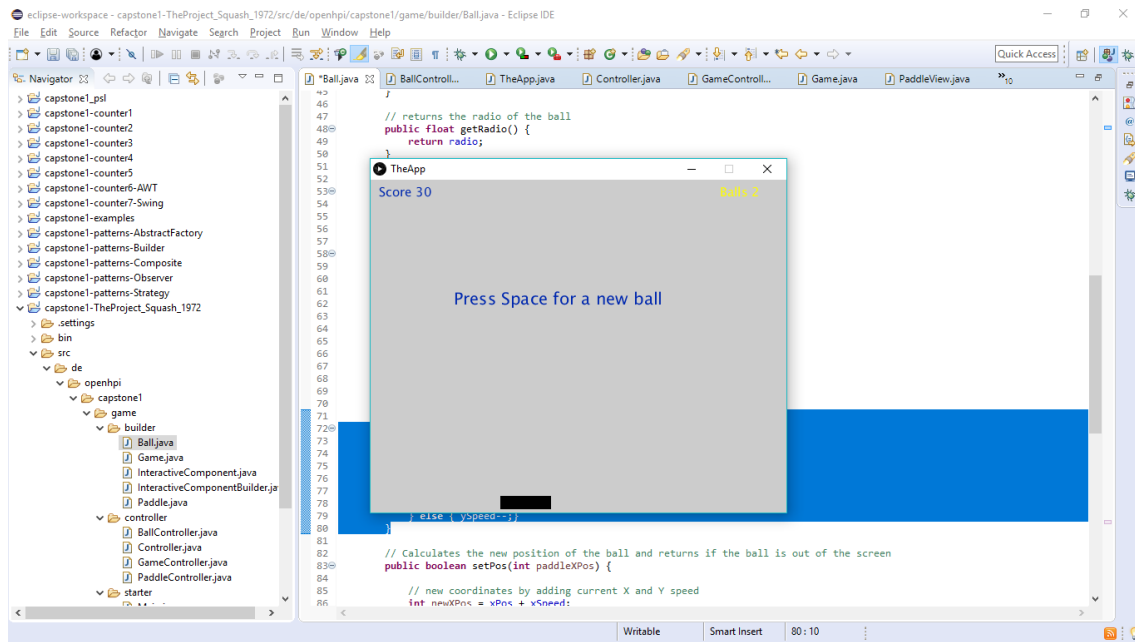
// Increases the ball speed
private void ballSpeedUp() {
    if (xSpeed > 0) {
        xSpeed++;
    } else { xSpeed--;}

    if (ySpeed > 0) {
        ySpeed++;
    } else { ySpeed--;}
}

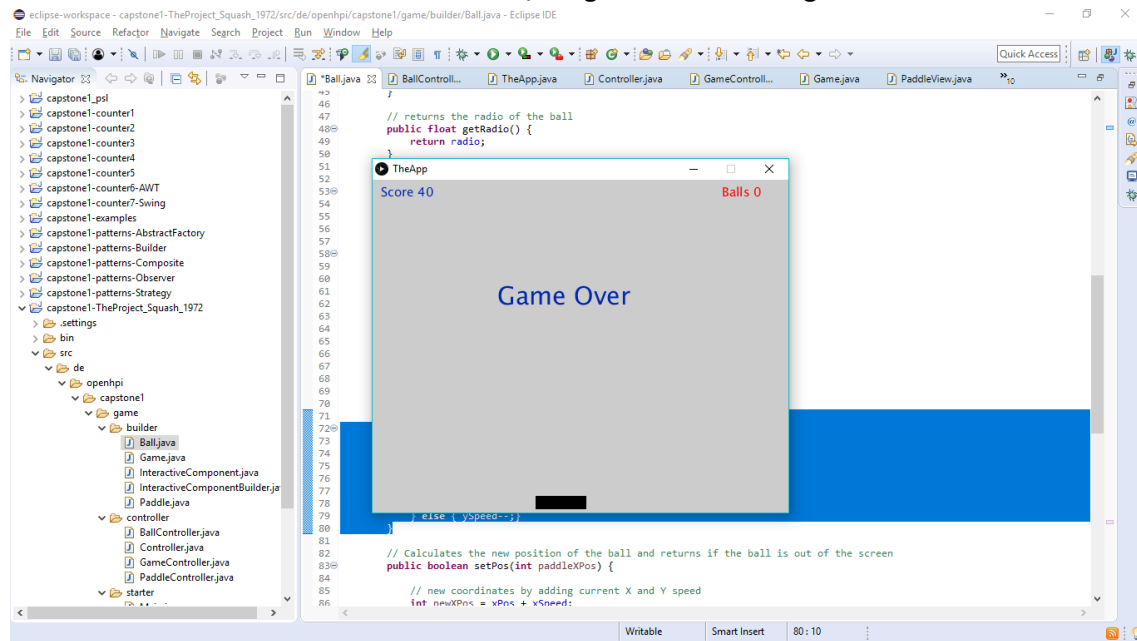
```

When the ball passes the lower limit of the window, a new ball is put in play after SPACE bar is clicked.

The number of remaining balls is displayed in the screen with a different color



In the case that no more balls left, a game over message is shown



LAB REPORT

Another new thing for me...

Well, that is the good thing, I'm learning new things, and then I realize every time I know so little, and that is great

I've seen the videos of the course, trying to understand the design patterns and how to apply them to the project.

To be honest, at the beginning it was a bit overwhelming, but then step by step, and searching also in different places I was able to understand the MVC, which I've tried to use for the implementation

I decided to work alone, because I'm completely new with Java, and was not sure how much effort I was able to put.

Finally I've put more than I was initially expecting, but this is also because I was engaged by the task

The first object I implemented was the paddle. I thought it was the easiest one, but I struggled a bit how to handle the key events.

I was reading processing documentation, and finally I made it.

I was so proud...

Then I went for the ball. As I had the experience for the paddle, it was quite fast, at least till I had to check when it was hitting the paddle.

But after some reflection, I found the way.

Next step was to have a new ball when the ball exited the screen. Seemed not difficult, but I wanted to display some texts to guide the user how to get a new ball, and again I had to face the event handling

That made me think in having also a GAME object to keep the game status and then manage the keyboard events depending on the status of the game

After that I decided to go for the score, remaining balls and speed up functionalities

In the end I think I was able to understand the MVC pattern, and to create a game according as requested.

I feel now that I have made a good progress with Java, even if I have tons of things to learn, but the experience is definitively valuable