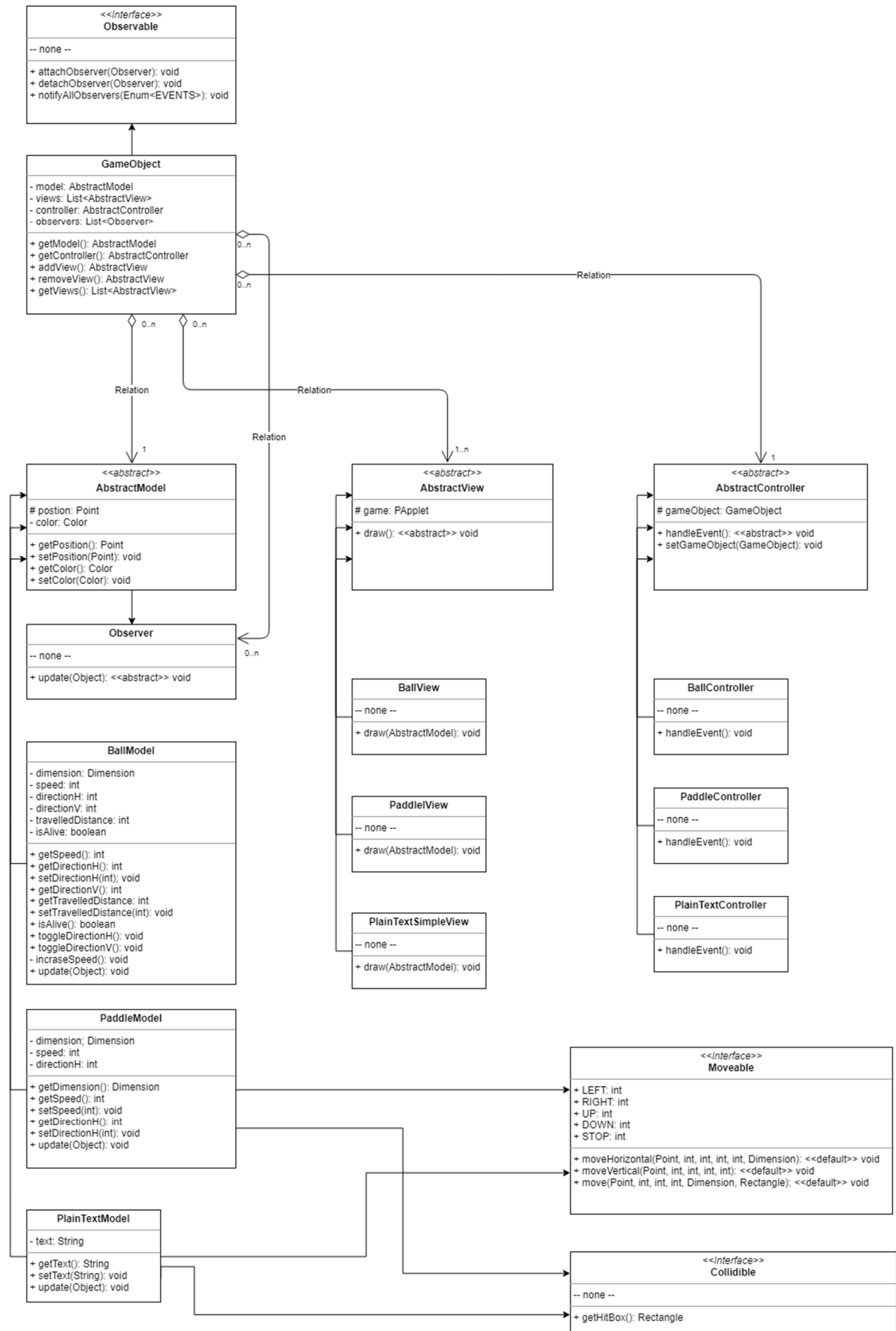


1 Class Model



2 Documentation



- 1) Shows how much balls you have left. If counter shows 0 then this is your last ball. After you lost it the game is over.
- 2) This is your score. Try to hold the ball in game as long as possible to achieve a good score.
- 3) The ball: it bounces off from left, right and upper boundaries of the playground. If it doesn't hit the paddle – which is the red rectangle at the bottom – when it hits the bottom of the playground it is lost. If a ball is lost, a new one comes into game. As long as you have balls left. Every time the ball toggles its direction it will speed up itself and the paddle.
- 4) The paddle: press a to move left, press d to move right. Try to hit the ball. If you do the ball bounces off.

3 Lab Report

3.1 Init the Git repository

3.1.1 Task

Create an account at gitlab, init the repository, check it out and push commits via commandline and eGit Plugin of Eclipse. And everything without uploading a SSH-Key ;)

3.1.2 Problems

While pushing via commandline (cmd on windows) a dialog popped up to ask for username and password. Typed in the credentials and hit the ok button, but nothing happened. No pushes at all.

3.1.3 Solution

Checking the commandline prompt and saw that the credentials have to be typed in there.

3.1.4 Happy things

eGit Plugin works fine :)

3.2 Say hello world in processing

3.2.1 Task

Start a processing application which displays a text and uses a font of my choice.

3.2.2 Problems occurred

The font could not be loaded.

```
[24] "Bahnschrift SemiBold SemiCondensed"
[25] "Bahnschrift SemiCondensed"
[26] "Bahnschrift SemiLight"
[27] "Bahnschrift SemiLight Condensed"
[28] "Bahnschrift SemiLight SemiCondensed"
[29] "Book Antiqua"
[30] "Book Antiqua Fett"
[31] "Book Antiqua Fett Kursiv"
[32] "Book Antiqua Kursiv"
[33] "Bookman Old Style"
[34] "Bookman Old Style Fett"
The font "Some Time Later.otf" is missing or inaccessible, make sure the URL is valid or that the file has been added to your sketch and is readable.
java.lang.RuntimeException: A null PFont was passed to textFont()
    at processing.core.PGraphics.textFont(PGraphics.java:4222)
    at processing.core.PApplet.textFont(PApplet.java:12886)
    at capstone.Capstone.setup(Capstone.java:25)
    at processing.core.PApplet.handleDraw(PApplet.java:2404)
    at processing.awt.PSurfaceAWT$12.callDraw(PSurfaceAWT.java:1557)
    at processing.core.PSurfaceNone$AnimationThread.run(PSurfaceNone.java:313)
```

3.2.3 Solution

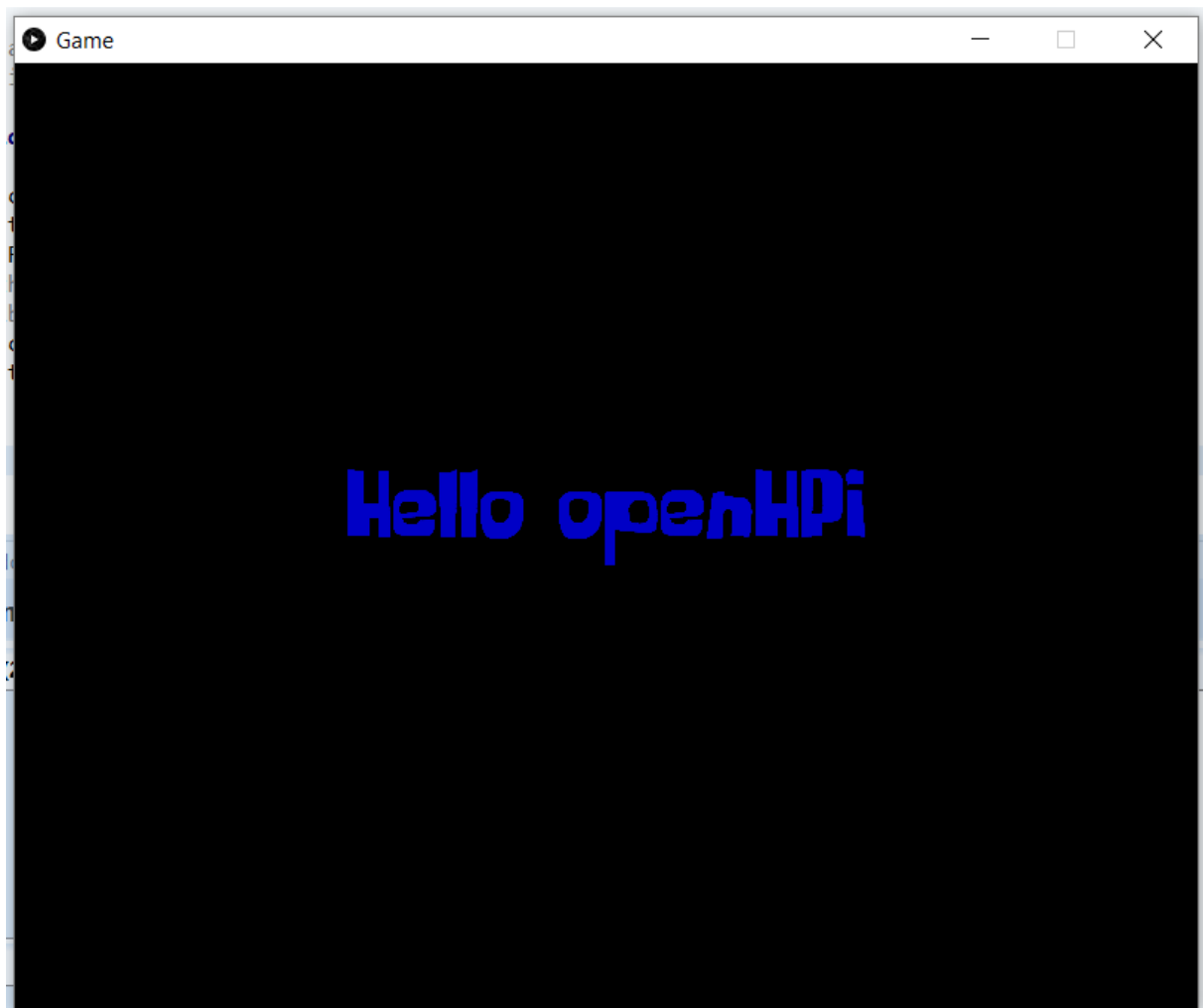
Fix the path to the font and implement a fallback if the given font could not be found.

```

/**
 * Sets the font to use to a given font and size
 *
 * @param name - the name of the font e.g "sometimelater.otf"
 * @param size - the size of the font e.g. 64
 */
private void setFont(String name, float size) {
    try {
        font = createFont(name, size);
        textFont(font);
    } catch (RuntimeException e) {
        // if the given font was not found, fetch the first of installed fonts as
        // fallback
        font = createFont(PFont.list()[0], 24);
        textFont(font);
    }
}

```

3.2.4 Result



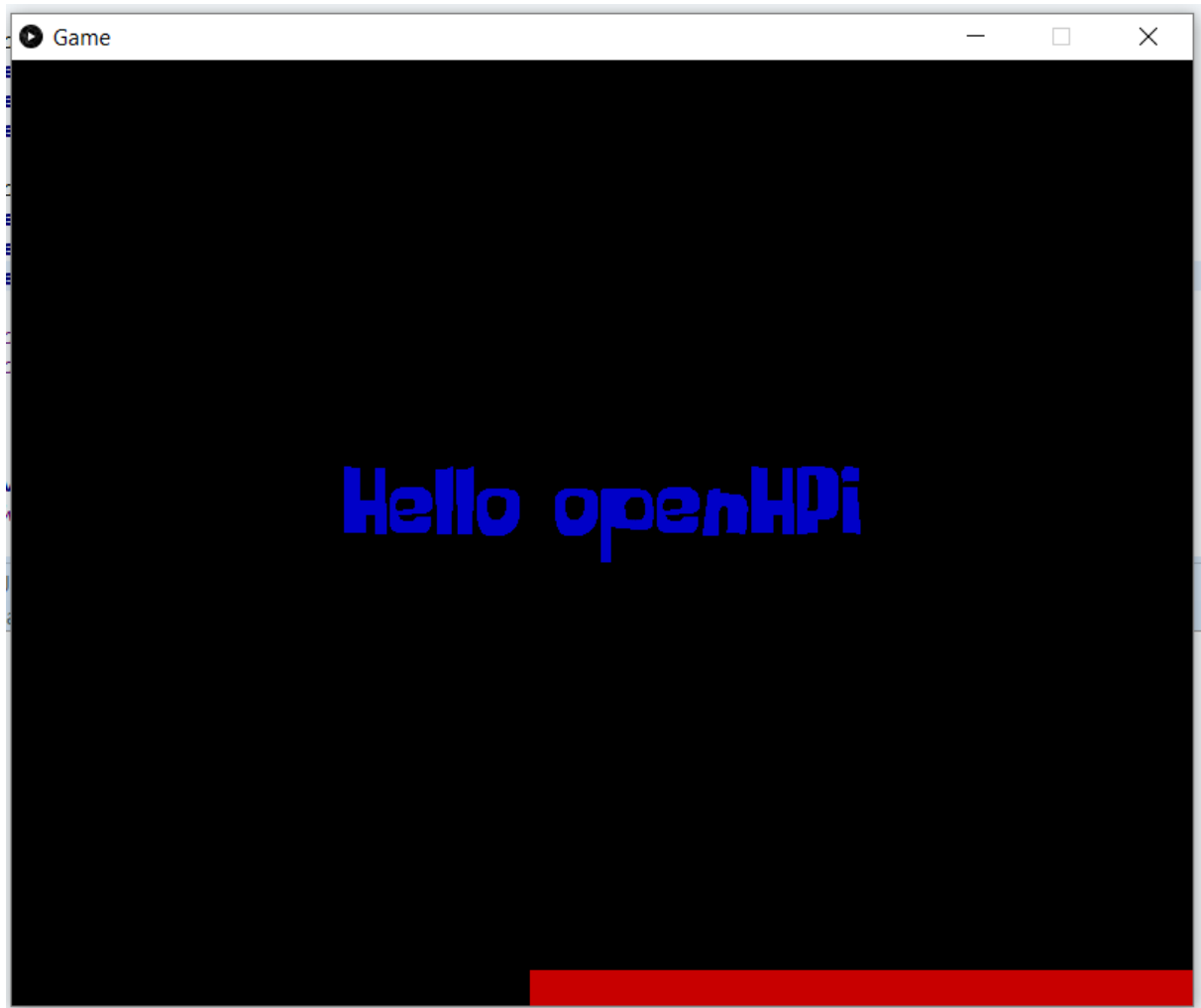
3.3 Display the paddle

3.3.1 Task

Render the paddle as a simple filled rect.

3.3.2 Problems

Paddle seems a little bit long ... Should be only 100px



3.3.3 Solution

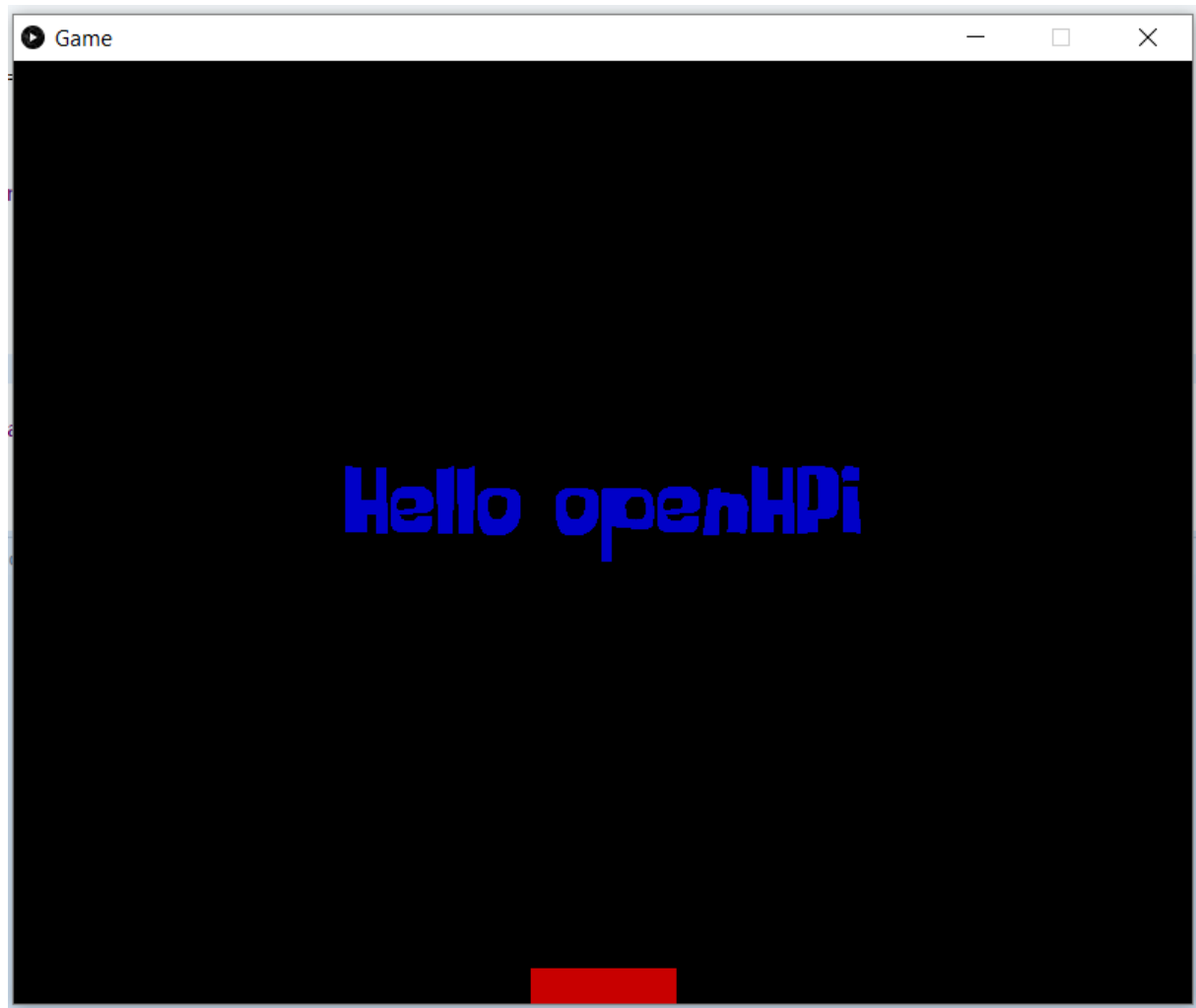
Read the documentation and change this code

```
alpha());  
y, paddle.getPosition().x + paddle.getDimension().width, paddle
```

to this

```
game.fill(c.getRed(), c.getGreen(), c.getBlue(), c.getAlpha());  
game.rect(paddle.getPosition().x, paddle.getPosition().y, paddle.getDimension().width, paddle.getDimension().height);
```

3.3.4 Result



3.4 Move the paddle

3.4.1 Task

Move the paddle via user input to left and right. Don't let the paddle leave the viewport.

3.4.2 Problems

The processing event `mouseMoved()` only catches events if the pointer is in the viewport of the processing window. Due to this it could happen that the paddle stops moving unintentionally. So this piece of code doesn't work for the purpose.

```
public void mouseMoved() {  
    // check direction of mouse and determine an event for paddle  
    Enum<EVENTS> event = lastMouseX > mouseX ? EVENTS.moveLeft : EVENTS.moveRight;  
    lastMouseX = mouseX; // remember position for next mouseMoved()  
    paddle.getController().handleEvent(event);  
}
```

3.4.3 Solution:

Move the Paddle by keyboard. The check if a key is pressed will be placed in the main loop `draw()`. And if a key was pressed the handling is delegated to a separate method.

```
public void handleKeyboardEvent(int key) {  
    switch (key) {  
        case 'a':  
            paddle.getController().handleEvent(EVENTS.moveLeft);  
            break;  
        case 'd':  
            paddle.getController().handleEvent(EVENTS.moveRight);  
            break;  
    }  
}
```