Michael Schreiber & Mario Strohmeier

Concept for the subject "Software Praktikum"

The idea:

- 2D jump & run sidescroller game
- level editor
- webapp (make it run everywhere!)

In the game you can run left and right, jump and shoot enemies.

Definition webapp (in our context):

We define webapp as a "simple" website. We use the latest technology (e.g. HTML5 techniques like canvas) to implement the game and make it able to run on every (modern) device. You can just access the game from every device with an internet browser via URL.

We try to detect the target device with javascript onboard methods like *window.navigator.userAgent* which - for example - lets us determine, if the user comes from an android operating system. We will also check if touch is available

```
var supportsTouch = "ontouchend" in document;
```

If this is the case, we can adept the controls to touch controls.

We will also determine the available resolution and adept the canvas and the controls dynamically (resize events).

This whole technique makes our webapp flexible in any way.

You can use your home PC, access the website and control the game with your keyboard or you can access it with your smart phone and control it with virtual buttons (touch).

How to:

We mainly use Javascript together with the library jQuery.

Our project will be done using the jQuery Plugin style, as described in the official jQuery documentation:

https://learn.jquery.com/plugins/basic-plugin-creation/

This makes sure, we use a continuous coding style.

For the game graphics we will use Photoshop. Most sprites will be done by the so called “image-map” system (one “big” image for one object and its animation states) which makes loading faster than having multiple, separate images for one object.

The project itself will be divided into three main parts:

1) 2D engine
2) game
3) level editor

1) This means we will - first of all - work on an abstract 2D engine which includes stuff like:

- canvas rendering logic
- animations
- AI & pathfinding
- controls
- level files parsing and interpretation

The concept of the 2D Engine will be similar to how old games (e.g. on the Super Nintendo System) worked. There are 4 layers:

- ui layer
- sprites / objects layer
- foreground layer
- background layer

There is also a hidden layer for collision detection.

The 2D Engine supplies different "APIs" where we can connect the other two parts of the project (game & level editor).

2) After we have some basic 2D Engine we will go ahead and create the Level Editor:

- create / save / load level files
- ui elements:
    - define level size, tilesets
    - edit the different layers - named above - by placing objects
- connecting to the "APIs" from the 2D engine

3) After this we will start implementing the game itself:

- ui elements:
    - buttons for touch (move left, move right, jump, shoot)
    - the main menu
- connecting to the "APIs" from the 2D engine

In theory, the game itself shouldn´t contain that much code. It´s just putting all the components together and controlling what the 2D engine supplies.

Conclusion:

Thank you for accepting our idea. We are happy that we can work on such a project.

We hope you like our detailed plans of creating the game and we are looking forward to hearing from you.