Interactive web programming

Nathan Hand

Nah14@aber.ac.uk

http://users.aber.ac.uk/nah14/iwp/coin/

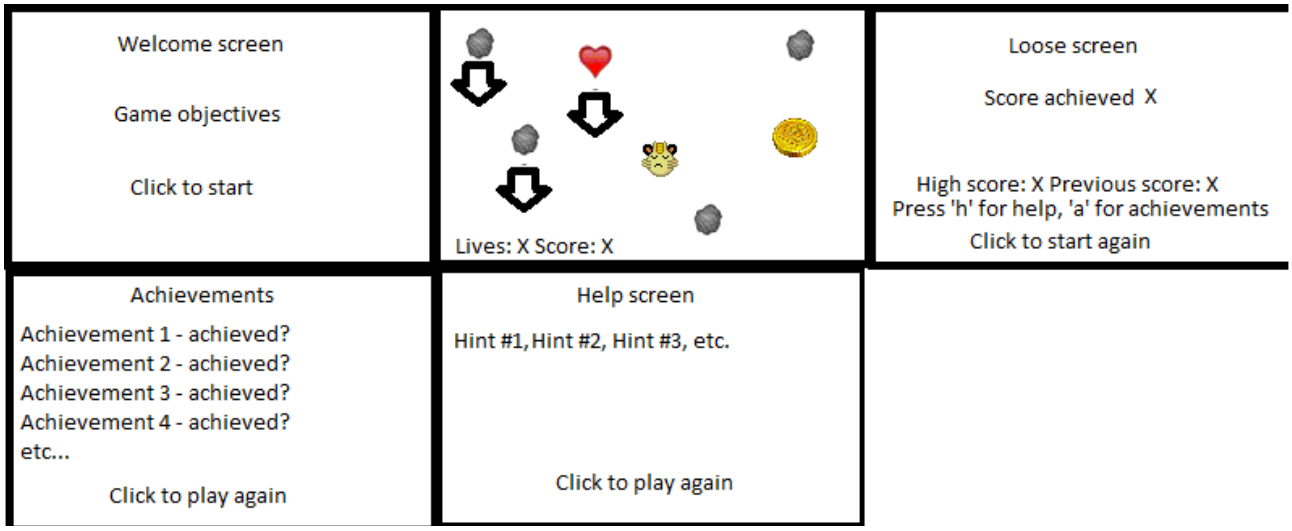
Executive Summary

A working version of the game can be found at the following address:
http://users.aber.ac.uk/nah14/iwp/coin

The game has a very simple game structure, with a very simple beginning and end states. The design was very simplistic, there was a choice of projectiles to come from the top of the screen and a choice of a small character that would be appropriate to collect coins. I chose “Meowth” for my character because that “Pokemon” is known for collecting coins/ using coins.

The game mechanics are simply to move your mouse and collect the coin, mouse move is the only mechanic in the game. You can additionally use keyboard control if you want for moving around your character however this doesn't offer a decent amount of speed compared to mouse control.

Below is a simply storyboard showing the various different screens that the user is expected to see:



As shown from the storyboard there aren't many screens in the use of my game, all of the screens are quite simplistic and easy to read.

Technical Overview

This game was entirely developed using client side technology, the reason for the use of a client side technology was that anything client side is run off the users hardware and not the hardware of the server. This therefore creates a fun game that users can play on your website while keeping all CPU usage of your hardware down all the website has to do is serve up the JS files which is incredibly simple and easy.

Alternative software that could have been used for this type of project would be things such as Flash (common place in the majority of browsers). Silverlight not very widely supported/ a well known and used platform. Those are 2 examples of alternative client side possibilities, other ideas could be to use Java servelets.

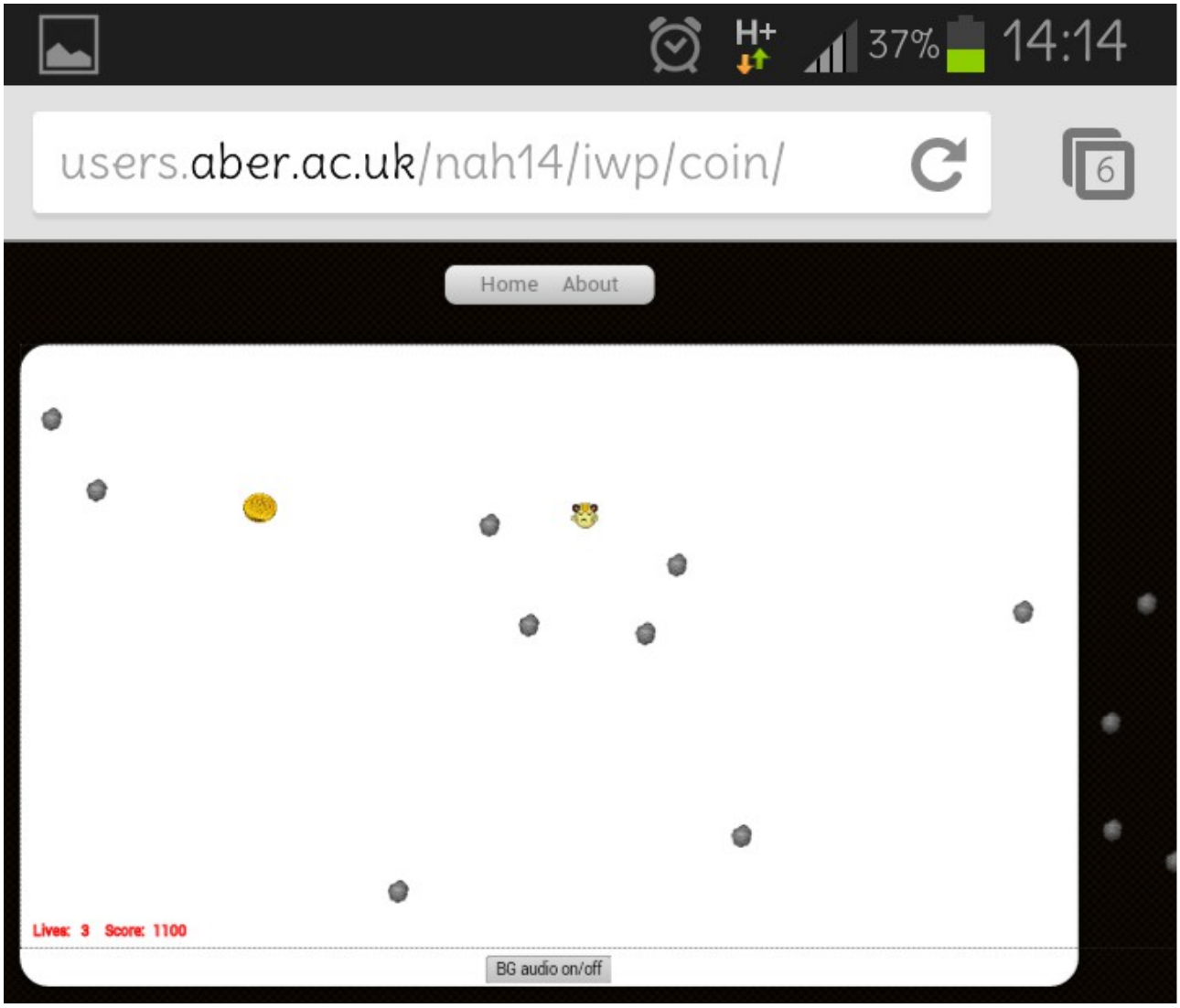
I didn't choose to incorporate server side additions, however if I were to do so I would have included session storage over local storage and stored the high score in a SQL database for a stats page etc.

Software Testing

I have tested my website on a large number of different browsers and the additional platform of android here is a testing table:

Browser	Working?	Other notes
Chrome	Yes, no issues.	N/A
Firefox	Yes, no issues.	N/A
Internet Explorer	Yes, no issues.	N/A
Chrome for mobile	Yes, no issues.	N/A

Here is an example of it working under Chrome for mobile:



The interesting use of a mobile/ touch screen device's is that because the primary method of moving around is via the mouse cursor location the actual way that you can move around the screen is the use of touching the screen which causes you to teleport around the screen rather than constant mouse movement.

So based on all of my testing my assignment actually works really quite well on all platforms tested, the only platforms not tested were additional operating systems (Linux and Mac-os) but this should technically make no difference as the browser should be the same across the different operating systems. I have missed out a few browsers in my testing but I included the primary/ majority of browsers that are available today.

Reflections and future work

This game has lots of room for improvement, and extendibility lets start with some potential improvement ideas. If I was to improve this game I would completely re-write the menu system, I didn't spend as much time as I would have liked on the menu system and as such it it's reduced to simple press 'x' to change screen or “click to start” that makes the game rather unattractive. I would have also included additional “special items” which would give the player different abilities such as invulnerability from rocks, or perhaps a “clear rocks” screen which allows for longer playability.

Generally speaking to improve extendibility as well as improvement on the game as a whole (rather then the certain doom you currently face) I would have to implement a feature to destroy rocks/ remove rocks when you attack them or collect some kind of special item.

My game essentially “goes up a level” every time you collect a coin as a new rock spawns and therefore it gets harder. However I could introduce different sizes of rocks, or projectiles at different levels to make it even harder.

I think that HTML5 and canvas is doing really well at the moment and allows for pretty much as many abilities as you would in other programming languages. I don't foresee this game being able to make money, the game is set-up as a simple time sync and fun “high scores” game. In truth all games could make money via in-app purchases however I don't think the game is interesting enough to warrant any real money transactions.

Overall I think for a game that I created from scratch without the use of any libraries I have made quite decent progress. If I were to do this assignment again I might have spent some time investigating libraries and might have made a different type of game or found ways to massively improve my game playability. The mechanics in my game are incredibly simplistic and to create game-play mechanics from scratch while possible would be made much easier using an existing library and add something to the game that might get repeat users.