CS 25210: ”A HTML5 Canvas game using animated sprites”

Due on Tuesday, April 22, 2014

James Euesden - jee22

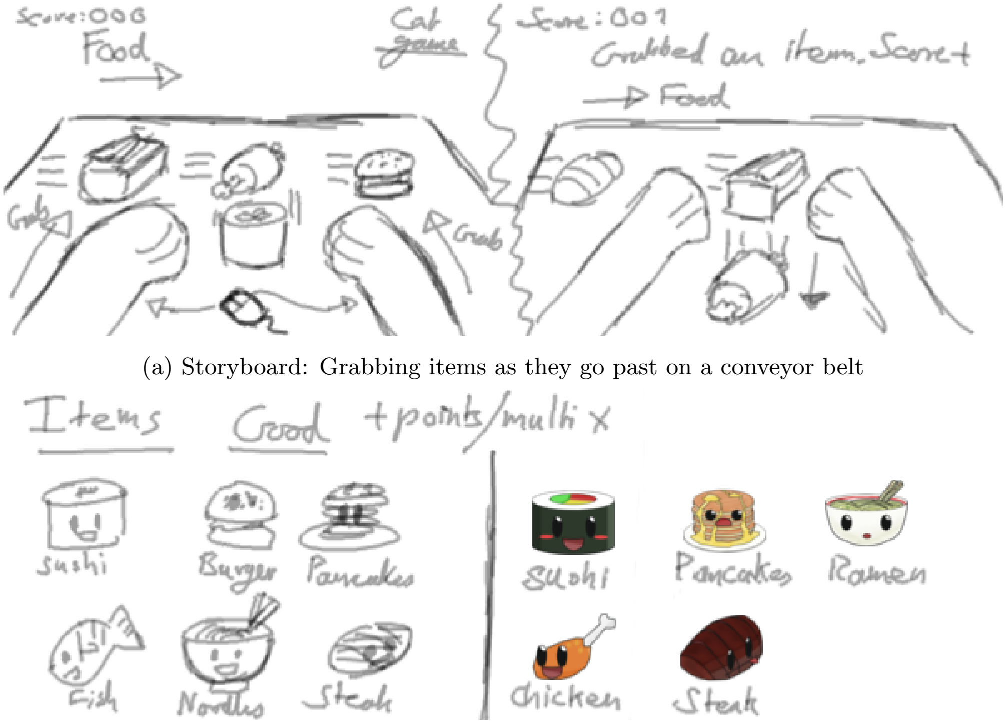
Contents

Introduction

The task set[1] was to create a HTML5 canvas game. I chose to create a ’sneaky, hungry cat’ game. This can be found at: http://users.aber.ac.uk/jee22/ftw

Executive Summary

I wanted a game that was simple to play yet challenging to conquer, that had a ’beat your friends’ aspect. My original concept was a conveyor belt, where items would pass the player that they had to grab in a set amount of time. Each item would be randomly generated and have it’s own score value. This was an okay idea but wasn’t challenging enough.

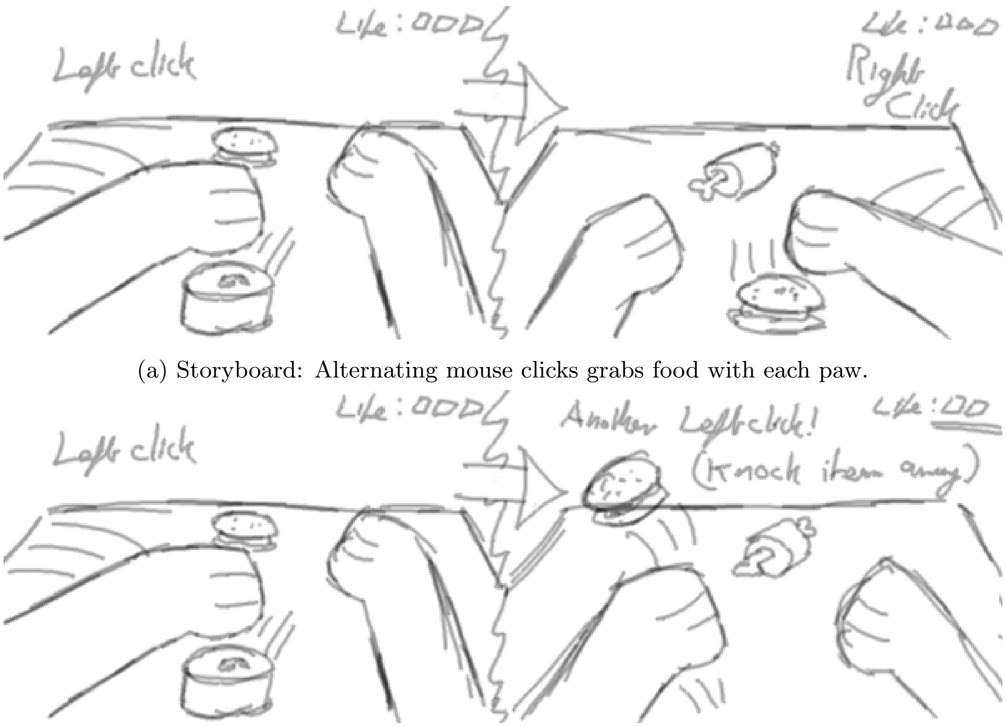


(a) Storyboard: Grabbing items as they go past on a conveyor belt

(b) The concept for the ’good’ food items and their eventual finished designs

Figure 1: Original concepts and storyboards

I developed this further to where the user could click as fast as they wanted, but had to use alternate clicks (Left to Right) to collect items. A consecutive click of a single button (Left to Left/Right to Right) would result in knocking an item away and losing a life.This presented issues with animating as fast as the clicks and didn’t really seem engaging.



(a) Storyboard: Alternating mouse clicks grabs food with each paw.

(b) Storyboard: Clicking the same button twice in a row knocks items off the table

Figure 2: Concepts: Mouse clicks and their results

My resulting idea was to combine both of my previous ideas, plus a couple of ’wildcards’: Bad items and the ’Watcher’.

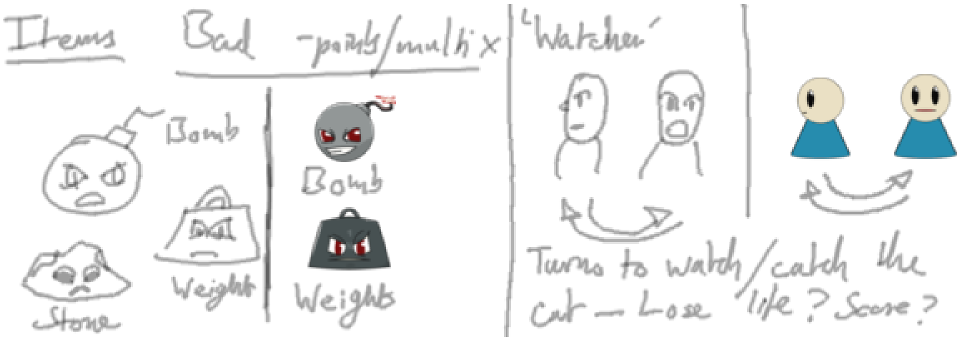


Figure 3: Bad items and the ’Watcher’.

Both bad items and the Watcher could remove some of the players score/multiplier/lives, with the bad items needing to be ’knocked’ away while the Watcher forces the player to be aware of the game as they played.

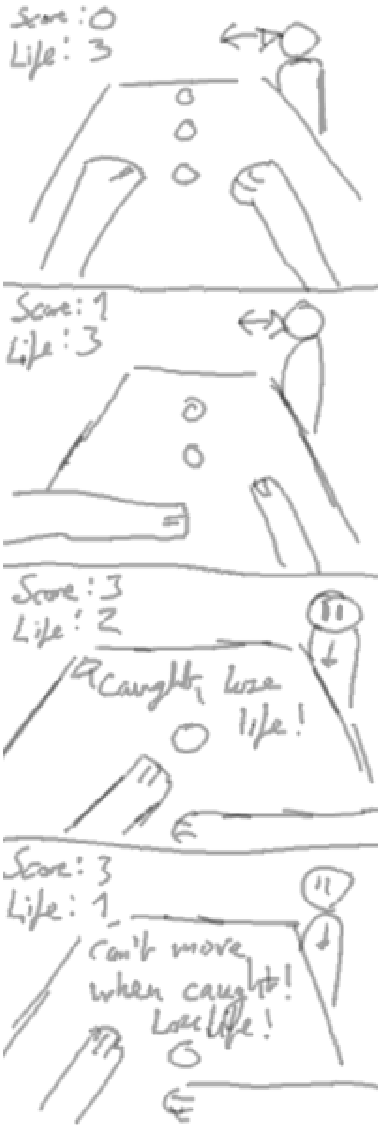


Figure 4: Storyboard: A player caught by the 'Watcher', then trying to move again.

My final game has a player use left and right click alternately to 'eat' good items for points, and use them consecutively to knock bad items away. They must also be mindful of a 'Watcher' that will randomly stare at them. If they click to move when the Watcher is looking, they lose a life.

The player starts with 5 lives. Every 15 good items eaten doubles a score multiplier, but also makes the Watcher stare more frequently. Any failure (knock a good item, eat a bad item, get caught), resets the multiplier. Eating a bad item reduces the players score and the game ends when all lives are lost.

These aspects make the game dynamic, with moments where the player must pause and wait for the watcher, or get caught outright by them when quickly clicking. I feel my resulting game fulfills my purpose of a simple game, yet tricky enough to make people want to play more and challenge their friends.

Technical Overview

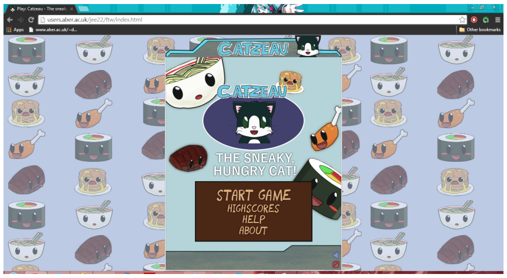
The application is written in HTML5, using the $< canvas >$ element and JavaScript. There is no server side script, so the game runs on a users machine, saving load on my webpage's server. I used the JQuery library to detect the particular mouse button being clicked and disable the appearance of the menu on a right-click.

There are other technologies I could have used (e.g. Flash). I decided to stick with JavaScript though as it is relatively well supported by most modern browsers and well documented (with help accessible for it over the Internet, i.e. W3Schools[2]).

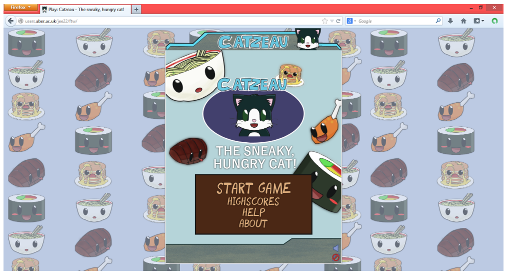
JavaScript is relatively easy to program with, coming from using languages such as C and Java previously, and can be programmed functionally and made to seem like an Object Oriented language. Being able to control the flow of the data and treat it as a language with ’objects’ made coding simple, and easy to control the elements of the game on the canvas and in the data.

Software Testing

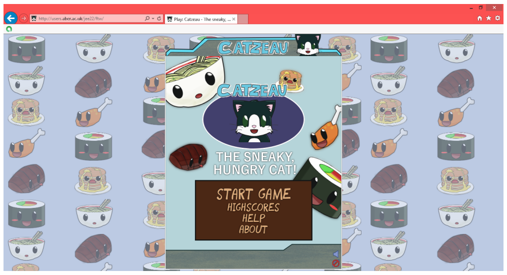
I tested my game in Google Chrome, Mozilla’s Firefox and Internet Explorer Edge browsers. The game functions similarly on each, with little to no discrepancies. As can be seen below, the game looks very similar in each browser and, when playing the game, seems to run the same too.



(a) The start menu in Chrome



(b) The start menu in Firefox



(c) The start menu in Internet Explorer Edge

The game runs on a mobile device, but since the game uses two mouse buttons on a single screen, it only responds to the left click. I could program the application to accept taps on the two sides of the screen in replacement, but I felt this was out of the scope of the assignment, and could be a future revision.

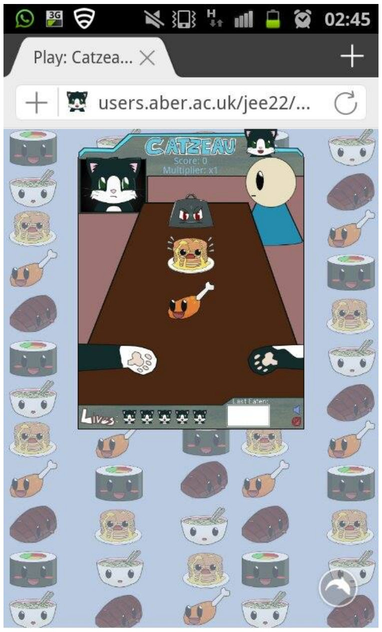


Figure 6: The gameplay using the Dolphin browser on a Samsung phone running Android 2.3

There were a few differences between browsers that needed to be accommodated for during my testing, such as IE not pausing the sound effects after being played, and some issues loading localstorage in Firefox. These tended to be minor and did not cause any current standing issues.

Reflections & Future Work

There are multiple ways I would improve the game. First, I'd implement a working mobile version. In terms of gameplay, I would add more varieties of items, and improve the graphics and AI of the 'Watcher'. I also wanted to implement a 'bonus mode', where a particular item could be consumed to 'dubstepify' the game.

Following this would be different difficulties, and modes where there is no Watcher/only good items. I even considered rewards for high scores, like different aesthetic affects or permanent toggle-able power ups. I'd also add a global high scores table, with social integration for posting scores to Facebook/Twitter, boosting the competitive side.

These are all things I'd implement were I to continue working on this project and to be considered as a marketable product, especially as it seems to fit in the vein of casual mobile gaming. Right now it is not ready to generate money. I have learned a great deal from this game, and know I will continue to create HTML5 applications in the future, utilizing what I have learned during it's creation.

References

[1] H. Dee, ”CS25210 Coursework 2014: An HTML5 Canvas game with animated sprites.”, CS25210 Assessed Assignment 2013-2014, February 2014

[2] W3schools.com (Accessed 18/04/2014) JavaScript Tutorial [Online]. Available: http://www.w3schools.com/js/