Semester Project 2 Report

Semester Project 2 Report

Introduction

Let me start off by saying this was an excellent assignment, with the perfect subject (Game of Thrones), perfect assignment (making a game) and the perfect amount of freedom to be creative. It's been some tough weeks, with some hurdles being harder to overcome than others, but I'm quite pleased with the end result and I hope you will be to.

I used more time than I initially planned to on the design part of the assignment, but the result of that is that every piece of artwork on the website is made by me. Some time was used to "re-learn" various aspects of illustrator/photoshop due to it being quite a while since I have used it, but once you learn how to ride a bike, you never truly forget it.

I really wanted to make the game multiplayer, and completing process 2, but I decided I would complete the website first, then eventually do it if I had time left in the end, so I did not risk running out of time and not having anything to deliver. In the end I did not have enough time to even look at completing process 2.

Links:

Website: http://www.sleekdesign.no/semPro2
Github: https://github.com/TomThorsen/Semester-Project-2
Prototype: https://xd.adobe.com/view/d48faebf-68c3-41ad-4771-3221dd349bea-a2be/
Gitkraken Glo Board: https://app.gitkraken.com/glo/board/WyZNs90hvxAAROdZ

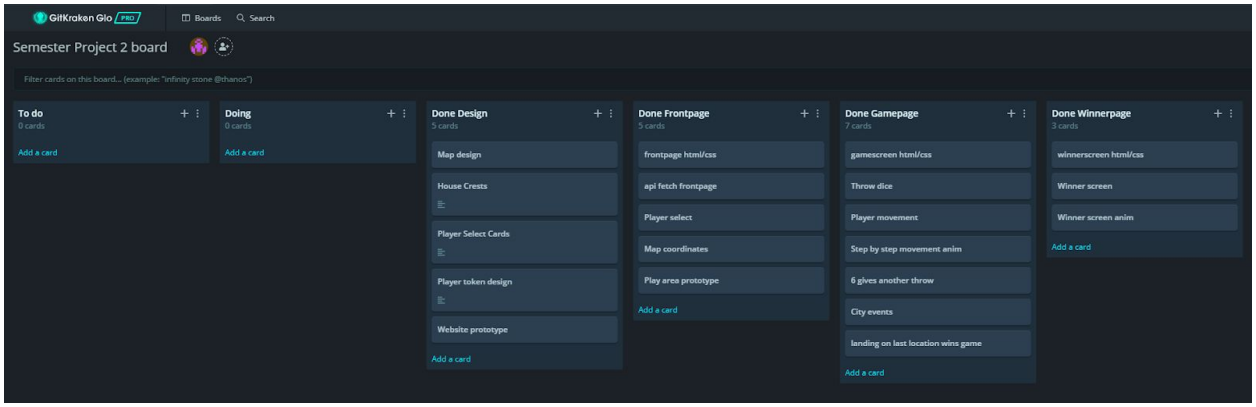
Semester Project 2 Report

Main Content

Planning Phase

My initial “mental” planning phase was longer than normal, as I had the entire summer vacation to think about the assignment before starting it. As such I did make up a pretty clear mental image of what I imagined the game/website to be.

I decided to give Gitkraken’s “Glo” board a try for planning and progress-tracking, and it was an easy and good to use alternative to trello, since I was already using Gitkraken for git commits.



(Gitkraken Glo board)

I started off by adding all the cards that I remembered being needed off the top of my head, and added more as I was working on the various stages. All the cards were in a single “Done” category originally, but I separated them now for illustration purposes.
Although not being the most in depth planning, I feel like it did the job and provided me with what I needed. I could obviously have made a gantt chart as to plan out time usage over the period, but after some of the problems I encountered, I’m pretty sure it wouldn’t have been of much help anyways, at least not in this case.

Semester Project 2 Report

Research and Design

Research

Before putting pen to paper, I spent some time just browsing through various web pages related to Game of Thrones, to get a feel for the general artwork and design related to the series.
I also went through quite a bit of images of both characters from the show, and different artwork. One thing I noticed was the lack of artwork that wasn't copyrighted, which lead me to the choice of making it all myself. I saved the most interesting pictures I found so I could use them for inspiration when creating the artwork for the web page.

Color Palette

(Website Color Palette)

After doing the research I did, I did pick up on various different color choices used, and I managed to put together a color palette of my own. The palette is centered around the purple and gold, that gives off a feeling of something royal, with the 3 other colors giving a nice contrast. These colors are used in nearly all aspects of both the artwork and the website itself, and I feel the color combinations go together nicely. It gives a nice soft feeling, without being overly hard on the contrast.

The exception is the player cards and player tokens, as I needed more colors to differentiate the different players, but the colors used seem to go nicely with the overall theme

Typeface

I spent some time looking around, seeing if I could find a suitable font to use. I already had my mind set on using google font as my source for the font, so when I came across the font Cinzel, it immediately jumped out at me as a perfect match. Normally I would use a combination of sans-serif and serif, but due to the nature and “feel” of the game/webpage, I decided against it.

Cinzel also had a more stylistic version called Cinzel Decorative that I decided would be perfect to use for the headers on the webpage, to give it even more of a Game of Thrones feel. Cinzel looks old and artistic, and could be described as something used in the medieval ages, so a perfect fit for the task at hand. The decorative version is a bit “over the top” for use in normal paragraphs, so I only wanted to use it for headers. I think it would be fairly hard to read if it was used in a smaller sized font.

I also found a more accurate Game of Thrones font to use for the main header, so it looks the part.

Map Design

(Map design steps)

I decided that my first actual “pen to paper” tasks should be designing the game map itself. I used some time looking around for different game of thrones maps to use as a template, and just edit that with my own artwork on top, but everything decent I found was copyrighted. So I went with the choice of making a new map from scratch instead.

I included a picture here where I show the different layers of the map, as to show the design process of it. I made the map in 1440p resolution, so the photoshop savefile is around 150mb, thus I was unable to include it in my git repo, thanks to my terrible upload speeds. I can obviously upload it somewhere suitable if you want the source for it.

I started the design of the map by manually tracing the outline of an existing Game of Thrones map in Adobe Illustrator. This was a very tedious process that took a fair bit of time, but using automatic functions for creating a path did not give a desirable result.
Once done, I exported the completed outlines to Adobe Photoshop to continue my work there.

In photoshop I started off by changing the colors to a brownish color, as I already knew I wanted to give the map a more antique look. Once that was done I duplicated the outline layer of the

landmass a couple of times, using one with a small blur to give a more defined outline of the map, and the second one with a more spread blur to better highlight the coastal areas.
I then used the “clouds” generator in combination with transparency to create the “wear” of the map. Then I put a blur on the entire edge of the picture with a deeper brown color, and finally a texture that looks a bit like old parchment paper.

I now had a map that looks the part, but had no details, so this was the next task at hand. With this in mind I first made a little low-res drawn picture of a simple mountain, which I then converted to a brush. Afterwards I did the same thing for a tree. Using the created brushes, and changing the brush behaviour to randomize size, spread and placement a bit, I used them to “paint” the vegetation on the map.

Now over to the locations. I started this process by making some icons in illustrator signifying the different types of locations the player could land on.

(Location logos)

I made one logo for coastal towns, one for normal towns, and another one for cities that would typically have a castle. I made a banner for these to display the name of the location.
Once these were done I placed them in approximately the correct position on the map, and I added a circle below each of them so to better display that the player could land there.

I was now ready to make the finishing touch on the map, namely the path itself.
I used a round brush with a dark green color, changed the spread so it would draw “dots” as I painted, then I manually made the paths between each location, making sure to make some twists and turns along the way. Afterwards I made some circles in between each location to signify all the steps along the way. I ended up with 41 steps, rather than the 30 that the assignment asked for, mainly due to it looking very “short” with just 30.

Token Design

(Player Tokens)

For the player tokens I started off by making the shape of a badge, and I experimented a bit with the placement of a line for the text to be displayed, until I settled for a text line on top, with a smaller inner border to break up the design a bit. I then proceeded to do the lengthy task of manually creating outlines from different sources of inspiration for each of the house crests I had already picked. The colors I chose for the different houses are mostly similar to what that house actually uses, with some minor changes as to give better contrast and visibility.

I also put a bit of a rough texture on the solid color background behind the figures, but in smaller sizes this is not really that visible. It does show on the final winning page however, as the player token is a bit bigger there.

In retrospect I think I used a bit too much time on the details of each figure. Since the tokens are fairly small, I could have gotten away with using a bit less time on getting everything perfect.

Character Card Design

(Character Cards)

When it came to making the cards that would be displayed on the front page for each character I only felt it would be natural to use much of the design I already made for the tokens, not only because it saved me work, but also so the player would easily recognize the character he picked.

I started off making a simple frame, followed by a shape that could be used for the name. I made a curved shape with the appearance of a “shield” that I put in the middle that the logo could go in the middle of. Finally I drew a sword, duplicated that and made the two swords cross, with a couple of shapes to make somewhat of a decorative top on the card.

Once this was done I only had to put the logo in the middle of the “shield”, and write the full name of the character the card belonged to. I decided these cards should be fairly short when it came to their height, as I had already planned for the information I got from the API provided to go below them.

Semester Project 2 Report

Prototype

As I had already thought about this assignment for my entire summer vacation, I already had a pretty clear image of how I wanted the layout of the different pages to be, thus I did not make any physical hand-drawn sketches of the webpage, I rather went straight to Adobe XD to make a prototype once I was done with the artwork

Prototype: https://xd.adobe.com/view/d48faebf-68c3-41ad-4771-3221dd349bea-a2be/

Frontpage / Player Selection

I started off by making a main logo for the webpage using the selected fonts and color to put in the top center. The page being as simple as it is would not need any additional links or dropdown menus.

Then I added a paragraph with some simple descriptive text as what the game objective is, followed by a header encouraging the player to select a character.

Then there's 3 buttons, one for player one, a second for player two and the final "start game" button. Naturally the character cards for selection follows right after the buttons, so the player easily understands what to do.

Looking at the prototype now, I see the spacing of the character cards being a bit much, so I'm glad I managed to fit 5 cards on a single line on the finished page, so that the area looks more "filled", and an added benefit is that it takes up less space vertically as well.

Gamepage

Keeping in the lines of everything being simple I again started off with a centered logo at the top, but I now added a "quit" button to the left and a "roll dice" button to the right.
This was a conscious choice as I had already taken the choice of using HTML canvas for the animation, and I knew I would have to have a canvas overlaying the entire map, so I didn't want any other html elements on top of it that might cause me trouble.

Directly following this is the entire game-map with a couple of tokens already placed for illustration purposes. I forgot to think about having something indication whos turn it is, and what position the player was on, but I added this in the development phase.

Winpage

Again simple and easy to understand was my key thoughts as I put the logo on the top center, followed by a bigger version of the winning players token.

This is followed by some text congratulating the player and a centered button to play again.

I had not thought about what to do for the animation part of the winning page, so this was again something I added in the development phase.

Development

HTML/CSS

Once I was done with the initial design and artwork phase, I started work on HTML/CSS. Using the prototype as a guide I first put down all the HTML elements, renaming, resizing and adjusting artwork as I went on. I also made sure to use proper “alts” for all images, and tried to make everything as organized as possible.

Once the HTML was in place it was over to CSS. I already had an idea that 5 character cards in a row instead of 4 might look better, so once I had adjusted the flex settings for the cards, I saw that it made sense. I also made sure to adjust everything on the frontpage to be responsive for mobile users as the assignment called for.

Although the pages was not the biggest when it came to HTML and CSS coding, the gamepage and the canvas overlaying the map-image gave me a bit of a problem. First I couldn’t get them overlaying at all, but after some trial and error, I managed to get that working using relative and absolute positioning. I then encountered the problem of canvas elements not showing, which gave me a suspicion that the image was actually on top of the canvas. Thankfully I was already aware of z-indexing, and this quickly solved that problem.

Javascript

Initially starting out I was thinking of using jQuery, and maybe a javascript api for canvas game development, and although it might have made my job easier, I feel like that wouldn’t have really shown what I was capable of.
So I decided to write javascript without any API to help me along, and although it gave me some big hurdles, especially when it came to the canvas animations, I feel the end result works well.

Semester Project 2 Report

Frontpage / Player selection

I started off doing the actual player selection function here, rather than fetching the API. I wanted to get that done first, and the API I could add afterwards as it didn't really interfere with the selection process itself.

For the buttons I added click event listeners that adds and removes html classes to them depending on which button is selected. This made it possible for when a user clicks a button, the already selected button gets unselected. It also made it easy to track which of the buttons were active for the character selection.
For the start game button I added a condition looking if any of the 2 variables for player characters were empty, and if so it would tell the user to select 2 characters.

I added some CSS effects for hovering over the character cards to highlight that they were items that could be selected. I then wrote a "for loop" that iterates through each of the cards, adding an event listener waiting for a click. Using the same method of HTML classes to see what's selected and unselected in combination with adding a border color around the playercard depending on which player selected it I managed to make the selection process work.
Depending on what player button was pressed, the variable for selected character is updated.

Now over to the API, something that gave me a bit of a hurdle to overcome. Since I had already established the character names during my artwork phase, and made logos for each house, having the API select random characters was definitely not an option.
I started out experimenting a bit on how the API worked, trying to fetch the info for the characters I had chosen.
After a bit of trial and error I realised the API has a paginator, limiting the fetch request to a maximum of 10 results by default, or up to 50 by changing a setting, which really would not help me at all. I had the idea of just fetching the entire API table, then sorting through it for the characters I needed and just pushing them to a new array, but that would obviously not work.

I did at one point make a function that went through all the pages of the api and combined it all to one array, but this was obviously not a viable option, as both traffic and time is horrible.

So after a while of more trial, error and research I managed to figure out that API url + ?name= + "FirstName+LastName" worked. I must say the API was pretty poorly documented, and I'm honestly unsure on how I actually found out that worked in the end.

At least now I had a way of getting the API information back on a single specified character. What I then did was make a loop that on each iteration ran a function asking for the return of the provided character in an array. I later on saw this giving problems from time to time, especially in firefox, as the page loaded before the loop was done, and it didn't have the character information to populate the HTML with.

I knew a solution to this is using promises, so I rewrote the entire script using promises.
After a bit of reading I saw that I could use promise.all for checking that everything is fulfilled before continuing, so this is what I did.
I made a “for loop” that ran through all the chars in the array, pushing each result to an array, and once all promises are fulfilled it runs the loop populating the HTML information for each character card.

The final task for the frontpage was to keep the information of which character the user selected on the transfer over to the gamepage. I found a couple of ways to do it, but using “localStorage.setItem” seemed to be the most fitting one to use.

I now had a fully working frontpage that populated the API information, and let the user select the characters for player 1 and player2, which was then kept in storage when the user clicked “start game” and was moved to the gamepage.

Gamepage

The first thing I did for the gamepage I actually did before even making the frontpage. I had long thought about a way of actually getting the coordinates of each of the stop locations on the map, so this was something I wanted to get out of the way.

I got the idea of making a simple script that took the coordinates of the mouse when clicked, and console logged the number of the waypoint, as well as the coordinates.
Once I made the script, getting each location was as simple as manually clicking on each location on the map in order, and once done I had a console log of all the coordinates. I could then export this, and format it as variables.

Initially I had the waypoints as separate variables, but I quickly realised that was pointless and changed it into an array.

I won’t be able to go into detail on all the different various steps back and forth, fails and successes and troubles this part of the assignment gave me, as I would go way over the word count allowed for the report, but I will detail the important ones. The complexity of the script grew quite a bit as I went on, and I’m certain I could probably shorten it down a bit and make it easier to read, but time was not on my side anymore, and making it work was the most important thing.

I started off making a simple script for the dice roll using math.random, which I attached to a click event listener of the “roll” button. I then made a function for setting up and running the canvas itself, drawing the initial positions of the player tokens, which I had previously gotten the variable value from the previous page using “localStorage.getItem”.

Now I will have to stop myself from going into too much detail, but this is where my world crumbled, and I got stuck for 3 days bashing my head against the wall with the same problem.

The problem became apparent when I wanted to do something I initially thought would be simple enough to do, simply moving the player token from one location to another. I mean, I had the starting coordinates, and the dice number the player got from the roll, so it's just a simple process of adding the number to the current player location, then using this result to pick the next coordinate from the waypoint array. So in fact I had the starting X and Y and the upcoming X and Y.

It quickly after a bunch of research that canvas does not really have a simple "move to" function to add a couple of sets of coordinates to make something to move from one position to another, so this would have to be calculated manually. After a bunch of reading, I become an expert in making something move in a straight line, but not stopping…
Not really helpful in my case. On day 2 of frustration there's a light at the end of the tunnel though, combining different sources of documentation and tutorials I'm able to calculate the amount of time to move based on the length between the points, and what direction to move as well.

Great! I can now move the token from the location it was on, directly to the position it rolled. But boy did that look bad, it obviously needs to go through each of the points in between as well, it can't simply jump directly to the location rolled. Not to mention that when the tokens land on the same location they overlap each other..

The tokens overlapping each other was fixed fairly easily by conditions and basically having them move a bit to the side of each other if on the same location. Making them go through each location on their path to rolled location wasn't that easy though, even if I knew logically that I just had to run each part of a roll in sequence. As in, you roll a 4, go forwards by 1 position 4 times. The solution I came up with in the end basically runs the animation cycle once for each location increase, checking each time if it's at the desired location, if not, run again.

Progress has been made, I have tokens that move step by step to the correct location based on the roll of the dice. The assignment asked for the user to get a second turn if he rolls a 6, so I added a condition checking for this, and I also made a new canvas element in the form of a text box, to be used for both this, and the events that is triggered if the user lands on a city.
I also made a function for a button to be made in canvas, so if the user lands on a city and gets a punishment, he has to click "ok" before the punishment is given and the token is moved back.

The condition for giving the user a second throw if he rolls a 6 was easy enough and didn't really pose any problems, the player punishment on the other hand was a bit more tricky, as it again required my to animate the token, but in the opposite direction this time. After a bit of fiddling I made it work by making a separate animation sequence for the punishment functions where it subtracts the position rather than adding it, completing once at the desired location.

The final problem I remember off the top of my head I discovered during testing, and that was that sometimes during the pageload of the gamepage, only one of the tokens would show up. I immediately thought this would be because it loads the canvas and the included function to draw the tokens, before the user had been able to load the image, thus having no image to load.

This was solved by using promise.all again, but this time checking that all images were loaded before running the function to initiate the canvas. And to my surprise this worked flawlessly on the first try.

As I said, I could write page after page going into detail on what I did on this page, but I feel I managed to relay the important issues and I'm already running out of words to use.It definitely was a good learning experience.

Winning Page

Still reeling from some really rough days making the gamepage work as intended, although also proud that I did, I set to finish the winning page. The initial part of making the page display the player token that won and text saying player 1 or player 2 was easy enough using "localStorage.getitem" to get the correct variables given from the gamepage with the winners information. Running out of time due to the gamepage coding taking too long, I decide on some simple "snow" particle animation for the page, that was fairly easily accomplished using a tutorial.

I had an idea of making some swords that would move into place etc, but with the humbling experience I just had animating on the gamepage, I simply wouldn't take the risk of running out of time.

Notes

As a final touch to add a bit more of a "game" feel on the entire thing I added sound effects for button clicks, player movement, dice rolls, landing on a punishment location and winning the game. I think it adds another dimension to the game, and was easy enough to do.

Conclusion

Starting this report I was thinking on how I could possibly manage to write 2500 words about this assignment, but when I started writing it quickly dawned on me that I've done way more than I remember, and now I'm close to 500 words instead. There's a bunch of things I've forgotten to mention as well as other minor things I've left out, but I think I managed to get the jist of it.

The assignment as a whole has been perfect, basically challenging me in all the parts we've gone through for the past semester, even without specifically asking me to do so. I've had to use promises, API functions, fetching, not to mention Canvas, more specifically Canvas animation.
I get that an assignment of this scale is mainly to show off what you've learned the past semester, but I also learned an incredible amount while doing it.

The only thing I could possibly point out is that there is a clear lack of decent Game of Thrones artwork that is not copyrighted, so it could maybe be included in the description that it's not easy to come by and might be better off making it yourself. I don't see this as a negative though, as I really enjoyed doing some artwork for the page as well, even if it took some more time out of my budget than I originally accounted for.

As I previously stated, if you want the source .psd file for the map design, please let me know and I'll figure a way of uploading it to somewhere. I tried including it in my git repo, but after trying to push for X amount of hours, I simply gave up.

Sources

Resources:

Sound Effects
https://freesound.org/
City and Character names
https://gameofthrones.fandom.com/
Canvas snow effect
https://www.html5canvastutorials.com/advanced/html5-canvas-snow-effect/

Fonts:

Game Of Thrones font
https://fontmeme.com/fonts/game-of-thrones-font/
Cinzel
https://fonts.google.com/specimen/Cinzel
Cinzel Decorative
https://fonts.google.com/specimen/Cinzel+Decorative

Software:

Jetbrains Webstorm
Adobe XD
Adobe Photoshop
Adobe Illustrator
Gitkraken
Gitkraken Glo
Google Docs