Coursework 2 Report

Richard McMaster

40280868@napier.ac.uk

Edinburgh Napier University

Module : SET09103

Burke & Hare Story

Introduction

The Burke and Hare Story is an app that takes the user through a fictional version of Burke and Hare's events. The story has seven chapters with four possible outcomes depending on what option the user chooses in the last chapter.

The intention was to create an "Adventure" game type app that was driven by GPS location. The idea was, the user would have been guided around Edinburgh (see appendix D) to each location in a set sequence and each chapter would be unlocked once the user was within a set distance of the location, in this case it would have been 50m.

Along with the "Adventure", there had to be a way the users could get to the story without physically having to go around Edinburgh. This had to be done so if the app was accessed on a desktop or laptop which are generally static devices, the app could still be useful to them.

My development approach was to create the static version first. If we had the story content displaying as it should then the Adventure could simply be a process that controlled the access to each chapter.

Design

The site has been designed visually to be relative to the time period of the 1800's. I done this by creating custom css and by using custom graphics/images I created myself or had access to from another project.

There are 2 main layouts used. On larger screens such as tablets and desktops, the user can see both pages of the book. All images and audio controls are always on the left and text content is always on the right. Then on smaller devices, they only can see one page at a time, but this is presented to them as either an audio version or a text version of the story which works well, I think. See appendix A, B and C.

This app currently only uses one json file that holds the details for each chapter in the story.

Chapter json structure example…


```
{
"chapter":"1",
"chapter_title":"Deacon Brodies Tavern",
"audio_src":"location_1.mp3",
"text_content_source":"location_1.html",
"location_lat":"55.949534390311946",
"location_lng":"-3.1928888538919864",
"main_image":"location_1_1.png",
"old_map_img":"location_1_map_img.png",
"secondary_image":"location_1_2.png"
}
```


By doing it this way it is possible to create new stories by creating a new json file and following the same structure. If the user selected a different story, we could use that json file to display the correct content.

As you can see in the json file, some values are just filenames. To do this, I created subfolders within the apps static folder to hold this content. There is an audio folder which holds all the mp3 files for the chapters so the python code already has '../static/audio/' in the src path and I just add the json value to the end. This same method has been used with regards to html content and images.

Example…

```
src="{{ url_for('static', filename='images/story_images/'+story["main_image"])}}"
```

Each chapter and outcome has its own app.route. The root will display the introduction chapter, but each chapter or outcome has its own route such as /chapter1/ or /outcome1/.

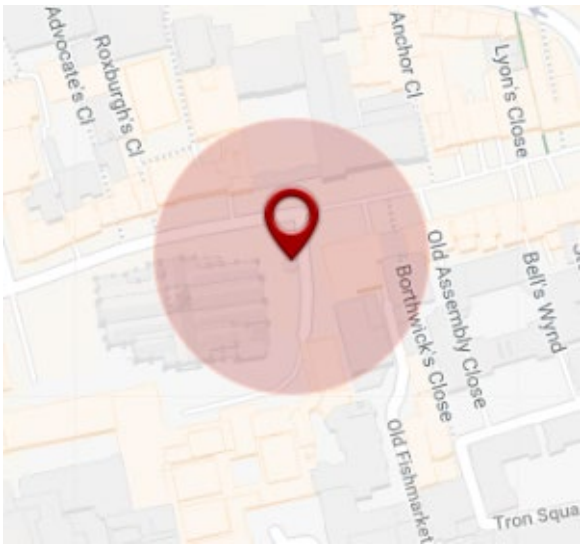
As part of keeping the sites theme, I had to create my own media player. I done this by creating my own Icons, css and javascript functions to control the media player.

Enhancements

GPS Location

In the end I ran out of time and into a technical issue, I can't get a devices location unless the server the app is running on is secure. I have done previous work with GPS and know how to calculate the distance between two latitude and longitude coordinates, the problem is I can't get the users coordinates as it is not running on a secure server.

If I could have hosted this on a secure https server, I could have used the google maps api to check the users location compared to the chapters location, for example…



If the user was within the red circle, that chapter would become unlocked to the user.

User registration.

For the adventure to work, we would need to remember each users individual process. At first, I thought it could be achieved with cookies alone but realised it wouldn't be enough. Cookies on their own would technically remember a "devices" progress rather than the "users" progress. If cookies were deleted from that device, all the adventure progress would be lost. To account for this, I would make the users register to receive their invitation to the adventure. This way the user could log in from any device at any time and recover their progress. The json could be something as simple as…

ref	1	2	3	4	5	6	7	last_active
fxEdfR475966ChGS	0	0	0	0	0	0	0	date

This is how it would look when the user started an adventure and as the user unlocks each location, we change the 0 to a 1 in the record. This means when a user returns to the site, we check for the cookie value on their device, if there is one, we match it to a record in the

database and we know what that users next location should be. If there is no cookie value, the user is given the “Start Adventure” or “Login” button.

Users are timed.

As part of the adventure, I thought it would be good if the users were timed between locations. The users could be given a different sequence of events depending if they get to the next location on time. I would do this by recording the date/time of when they requested the location for number 2 and then checking the date/time of when they got there. If it was within a set period, they would get one option, if not they would get another.

Critical Evaluation

Display/CSS
Overall, I like how the story mode turned out. There are some issues on certain screen resolutions, but these would be tweaked given more time to test the app.

Originally, I had the story controlled with javascript and then changed it to python at the end, this has left me with an issue where if the user is on a mobile device, looking at the text mode and then click next chapter, it opens the next chapter but in audio mode. This is because audio mode is the default view. I would fix this by passing a value in the url that stated which mode the user was in, I would then read that value and set the display mode when rendering the page.

HTML Content
For each chapter there is a relative html file within a content folder for example, chater_1.hml. This simply holds the text content of the that chapter. At the start, I was expecting there to be more content in html file such as images which turned out not to be the case. Looking at it now, I should have just entered the text directly into the json file instead of including a file. This would make it a lot more scalable and would require less disk space on the server.

Personal Evaluation

In the end, I feel I spent too much time on css trying to make it look good and not enough on functionality.

At first, I had developed the functionality to work mainly with Javascript. The reason for this is because I wanted to create it in a way where it didn’t necessarily need to be ran in python so now, I have a version of this site saved elsewhere that will run purely with javascript.

Doing this eat into my development time and nearer the end I had regrets about doing this. Although “Story Mode” is complete and it wouldn’t take much more work for the adventure side to control when content is displayed, if I had started out developing the app in python, I feel I would have been able to get a lot more work done with regards to the Adventure.

Overall, I feel confident that I could have developed this and made it all work, but I did under estimated how long it as going to take.

Knowing very little in terms of using bootstraps for styling, it does seem to be a way that I could have maybe developed this quicker by having a bootstrap take care of the layout etc…

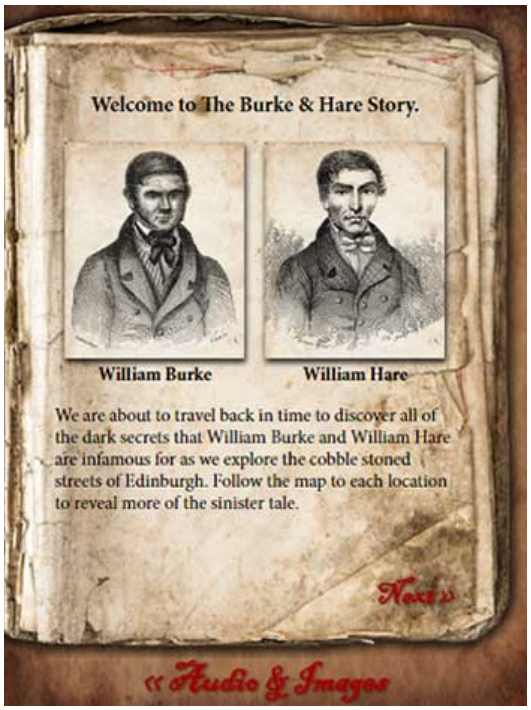
This is something I will look at in the future.

Appendices

A.

Mobile – Audio Mode

B.

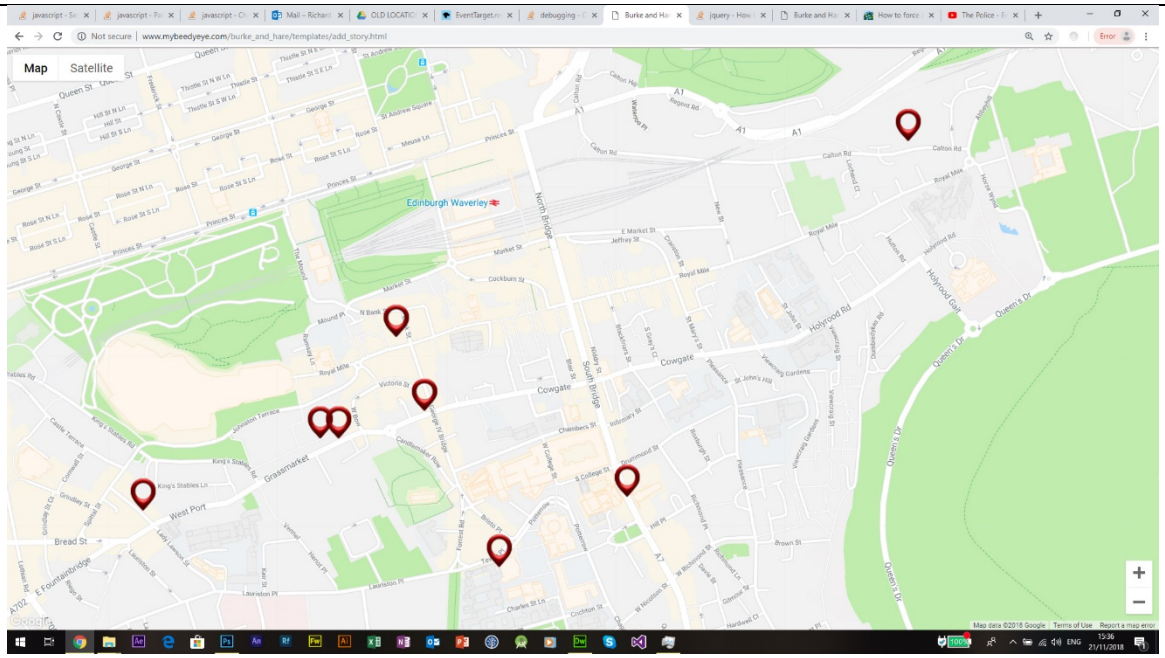


Mobile – Text Mode

C.

Desktop – Full View

D.



Chapter locations