Lee, Doo
CMPS 112
Winter 2015

Final Project Report

Introduction

Here I present a simple interactive text adventure game written with the following programming languages:

-Haskell
-C++
-Javascript/Jquery and HTML/CSS.

The goal of the project was to create a game in each of the mentioned languages, showing how different programming paradigms are used to make the identical application.

Haskell was chosen to represent the style of functional programming. It is a purely functional language, which means it is lazy and has type inference. It is lazy in that it does not evaluate until necessary.

C++ was chosen to represent the style of object-oriented programming. This popular paradigm is modeled around the use of objects to fulfill a program.

Javascript/Jqeury and HTML/CSS were chosen to represent the style of procedural programming as well as graphic user interface and its interaction with the logic of the code.

The Rules of Game

The game places the player in a large desolate land in which the player must find the treasure that is placed randomly on a map.

In the Haskell version, the user simply inputs one of the following characters: n, s, e, w to indicate which direction the user is going (north, south, east, west).

In the C++ version, there is an added hostile enemy object that will eat the player alive if the player goes too close to it.

In the Javascript version, the player can click on one of the four directional images to indicate which direction the player wants to go, but there will be no monster object. (the monster object has been added to simply showcase the power of the object-oriented programming).

Development Process

Haskell has been quite challenging in that it required a significant amount of time to really minimize the code into something simple that could be understood easily. At first, I have started with more lines of code than what I have submitted.

I was left with two choices:

```
x  <- randomInt
y  <- randomInt
x' <- randomInt
y' <- randomInt
```

or

```
[x, y, x', y'] <- replicateM 4 randomInt
```

to extract all four at once. I decided to go with the initial choice to render the code more easily readable for the grader to see what is exactly occurring within the logic of the code.

As for C++, it was a language that I was not fully comfortable with, but due to the large amount of resources available to me to give me a full understanding of the language, the object-oriented portion proved to be quite simple since I was already very familiar with working with objects with Java and C. Thus, it was easier than Haskell and took less amount of time to complete. I have added a monster and mapping feature.

The javascript portion was quite challenging in that I needed to really learn how Jquery can be used to work with Javascript's logic. The BindEvents() function is where I am relating to the HTML page. Using the Jquery library I can easily "select" the parts of the HTML I want.

For example,

the HTML:

```
<a id="PlayAgain" href="#">Play Again?</a>
```

Has the id "PlayAgain". Using jQuery, my event is assigned as follows:

```
 $('#PlayAgain').on({
            'click': function(e) {
               e.preventDefault();
               $gameend.hide();
               NewGame();
            }
         });
```

The "selector" part: $('#PlayAgain') lets me select the a tag, and then using .on() I can assign events to said tag. In this example I assign the 'click' event, and attach a function to run when it is clicked. The "e.preventDefault()" prevents the link from trying to follow its href attribute, and then I'm simply running whatever I need to inisde the function.

Challenges

The only challenge for me was learning C++. Professor Flanagan is an outstanding lecturer who really helped us understand the aspects of functional programming, from list comprehensions to typeclass declarations. So I was able to implement the Haskell part without much difficulty. The Javascript portion was quite challenging in that I needed to coordinate among 4 different languages, including Javascript, Jquery, HTML, and CSS. It was a very great opportunity for me to really learn how these can work together to form a web application. And finally, the C++ portion proved to be quite difficult in that it was a wholly new language for me to learn (C++ is really awesome! I have never learned the language until my peers recommended that I learn to appreciate its beauty.)

Resources

http://www.tutorialspoint.com/cplusplus/cpp_classes_objects.htm
https://hackage.haskell.org/
http://www.amazon.com/gp/product/1118531647/ref=s9_simh_gw_p14_d0_i4?pf_rd_m=ATVPDKIKX0DER&pf_rd_s=desktop-1&pf_rd_r=17R2JPACTHA3KFSJH9Y5&pf_rd_t=36701&pf_rd_p=1970559082&pf_rd_i=desktop
http://en.wikipedia.org/wiki/Associative_containers
http://stackoverflow.com/questions/7265550/haskell-could-not-find-module-system
https://hackage.haskell.org/package/random-1.0.0.1/docs/src/System-Random.html