

Hunter Game

Chewie (Shuiqiang) Lin

March 11, 2016

1 Introduction

In this game design, I wanted to create a single player game where the player traverse through all the spaces acquiring different items that achieves a goal in some way. Some initial ideas are traversing through a maze in order to find a way out or evading predators like lab 5. In the end, I decided to go with a RPG game where the person try to survive in the wilderness.

The plot of the game started simple where the person gathers food but was killed in the end due to food poisoning but abandoned that for something else. It would be more fun if there was some danger so I incorporated code from previous tournament game assignment and added animals into the game play where the player can die of injury. This makes the game more fun with risk and challenges.

2 Requirements

There are several requirements needed for this final project and each one had challenges in the process. One of the requirements is to have a backpack/bag to store items so I also created an item class. The space class and its children classes are challenging because they have to be linked to each other though pointers and allow the players to do differerent actions in each place.

During the inital design of the game, I figured I needed a couple more classes to keep track of these elements. From there, I made a game class to track the player space pointer, backpack, and initialise places like the campground, river, forest, etc.

There are many other miscellaneous details such as possible actions available, items found, environments during the design phase that may or may not have been implmeneted. As mentioned before, I later adopted animal creatures into the game to make it more fun but that altered the some of the details in the game.

Members: space ptr, backpack list, campground(),

Functions: getDirection(), getList(), setList()

Space class

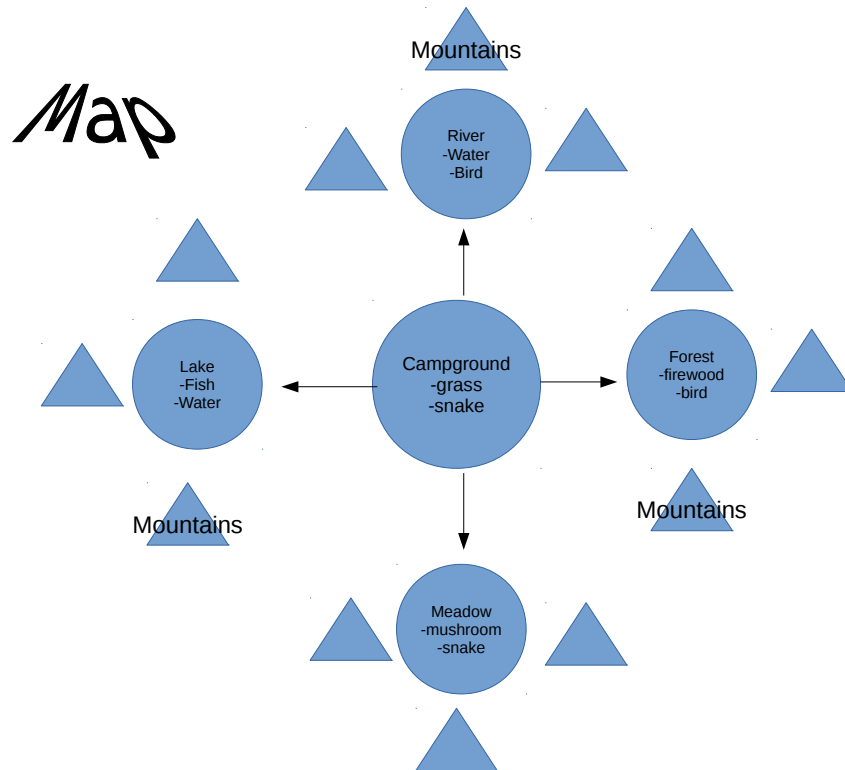
Members: string name, item list, pointers, animal

Functions: Find(), Gather(), display(),

campground class

Members: derived from space class **Functions:** derived from space class and battle(), gather()

Other derivatives of space class are forest, meadow, lake, river, and mountains. The following picture shows the layout of the entire landscape in the game.



3 Program Outline

Listing 1: main outlines

```

//Show the option to the user
Menu()
    -Mission
    -start
    -Quit

Game()
    -Intro
    -Action list
//action list

//Interface to the user

Location:      Strength:      Time Left:

*picture

Find(), Gather(),
go east, go west, go north, go south

What to do? ...
  
```

4 Test Plan

input start game

values 1 to start

purpose test the game space and see it works crashes

function new space(), campground();

expected outcome: display welcome msg indicate it works

observed outcome: successfully display the campground name

input Actions

values 1 - look, 2 -Gather 3- Attack

purpose test functions in the space before replicating it for other environments. Each one of the functions are developed and test separately.

function show(), find(), addItem(), attack(),

expected outcome: list resources, animal, items in bag, kill animal, and gather meat

observed outcome: Separate outcomes for listing of resources, animal, add items.

notes: Other functions such as check requirements, throw items away are added later in the game.

input directions

values w - go west, e - go east, n - go north, s-go south

purpose test all the space links and and check resources, animal, etc are correct.

function river(), forest(), meadow(), lake()

expected outcome: see different item and animal in each space

observed outcome: Initially it can only go east and west but later added more spaces.

5 Analysis

The game has changed quite a bit since the initial design from game flow to implementation details. The most significant change was introducing animal creature to the game. This element makes it more fun and increase the number of ways of dying or ending the game. In order to make it work, the original fantasy game creatures were modified. For instance, barbarian were changed to human, vampire were changed to bird class and fish class because of its ability to evade attack. And medusa were converted into snake that can kill its prey with poison.

A large amount of detail work went into designing the 5 different environments. Each geography gets a different ascii artwork to clearly show a different space but also items and animal. In the interest of time, I decided to limit one item and one animal to make it easier. Also, instead of killing the player with poison mushroom, I allowed the animal to attack the human as well which makes it a better survival game overall. These features were later added to the game but they took more time to test and debug as well.

Perhaps the most frustrating details to implement was the item list. In order to end the game, the bag need to have all the required items but I went down the wrong path of removing and adding items on the list which caused many segmentation fault errors. In the end, I created a third list that I can modify anytime and compare to the mission list.