COMP-206
Software Systems
Assignment #4
Due: April 11, 2017 on myCourses at 23:30

Retro Multi-player Web-based Dungeon Crawler Game

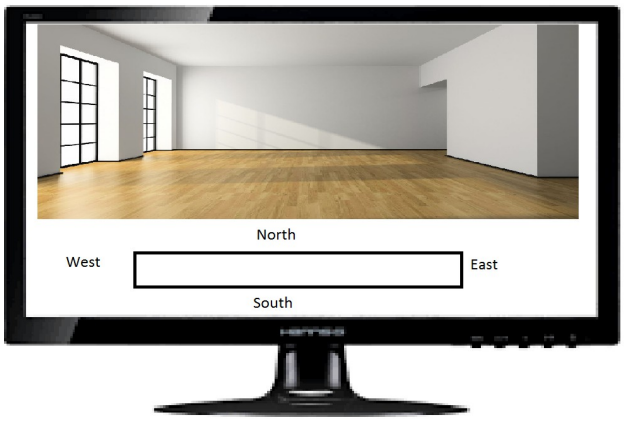
You can do this assignment alone or in a team of 2 or 3 students. Three students is the recommended number. There will be a Google doc where you must register your team. We will use this doc to assign the TAs to grade the submissions. You will need to provide your team member names (even if you are working on your own – just write your name) and the name of your team.

Our dungeon will be a series of interconnecting rooms. Each team will build one room. The room will have a single challenge in it. Players will use up or gain manna and gold pieces. Every room will have four exits that each lead to another team's room. Given that we have about 300 students we could have upwards to 100 or more rooms. The player dies when their manna goes to zero. The player wins when they collect 100 gold pieces.

Detailed Description

The Room

Every room will have the same layout. It must be patterned after this diagram:



Every room will have 7 elements: (a) a picture of the room, (b) an input text box where the user can enter game commands (with a submit button), (c) 4 teleportation buttons called North, South, East, West that when pressed cause the player to exit the room and teleport to another team's room, and (d) an output text area where the room can display information to the player in response to their action. You will need to get the URLs of other teams in the classroom.

Room Theme

Your room can be anything you like. The theme can look like anything you want. The challenge or Boss can be anything you imagine. The only restrictions are: you have to program and build everything yourself, it must be legal, it must be respectful, and it must follow the rules of the game (described next). Have fun with this !!

Game Rules

Only one player can be in a room at any time. Players begin the game with one manna and zero gold. A player dies when they have zero manna. A player wins when they have 100 gold.

The room begins with 10 gold pieces and 10 manna hiding behind a challenge (or Boss). If a player wins the challenge (defeats the Boss) they get access to the hidden manna and gold. They can take at most 5 units of something: 3 manna and 2 gold pieces, for example; or 5 manna and 0 gold pieces. These amounts are deducted from the hidden resource. The next player who comes to this room will see what is left.

Players use up 1 manna to transport themselves. When a user transports into a room it is the job of the room to deduct the manna (adding that manna to its hidden resources) and checking to see if the player is at zero manna. If the player is at zero manna then the player does not see the room but instead some other end of game scene (up to you what that might be, but must be respectful and legal).

It is the job of the room the player is exiting from to check if the destination room is occupied. If it is not occupied then the player transports otherwise they see a message telling them that the room is occupied and they stay where they are.

Players do not use up manna attempting to defeat the Boss or win the challenge. They can try as long as they want.

Players can drop 2 gold pieces to get 1 manna from the universe. This manna does not come from the hidden resources of the room, but the dropped gold is added to the hidden resources of the room.

If a player succeeds in getting 100 gold pieces then it is the responsibility of the room the player is currently in to realize this and change the screen to some other end of game scene extolling the virtues and bravery of that player. You do this in any way you like, but be respectful and legal. There is no exit routes from this win state. The game is now over.

The player can exit the game by typing the command EXIT in the text box. It is the responsibility of the room the player is in to display an appropriate sorry to see you go screen (legal and respectful) and then take all that players manna and gold adding it to the hidden resources of that room. Players also end the game by winning or through death. In the case of death all the player's resources are added to the hidden resource of the room. In the case of a win the room does not get the player's manna and gold pieces. Players can also terminate the game by closing the browser at some random moment. Closing the browser will be a problem because the room will still think that you are occupying it, so let us not do that...

You begin the game from your room. The room is marked as occupied.

Technology Summary

- The room
  - HTML and CGI
  - The initial room is created as an HTML file called room.html.
- The room's hidden resources
  - CSV text file called resources.csv
  - Format: manna, gold, occupied
  - Eg: 10,4,1
    - 10 manna and 4 gold pieces available, and the room is occupied (number zero for not occupied)
    - Manna and gold pieces can never go below zero.
    - Players cannot get manna or gold pieces when at zero
- The challenge or Boss
  - Either a C program or a Python program, your choice.
  - You must write this program, do not take it from somewhere
  - The player interacts with your game by typing PLAY in the text box
  - The interface cannot change. The game must display information to the screen and the player has to reply by entering something in the text box. This does not need to be fancy, you could redraw the entire screen after each interaction. When playing the game you no longer see the North, South, East and West buttons.
  - The player can stop playing at any time by typing QUIT. The player returns to the normal room display having the North, South, East and West buttons.
  - You could make a guessing game, a riddle, a fight game by typing a word like ATTACK and then see what happens randomly, anything you like (again must be legal and respectful).
  - Display simple instructions so that people know what to do.
  - If a player wins then display your hidden resources and ask then to take some of it (following the rules).
- Text box processing

- C program called room.c and a.out.
  - The C program receives as input the player's inventory and the command the player entered.
  - It executes the command and redraws the room with the player's inventory.
- Supported commands:
  - DROP n
    - Drops n gold pieces. For every 2 gold pieces the user gets 1 manna from the universe. The gold pieces are added to the hidden resources of the room and removed from the player's inventory.
  - PLAY
    - Activates the room's game
    - Game can be terminated early by typing QUIT
  - EXIT
    - Marks the room as not occupied and all the players manna and gold are added to the room's hidden resources. Sorry to leave scene is displayed.
  - REFRESH
    - Just redraws the screen with the player's inventory preserved.

- Player's inventory
  - CGI hidden tag using CSV string
  - This must be regenerated after each CGI interaction to reflect the player's “inventory” of manna and gold.
  - Format:
    - <input type=”hidden” name=”inventory” value=”5,130”>
    - This player has 5 manna and 130 gold pieces
- Transporter program (a single program used by North, South, East, West)
  - Python program called transporter.py
  - This program receives two parameters:
    - The player's inventory
    - The URL of the room the player came from (to the a.out file)
  - The program first checks to see if the room is occupied
    - if it is occupied then the program uses the URL to call the C program to regenerate the room the player came from.
    - Passes the player's inventory using the command REFRESH
  - If the room is available then transporter.py is responsible to generate the room including the player's inventory as a hidden field. You can do this either by calling the a.out file or by generating the initial page from the Python program.

FOR THE GLORY

Make it pretty. Upgrade the website with CSS or JavaScript.

HOW IT WILL BE GRADED

Points removed for bad practices:

- -1 for not following instructions
- -1 for not indenting, spacing, and/or commenting
- -1 for not using good variable names

This assignment is worth 20 points:

- +1 : Home Page
- +5 : C program
- +5 : Python program
- +2 : CSV file processing
- +2 : Player's inventory processing
- +3: Challenge/game
- +1 : Win processing
- +1 : Death processing
- Glory question comments from TA
    - Best of glory solution will be posted on the announcements section of myCourses by each TA for the group of students they are grading (if anyone qualifies).