

Andrea V. Nieves Rivera
Christian Matos Rivera

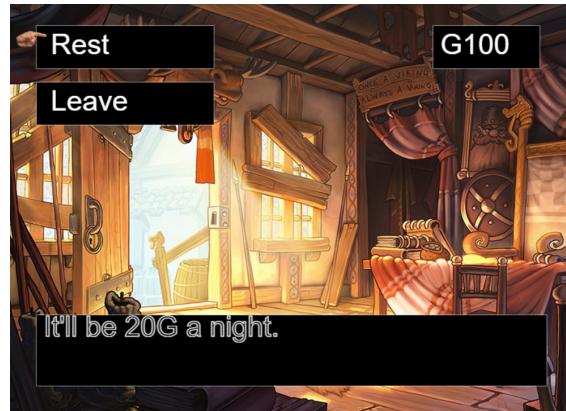
Final Project: First Person Dungeon Crawler Game

Purpose and Usage

This project is a first person dungeon crawler game. The objective is to clear the stage by traversing the dungeon and reaching a final boss. As the player progresses the dungeon, they may find random encounters of enemies that they can fight through turn-based combat. In this combat, they have control over 2 characters, their attacks, their spells and a set of items that can be bought through a shop found in-game.

To run the game, open an HTTP server with python -m SimpleHTTPServer 8000 within the dungeon-crawler folder, then open the local host on a browser.

The player begins in the town menu, where they may access the Inn, Shop, Bar, or Leave to enter the dungeon.



Inn

The Inn is a menu that can be used to Rest. When the player rests, all Health Points and Magic Points of all characters are restored. Resting costs 20Gs, which is the game's currency.

Shop

The Shop is a menu that can be used to buy health and magic potions and add them to the inventory. They can later be used to heal characters or increase their magic points and they cost 30Gs and 50Gs respectively.



Bar

The Bar is another menu, where the player may talk to some characters.

Dungeon

The dungeon gives access to the main map of the game. There, players can walk and encounter monsters through random encounters.



Combat

When a player encounters monsters, they are randomly generated from a master list of enemies. Enemies can attack the player with randomly selected targets. They will be given options, such as attack, magic and items. The right side contains dialogue of the last move that occurred. Once all monsters are defeated, the player may move once again.

Project Architecture

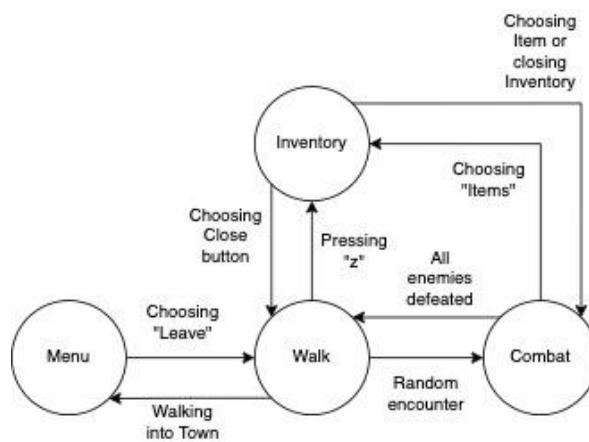
Model

The model of the project consists of:

- facing: Where the player faces in the dungeon
- visible: An object that describes the visible tiles to the player
- position: Tile position of the player
- encounter chance: A coefficient to increase the chance of enemy encounters
- money: Current currency of the player
- inventory: An array of items the player has
- battle cycle: An array that contains the players and enemies on the field when in battle
- enemies: An array where randomly generated enemies are stored
- lost: Turned on when all characters are defeated
- chosen attack: Stores an Attack or Spell to be executed
- state: Describes whether the player is walking, battling, in menu or in inventory
- menu: Contains the options the player can have on a menu, depending on the state
- selected: Contains the selected menu item index
- current line: Contains a string with dialogue lines to be displayed outside battle
- combat line: Contains a string with dialogue to be displayed in battle

View and Controller

The project makes use of different states to answer differently to user interaction. These states are: menu, inventory, walk, battle, character select and boss. The following diagram describes the interactions between the mains states of the game:

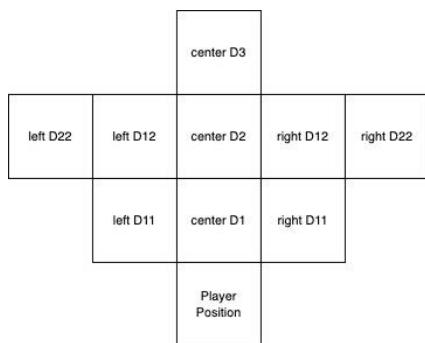


The states are used to modify the controller and the view as needed for proper user interaction.

Technical Challenge

The meat of the project consists of the dungeon exploration and its proper display on the screen. To simulate a three dimensional perspective utilizing only two dimensional images we needed to have a scale and a way to skew images properly. We found a function `drawImageInPerspective` that displays images in a skewed perspective given the four corners where the image would be placed. After calculating the correct numbers for perspective display, we were able to simulate depth in the display of images properly by creating a wireframe around those coordinates that depends on what is visible for the player..

The next part was figuring out how to display the blocks in front of you in the direction you're facing. For this we utilized a “class” `Visible` that stores the different indexes that are visible to the player at any given time, up to a depth of 3 blocks forward. Through the position in which the player exists, and the way the player is facing, we determined if the spaces in front are walls or empty spaces. Accordingly, we would draw the blocks or empty spaces from back to front. This way, the empty spaces would allow you to see blocks behind the space as long as the walls are close enough to where you’re standing. Below is a diagram of what the `Visible` object represents:



After figuring out the proper display of the dungeon perspective and modifying it accordingly as you moved through it, the rest was implementing menus and modifying the model to give more functionality like combat, item consumption and more through different algorithms..

REFERENCES

`drawImageInPerspective` function by Tom Rubaj on stackoverflow:

<https://stackoverflow.com/questions/10426887/how-to-skew-image-like-this>

Skeleton, Dragon, Demon and Imp images from Dragon Quest, artist Akira Toriyama

Town image from Final Fantasy IX, Lindblum, exact artist unknown

Character portraits and designs by @phereseisles on twitter

Wall from this site: <http://superwalrusland.com/ohr/issue26/pa/pixelart.html>

Shop, Inn and Bar background images by <https://www.artstation.com/danielthomas>

