Abstract

Dungeon Crawler is a 2D top-down dungeon exploration game featuring procedurally-generated rooms, floors, and enemies. The player can select their class, check their inventory, equipment, character status, and map, then navigate through the floor while fighting enemies, and ultimately descend down into the next floor.

Introduction

This milestone paper will run through a summary of my Dungeon Crawler project, an in-depth list of the current system functions accompanied with a UML diagram, as well as a list of the system's features, and planned features that went unimplemented due to time constraints, as well as a user manual, the requirements for the game, and a conclusion.

Current System Description

At present, the dearth of actively-used classes in the Dungeon Crawler game are spread out between the Dungeon Crawler main class, the Game class, the Player class, the Character class, and the Hero class, which is a subclass of the Character class. When a the program runs, it extends the Application class and overrides the start method, then creates a new scene object and passes it to a Game constructor before starting the stage with the scene.

The Game object then stores the scene and creates a new Player object, player. From the setupScene method, the Game class loads a number of different pane objects into the scene's root; this is what creates the GUI which the Dungeon Crawler user sees. The first screen simply has a button on it labelled "Start", which will then load a screen which allows the user to pick between three available player classes: Warrior, Rogue, or Mage.

When the player selects one of these, a method is called at the onMouseReleased event which passes the selection to the Game class's Player object, which in turn passes it to its own Hero object. The Hero object then set itself up based on the selection, which affects how many health-points (and mana-points, if the user selected Mage) that the character starts with, as well as what gear they have to start with, and finally their sprite sheet for their walking animation.

After this selection occurs, the user sees the final screen, which is a top-down view of a blank, rectangular room. They can control their character's movement around the room with the arrow keys or WASD keys; this takes place because the scene being depicted will run the Character class method "animate" when the onKeyPressed event is fired. The animate method determines the direction the character should move based on the key pressed, then cycles through the appropriate sprites on the sprite sheet instantiated by the Hero class while adjusting the character's position accordingly.

The user can select between one of four buttons across the bottom of the screen labelled "View Equipment", "View Inventory", "View Status" and "View Map"; upon being clicked, each button will open a new screen showing the player the resultant view. The player can move between rooms, which will usually be full of enemies. The enemies will move towards the player, and if they get within a certain distance, combat will start.

Combat is different based on which class the player has selected. During combat, the player can take certain actions that either involve attack the NPC enemies or moving back and forth between "slots" on the battlefield, which affects how easily they can land attacks and how

easily the enemies can hit them. After all enemies are defeated or the player dies, the battle ends, and depending upon which of the former caused it the player is either shown an end screen, or is allowed to continue exploring the dungeon.

Unimplemented System Description

Features which went unimplemented include a system for levelling up the player character, a system for rewarding the player character with looted items after they defeat an enemy, and a system where the player can trade with and hire peaceful NPC's to help them as they delve deeper into the dungeon. Beyond these major systems, there are also several less-significant quality of life sub-features, such as more icons for items, more types of weapons and armor, more types of spells for mages, and so on.

Requirements

Being a recreational activity as opposed to a tool or utility, Dungeon Crawler could not be said to address any problems in particular, aside from perhaps boredom. Conversely, however, purpose of the game is to provide entertainment for a consumer; or, more simply put, Dungeon Crawler's "requirements" are that it is enjoyable to play.

This goal is, unfortunately, slightly more nebulous than just writing a program for automatically calculating the circumference of a circle given its radius. Entertainment-value is qualitative rather than quantitative, and subjective rather than objective, both of which make it difficult to know for certain if Dungeon Crawler fulfills the criteria.

That said, there are some functionalities which seem universal, such as: making the gameplay easy for a player to grasp the mechanics of; making the game challenging enough that players do not become bored; and making the game varied enough that players do not feel as though it is monotonous and repetitive. The other obvious component would be a compelling storyline, however as Dungeon Crawler is a procedurally-generated game, it would be difficult to implement a storyline which spans any meaningful length.

Literature Survey

The example which most immediately comes to mind for a pre-existing game similar to Dungeon Crawler is one which I drew heavily upon for ideas in terms of gameplay and level layout, although not so much mechanics: the Binding of Isaac. A top-down shooter with procedurally-generated floors containing a random assortment of different types of rooms, and delineated into different levels by boss fights every three floors, people who have played the Binding of Isaac will immediately find some similarity with Dungeon Crawler's design in terms of in-game geography.

On the other hand, the current plans I have for Dungeon Crawler's combat system will be much more similar to those of the old Pokemon games; when a player encounters an enemy or group of enemies, an "encounter" will begin in which both the player and enemy takes turns attacking or performing other combat actions. Other games with similar themes include the Legend of Zelda, Wizardry, the Bard's Tale, and more.

User Manual

The program is fairly run-of-the-mill in terms of user interface. When moving one's character through the game, one can use either the arrow keys or WASD on the keyboard to dictate the direction the character will move, and pressing E to go through doors. Most other functionalities, such as checking one's inventory and one's statuses, can be performed by clicking buttons labeled as such. The only functionality which is somewhat esoteric is the quitting command, which can be performed whenever the user presses the Escape key.

Conclusion

Overall, Dungeon Crawler is relatively simple, straight forwards exploration-adventure game with a top-down graphical system and an abstracted combat system, featuring procedurally-generated floors and rooms with a variety of different challenges.