The first phase of my PAT for 2018. I will be making a 2D, top-down rogue-like game with quests and randomly generated maps.

IT PAT Phase 1

2018

Ralph McDougall

Contents

Summary

My project will be a top-down rogue-like dungeon crawler game that includes randomly generated maps and quests. This ensures that players have a different experience every time they play. Dungeon crawler games give the player a well-defined goal to work towards which will not overwhelm the player (Patatas, 2013). No other rogue-like games could be found that have randomised quests for players to complete.

My project will be beneficial to groups of friends as each time a player plays, they will have a new experience and have new stories to share. The constant challenge of fighting swarms of enemies in a labyrinth will help to improve the player's problem-solving ability and spatial awareness as they have to try not to get trapped in dead-ends.

The player will get upgrades by picking up items. This will make the game more interesting for the player. Quests can be unlocked by talking to non-player-characters and completing the quests will give the player boosts that do not get erased when the player starts a new game.

(I will add game sounds if I have enough time)

Specification of User Interface

- A profile selection screen where the player can make a new character, use an existing character or delete a character
- A screen where the user can enter the name of their new character
- A main player menu screen where the player can choose to start a new run, view active quests, view a help screen, upgrade their character or go back to the character selection screen
- A help screen with instructions on how to move and what the goal of the game is
- A screen where the user can view their active quests
- A screen where the user can choose what stats they want to upgrade
- Display a window where the game will run
- Show the user their current health, current damage and fire rate
- Show the user an on-screen text-box with the words being spoken by the non-player characters
- Allow the player to use the keys on their keyboard to move the character and shoot projectiles

Specification for Program Function

- The program can retrieve previously used characters and delete them if necessary
- The program can create a new character
- The program can generate a random map with items and non-player characters
- The program can run a path-finding algorithm to allow all enemies to find the player
- The program can let the player pick up items that will give them boosts
- The program will allow the player to get new quests by talking to non-player characters
- The program can spawn new enemies
- The program will be able to track quest progress
- The program will be able to load a boss level for the player
- The program will allow the player to upgrade their stats
- The program will save a character's stats for when they play again later

Specifications of Help

Internal: Data validation will be in place to inform the user if they have entered invalid data for the name of a character. The user will be able to open a “Help” screen from the main menu where they can review game controls. Non-player-character monologue will inform the player as to the goal of their quest.

External: All of the help will be internal so as to make it more convenient for the user

Specifications of Data Storage

Player profiles (database):

Storing:

- Name of the character
- Character ID
- Last time character was played
- Starting player health
- Starting player movement speed
- Starting player damage
- Starting player fire-rate
- Current player level
- Player XP
- XP for upgrade
- Number of deaths
- Number of enemy kills
- Number of quests completed

Reason for database:

A database is more appropriate than a text file for storing multiple instances of the same type of data where lots of data points need to be stored. A database will allow for easier connection with the active quest systems as entries can be related to one another.

Active Quests (database):

Storing:

- Character ID
- Name of quest issuer
- Text quest description
- Quest type (kill certain enemy, find certain item, etc.)
- Quest target description
- Desired target count (10 enemies to kill, 5 items to find, etc.)
- Quest progress
- Quest reward

Reason for database:

The active quests will change all the time, values will need to change, new entries will be added and some will be removed. Thus, a database is more appropriate than a text file as a text file would not allow for easy editing of data without anomalies appearing.

Quest Format(text file):

Storing:

- Quest description structure
- Potential issuer names
- Possible quest types and values

Reason for text file:

The quest format will be a lot less structured than the entries like current characters to give more space for randomising the quests. Quest Format elements like Subject and Objective are not related so it would not be appropriate to include them in a database.

Tiles(database):

Storing:

- Name of tile
- Tile ID
- Location of tile on sprite sheet
- Whether players can walk on it or not

Reason for database:

There can be multiple types of tiles that need to be stored and they are structured in the same way so a database is appropriate.

Enemy Type(database):

Storing:

- Name of enemy
- Location of enemy on sprite sheet
- Enemy health
- Enemy movement speed
- Enemy damage
- Enemy damage type (contact or ranged attacks)
- Enemy damage rate

Reason for database:

There can be multiple types of enemies and they all have the same structure so a database is more appropriate.

Sprites (png file):

Storing:

- Tile sprites
- Character sprites
- Projectile sprites
- Particle sprites

Reason for png file:

Databases and normal text files normally do not store images and png files are standard for storing images.

Map Sections (text file):

Storing:

- Descriptions of map segments that get stitched together to make a more continuous map

Reason for text file:

The map sections will be quite large arrays of data with varying size so a database would not be appropriate.

Hardware and Software Specifications

Development Specifications:

Hardware: Standard peripherals like a mouse, keyboard and monitor will be required. 128MB RAM, 128MB Disk Space and a 266MHz Pentium Processor are the minimum requirements for Java on Windows. (Oracle, n.d.) However, the recommended requirements for NetBeans are 2GB RAM, 1.5GB disk space and an i5 or equivalent processor. (NetBeans, 2015)

Software: An operating system such as Windows 7/8/10 will be necessary. The Java Runtime Environment and Java Development Kit will also be required. NetBeans 8.1 will be used to develop the program with MS Access, Notepad++ and Paint.NET being used for the databases, text files and images respectively. The Slick2D graphics library will be used for the graphics and for real-time user input. ucanaccess will be used for connection to the database.

Runtime Specifications:

Hardware: 4 GB of RAM should be more than sufficient to execute the program. 128MB of free space on the hard disk should be more than enough for the user. A minimum of 2GHz for the CPU should be more than sufficient. A bottom of the line graphics card should be sufficient for the user to enjoy the game. Standard peripherals like a screen, mouse and keyboard will be required.

Software: The user will require the Java Runtime Environment for Java 8 to run the program. They will not require any IDE such as Netbeans nor will they require the Java Development Kit (Oracle, n.d.). The libraries for the Slick2D graphics API and ucanaccess for database connection will come with the project (Anon., 2014) (Anon., n.d.). An operating system such as Windows 7/8/10 is required.

Bibliography

Anon., 2014. *Getting Started.* [Online]
Available at: http://slick.ninjacave.com/wiki/index.php?title=Getting_Started
[Accessed 28 2 2018].

Anon., n.d. *UCanAccess.* [Online]
Available at: ucanaccess.sourceforge.net/site.html
[Accessed 28 2 2018].

NetBeans, 2015. *NetBeans IDE 8.1 Release Notes.* [Online]
Available at: https://netbeans.org/community/releases/81/relnotes.html
[Accessed 12 3 2018].

Oracle, n.d. *Oracle.* [Online]
Available at: http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html
[Accessed 28 2 2018].

Oracle, n.d. *What are the System requirements for Java?.* [Online]
Available at: https://java.com/en/download/help/sysreq.xml
[Accessed 12 3 2018].

Patatas, B., 2013. *The Dungeon Crawler Recipe.* [Online]
Available at:
https://www.gamasutra.com/blogs/BrunoPatatas/20130119/185096/The_Dungeon_Crawler_Recipe.php
[Accessed 24 2 2018].