**Eric Faust**

**Benjamin Peake**

Imgd 3000 Project 3 Plan

**Game Name:** Roguelike Trail  
**Genre:** RPG/Roguelike  
**Game Description**: The player plays the lives of four adventurers on a quest to find a fabled treasure! They must brave the wilds to reach the ancient dungeon, and brave its depths, battling monsters and traps.

**Technical Features**: ASCII RPG with realtime lighting and items to buy and quests to complete.  
**Artistic Assets:** ASCII art of various monsters or scenes, such as a goblin, minotaur, large chasm, or trapped hallways

Our game is a side scrolling dungeon crawl adventure. The player would be shown an animation of them walking through the dungeon until an event happens, similar to Oregon trail. Monsters or other events appear, and the player would fight monsters in a turn based combat. The player will get treasure for defeating monsters and succeeding at events. The game ends. The two end conditions for the game are if the players party of adventurers all die, or they reach the end of the dungeon. The player starts with a party of 4 adventurers.

Classes that will be implemented:

**Creature** - This is a class for any living thing, such as the player’s adventurers, and enemy monsters. It has a health and attack.  
**Adventurer** - This class is derived from creature class, and holds health, attack and mana. If the health goes to zero, the character dies. An adventurer can either attack physically (using attack stat) or with magic (mana stat).

**Monster** - This class is derived from creature. It has health and attack.

**Character manager** - This singleton manager handles the player’s adventurers, so they can be easily retrieved and modified. On startup it generates the party of adventurers. Also keeps track of how much treasure has been accumulated

**Scene** - This contains a list of objects located in the scene. The game has a stack of scenes that can be shown when the player travels to a new part of the dungeon. It also contains the UI for the scene.

**Battle** - This class derives from the scene. It contains a monster, and extra functions for easily getting data from the battle (health, attack, treasure).  
**Dungeon Manager** - This singleton manager handles what part of the game the player is in, such as when the next town/merchant is coming, or what quest the player is on.  
**Safezone** - This class is derived from scene. It also holds information on what the player could buy here (potions, weapons, paintings of kittens), and what quests are available. An example safezone could be a town with a quest to kill a minotaur (accepting this quest opens a battle with a minotaur), or a merchant who has many items for sale.

**StatView** - This view object shows information on a monster or adventurer (attack, health, mana).

**Math** - This class holds helpful math functions, such as an easy to use random number generator

**Implementation:**  
The player has their party generated, and the game starts. The dungeon manager decides the layout of the next scenes, such as which battles are coming, and where the next safezone is.

This will create a linked list of scenes, each leading to the next. The players go through these in order (not knowing what the next scene is), and are only interrupted by optional quest battles given by safezones.

If the player successfully gets through 10 scenes, they get to a final boss battle, and if they succeed, they win the game. Their high score is their remaining treasure, so the less they need to buy, the higher a score they get.

**Distribution of Work:  
Eric Benny Shared**  
  
**Scene  
 Battle  
 Safezone  
Creature  
 Adventurer**

**Monster  
Character Manager  
Dungeon Manager  
StatView  
Math**  
**ASCII monsters**

**Schedule:**

Oct 9 - Alpha - All the classes required for encountering random battles. This includes Creature, Adventuter, Monster, Scene, Battle, Dungeon Manager

Oct 9 to 14 - Implement safezones, quests and merchants, final battle, and victory conditions. Implement as many ASCII monsters as possible.  
Oct 14 - Final playable!  
Oct 14 to 16 - Sleep  
Oct 16 - Presentation