Project 5 – Zork

**Problem**

Create a dungeon game that lets a player move through a dungeon and slay monsters.

**List of Inputs, Outputs, and Processing Required**

INPUTS

* East and west, along with a player name, are the only inputs accepted from the program.

OUTPUTS

* The program runs a game that repeatedly asks for a direction.
* The program will print the current room that the player is in to simulate a world/movement.

PROCESSING

* Random numbers create the probability of monster spawning and weapon drops.
* Fight sequences have a random chance to miss.
* The program must check for direction when the player tries to move.
* The program must check if the player is alive and a monster before doing battle.

**Identification of Classes and Their Responsibilities**

Class name: Participant

Responsibilities

* Serves as a superclass for Player and Monster.

Class name: Player

Responsibilities

* Represents the user character. Holds health, attack, and a name value and can hold a weapon

Class name: Monster

Responsibilities

* Serves as a superclass for multiple monster types in the game

Class name: Weapon

Responsibilities

* Serves as a superclass for multiple weapon types in the game

Class name: DungeonRoom

Responsibilities: Serves as a superclass for a start room and all other dungeon rooms

Class name : DungeonCell

Responsibilities: A subclass of DungeonRoom, it serves as a regular room with two exits that can hold monsters and weapons.

Class name: StartCell

Responsibilities: A subclass of DungeonRoom, it makes it impossible for monsters or weapons to spawn in the first room of the game

Class name: SteelDraconid

Responsibilities: A derived monster type from the superclass Monster

Class name: FrigidPhantom

Responsibilities: A derived monster type from the superclass Monster

Class name: RockTitan

Responsibilities: A derived monster type from the superclass Monster

Class name: RustyKama

Responsibilities: A derived weapon type from the superclass Weapon

Class name: MagicGauntlet

Responsibilities: A derived weapon type from the superclass Weapon

Class name: BlisteringMace

Responsibilities: A derived weapon type from the superclass Weapon

Class name: PossessedGreatsword

Responsibilities: A derived weapon type from the superclass Weapon

Class name: GameManager

Responsibilities: Contains methods and startup values that help move the game forward and allow the driver to interact with the game itself.

Class name: ZorkDriver

Responsibilities: The driver class for the project, which puts our code to use and lets us play the game.

**UML Class Diagram**

**\*See attached documents, final Diagram was considerably large so a screenshot would not be very kind to fitting into this document.**

**Algorithms**

**\*These algorithm examples are shortened to avoid cluttering the document\***

Class: ZorkDriver, fight() method

START

Player moves east = monster spawn;

Fight, player atk -= monster health

Monster health = ;

Fight, monster atk -= player health

Player health = ;

WHEN player or monster dead =

END

**Test Cases**

|  |  |
| --- | --- |
| **Go east or west? (With west false)** | **Results** |
| User says “east” | **Playerposition +1,** game continues into room |
| User says “west” | **Sorry, I can’t go that way.** |

Since for this game, StartCell is the only room with west set to false, we only need to test its fail statement in that room (since it won’t occur elsewhere). Running through the whole program, the game will only accept east or west as a given input value (excluding when the game will ask for the player name), and the game will progress naturally to the east with no issues. Even moving back to previous rooms correctly moves the player throughout the game.