

---

SOFTWARE  
REQUIREMENTS  
SPECIFICATION

for

Fingal's Cave

Version 1.4

Prepared by: Matthew Fricke

Submitted to: CS151 Students May 6,

2020

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Purpose</b>	<b>2</b>
2.1	Intended Audience . . . . .	2
2.2	Project Scope . . . . .	2
2.3	Part 1: Create Player and Non-Player Characters . . . . .	2
2.3.1	Part 2: Create Game Map . . . . .	3
2.3.2	Part 3: Game Logic . . . . .	3
<b>3</b>	<b>Program Design</b>	<b>3</b>
3.1	Dice Rolling Function . . . . .	4
3.1.1	Classes . . . . .	4
3.2	Lists . . . . .	6
3.3	Setup . . . . .	6
3.4	Game Logic . . . . .	7
3.5	Game End Conditions . . . . .	7
<b>4</b>	<b>User Stories</b>	<b>7</b>

## 1 Introduction

## 2 Purpose

Fingal's cave is a thrilling text based adventure game! This software specification defines the game logic, structure, play-throughs (user stories), and defines tests (unit tests).

### 2.1 Intended Audience

This specification is for software developers and testers.

### 2.2 Project Scope

Fingal's Cave is a game written by students enrolled in the University of New Mexico Intro to Computer Science course. Development of the game is divided into three parts each with their own due dates.

### 2.3 Part 1: Create Player and Non-Player Characters

In the first phase of development programmers will implement and test the character and weapon classes as well as all classes that inherit from them. That is, weapon, fist, rock, club, sword, character, player, mcguffin, troll, and imp.

The program will be submitted for grading using Zybooks.

The code for this phase is due on April 26th, 2020 before midnight.

Write a dice rolling function that generates random numbers used throughout the game.

Create a troll character with one of the names from the troll name list, give the troll 10 health.

Create an imp character with one of the names from the imp name list, give the troll 5 health.

Create a player character, give the player 40 health and a club.  
Create a McGuffin character.

### 2.3.1 Part 2: Create Game Map

In the second phase the developers will implement the game map that represents the game locations and their contents.

- Implement the location class as described in Section 3.1.1.
- Follow the instructions in Section 3.3 to populate your map.

The code for this phase is due on May 3th, 2020 before midnight.

### 2.3.2 Part 3: Game Logic

In this phase the developers write code that implement the player/monster combat, and victory/loss conditions.

The code for this phase is due on May 10th, 2020 before midnight.

## 3 Program Design

In this text based adventure game the player enters the famed Fingal's Cave in search of Billy MuGuffin a child last seen before his disappearance near the entrance to the cave. The player must avoid or defeat the monsters of the cave, locate Billy and rescue him.

## 3.1 Dice Rolling Function

The dice rolling function takes two arguments: the first argument is the number of sides on the dice and the second argument is the number of dice. The function returns the sum of the random dice rolls. For example if I call `rollDice(6,2)` it might return 7, which is the sum of randomly chosen numbers 4 and 3.

### 3.1.1 Classes

In all cases classes should support instantiation using the *default* constructor. In object-oriented programming the default constructor takes no arguments. It may be useful to have constructors that take arguments, but classes should still have a default constructor.

- Location
  - Character list: characters. A list of characters at this location.
  - Weapon: weapon. A weapon at this location if any.
  - string: description. A description of the location.
  - method: `addCharacter(Character)`: Adds the character passed as an argument to this location.
  - method: `removeCharacter(Character)`: deletes the character passed as an argument from this location.
  - method: `setWeapon(Weapon)`: places a weapon at this location.
  - method: `getWeapon()`: returns the `Weapon` at this location or `None`.
  - method: `removeWeapon()`: sets the weapon at this location to `None`.
  - method: `getCharacters()`: returns the list of characters at this location.
  - method: `print()`: prints the description of the location, the list of characters including their names and types, and the weapon at this location, if any.
- Weapon:
  - string: description
- Fist inherit from weapon
  - method: `damage()` returns integer. Draw integer from 1 four sided dice.
- Club inherit from weapon
  - method: `damage()` returns integer. Draw integer from sum of 3 six sided dice.
- Rock inherit from weapon

- method: `damage()` returns integer. Draw integer from sum of 2 four sided dice.
- Sword inherit from weapon
  - method: `damage()` returns integer. Draw integer from sum of 4 eight sided dice.
- Character
  - string: `name`.
  - integer: `health`.
  - method: `setHealth( integer )`
  - method: `getHealth()` returns integer
  - method: `setName( string )`
  - method: `getName()` returns string
- Player: inherits from Character
  - Weapon: `carried_weapon`
  - method: `moveNorth()`. Moves to a location to the “north” of the players current location. Returns False if moving north would put the player off the edge of the map.
  - method: `moveSouth()`.
  - method: `moveWest()`.
  - method: `moveEast()`.
  - method: `hit( character )`: reduces the health of the character passed in as an argument. The amount reduced is given by calling the `damage()` method belonging to the `carried_weapon` object.
  - method: `take( Weapon )`
  - method: `rescue( Character )` returns True if the Character is of type McGuffin and False otherwise.
  - method: `getCurrentCoordinates()`: returns the row and column occupied by the player in the Map matrix.
  - method: `getCurrentLocation()`: return the location object that contains the player.
- Troll: inherits from Character
  - method: `hit( character )`. Reduce character health by the sum of 2 six sided dice.
- Imp: inherits from Character
  - method: `hit( character )`. Reduce character health by the sum of 1 six sided dice.
- McGuffin: inherits from Character

## 3.2 Lists

Troll names: Basalt, Granite, Obsidian, Tom, Bert, William, Adelaide, Alice, Betty

Imp names: Biorite, Gabbro, Pegmatite, Peridotite, Alacranite, Dacite, Diorite

Location descriptions:

- This area is dimly lit with mossy walls.
- A dark area with a slimy floor.
- A hole in the roof lets the sunlight in illuminating the area.
- The floor here is sandy.
- The walls are covered with glowing fungi.
- A burned out fire-pit is at the centre of this area.
- An area with stalagmites and stalactites.
- The floor is streaked with multicoloured marble.

## 3.3 Setup

1. Create a matrix called Map with 4 rows and 4 columns. Each element of the matrix is a location object.
2. Set each location's description randomly from the description list.
3. Create 5 trolls and add them to locations objects at random positions in the Map matrix. Give each troll a random name from the names list. Set the trolls health to 10.
4. Create 3 imps and place them on the Map just as was done with the trolls. Initialise health to 5.
5. Create McGuffin object and place it on the Map.
6. Create player object and initialise with 40 health. Initialise carried\_ - weapon to *fist*. Place the player at a random location on the Map.
7. Create 3 rocks and place them randomly on the Map. Set the rock description to: "a rough looking rock".
8. Create 3 clubs and place them randomly on the Map. Set the club description to: "a heavy club".
9. Create 2 swords and place them randomly on the Map. Set the sword description to: "a bright sword".

### 3.4 Game Logic

The game is governed by an output/input loop:

1. Print location description
2. Print list of characters at the location
3. Print list of weapons at the location
4. Print available directions to move
5. Action loop:
  - (a) For each monster in the character list for this location perform the following:
    - i. Monster hits player
    - ii. Print player's current health
    - iii. if player's health is less than zero call `GameLost()` function.
  - (b) Prompt user to perform an action (move NWSE, take weapon, hit creature, or rescue Billy).
  - (c) Read action from player
  - (d) Perform the action the user specified
  - (e) Remove any monsters with less than zero health from the location.
  - (f) if the action is `rescue(McGuffin)` call `GameWon()` function.
  - (g) If the action is *move* then go to top of the input/output loop for new location, else to repeat the action loop.

### 3.5 Game End Conditions

Define function `GameWon` which when called prints: "Congratulations! You found Billy McGuffin and led him out of Fingal's Cave! Thanks for playing." and ends the program.

Define function `GameLost` which when called prints: "Game Over! You were defeated by Fingal's Cave! Thanks for playing." and ends the program.

## 4 User Stories

These are examples of game play for the final program. Output for Parts 1 and 2 will be different. Program output is in **bold**. Player input is in *italics*.

User Story 1:

A CS151 Text Adventure Game

---

---

Coordinates: (3, 2)

---

You see:  
A dark area with a slimy floor.  
A troll named William

Enter an action (E, W, N, S, hit, rescue, take, hint or quit):  
The dice roll [3, 2]  
William hits you for -5  
You have 35 health left.  
The dice roll [4]  
hits you for -4  
You have 31 health left.  
You could not go north

---

---

Coordinates: (3, 2)

---

You see:  
A dark area with a slimy floor.  
A troll named William

Enter an action (E, W, N, S, hit, rescue, take, hint or quit):  
The dice roll [1, 2]  
William hits you for -3  
You have 28 health left.  
The dice roll [4]  
hits you for -4  
You have 24 health left.  
You go south

---

---

Coordinates: (2, 2)

---

You see:  
The walls are covered with glowing fungi.  
A heavy club

Enter an action (E, W, N, S, hit, rescue, take, hint or quit):  
The dice roll [4]  
hits you for -4  
You have 20 health left.

---

---

Coordinates: (2, 2)

---



You see:  
The walls are covered with glowing fungi.

Enter an action (E, W, N, S, hit, rescue, take, hint or quit): The dice roll  
[1, 4, 2]  
hits you for -7  
You have 13 health left.  
You go west

---

---

Coordinates: (2, 1)

---

You see:  
This area is dimly lit with mossy walls.  
An imp named Dacite  
A bright sword

Enter an action (E, W, N, S, hit, rescue, take, hint or quit):  
The dice roll [1]  
Dacite hits you for -1  
You have 12 health left.  
The dice roll [5, 6, 4]  
hits you for -15  
You have -3 health left.

Game Over! You were defeated by Fingal's Cave! Better luck next time.  
Thanks for playing.

User story 2:

A CS151 Text Adventure Game

---

---

Coordinates: (3, 2)

---

You see:  
A dark area with a slimy floor.  
A troll named William

Enter an action (E, W, N, S, hit, rescue, take, hint or quit):  
The dice roll [3, 2]  
William hits you for -5  
You have 35 health left.  
You could not go north

---

---

Coordinates: (3, 2)

---

You see:

A dark area with a slimy floor.  
A troll named William

Enter an action (E, W, N, S, hit, rescue, take, hint or quit):  
The dice roll [6, 4]  
William hits you for -10  
You have 25 health left.  
You go south

---

---

Coordinates: (2, 2)

---

You see:  
The walls are covered with glowing fungi.  
A heavy club

Enter an action (E, W, N, S, hit, rescue, take, hint or quit):

---

---

Coordinates: (2, 2)

---

You see:  
The walls are covered with glowing fungi.

Enter an action (E, W, N, S, hit, rescue, take, hint or quit): You go west

---

---

Coordinates: (2, 1)

---

You see:  
This area is dimly lit with mossy walls.  
An imp named Dacite  
A bright sword

Enter an action (E, W, N, S, hit, rescue, take, hint or quit):  
The dice roll [1]  
Dacite hits you for -1  
You have 24 health left.

---

---

Coordinates: (2, 1)

---

You see:  
This area is dimly lit with mossy walls.  
An imp named Dacite

Enter an action (E, W, N, S, hit, rescue, take, hint or quit):  
The dice roll [2]  
Dacite hits you for -2  
You have 22 health left.

The dice roll [7, 7, 2, 8]  
You hit Dacite with A bright sword for -24  
Dacite has -19 health left.  
Dacite disappeared!

---

---

Coordinates: (2, 1)

---

You see:  
This area is dimly lit with mossy walls.

Enter an action (E, W, N, S, hit, rescue, take, hint or quit): You go north

---

---

Coordinates: (3, 1)

---

You see:  
An area with stalagmites and stalactites.

Enter an action (E, W, N, S, hit, rescue, take, hint or quit): You go east

---

---

Coordinates: (3, 2)

---

You see:  
A dark area with a slimy floor.  
A troll named William

Enter an action (E, W, N, S, hit, rescue, take, hint or quit):  
The dice roll [2, 1]  
William hits you for -3  
You have 19 health left.  
You go south

---

---

Coordinates: (2, 2)

---

You see:  
The walls are covered with glowing fungi.

Enter an action (E, W, N, S, hit, rescue, take, hint or quit): You go west

---

---

Coordinates: (2, 1)

---

You see:  
This area is dimly lit with mossy walls.

Enter an action (E, W, N, S, hit, rescue, take, hint or quit): You go north

---

---

Coordinates: (3, 1)

---

You see:  
An area with stalagmites and stalactites.

Enter an action (E, W, N, S, hit, rescue, take, hint or quit): You go east

---

Coordinates: (3, 2)

---

You see:  
A dark area with a slimy floor.  
A troll named William

Enter an action (E, W, N, S, hit, rescue, take, hint or quit):  
The dice roll [5, 6]  
William hits you for -11  
You have 8 health left.  
The dice roll [8, 8, 7, 2]  
You hit William with A bright sword for -25  
William has -15 health left.  
William disappeared!

---

Coordinates: (3, 2)

---

You see:  
A dark area with a slimy floor.

Enter an action (E, W, N, S, hit, rescue, take, hint or quit): You go east

---

Coordinates: (3, 3)

---

You see:  
The floor here is sandy.  
Billy McGuffin!  
A bright sword

Enter an action (E, W, N, S, hit, rescue, take, hint or quit): \_\_



Congratulations! You found Billy McGuffin and led him out of Fingals Cave!  
Thanks for playing.