# Team Project Proposal
# CS320-103: Software Engineering, Spring Semester 2020
# Due Date: Monday, 2-3-2020, by 7:00am

## The Greatest Text Adventure *Ever Made*

**Team Members:**

1) **Kai Barclay**
2) **Ralph Greaves**
3) **Brooke Tingley**
4) **Will Wyatt**

**Summary:**

*You blink your eyes, trying to take in your low light environment. You had been above ground, walking along a forest path, minding your own business. By the ache in your body, you probably fell down here—*but where is here? *You look up to see the hole you fell through, currently your only light source, too high to get to again. The village elders had always said the forest hid a great evil, that it was to be avoided, but you never believed them. You might as well believe them now, if you can get out of here. What do you do first?*

Computers have offered many avenues of advancement both in science and in creativity. One such advancement came in the world of games, with the advent of the text-based adventure games. As early as the 1960s, users would input commands through teleprinters, and the computer would return the response. The medium has come far since then, and we plan to make the most of it in this project.

The project will be hosted on a server; users will interact through a webpage. There will be a class structure supporting the main functions of the game; the exact details and interactions between these classes will be developed as the game is fleshed out. However, several base classes are a must: Map (with Room subclasses), Item, Inventory, Command, and Interface.

**Features:**

Interface: The primary interface for the project will show the prompt and descriptive text, while also including a text box for the user to input their action. The action will then appear in the output text before the next descriptive prompt is produced.

Map: This is the map of rooms that the player will be traversing in the game. The way the rooms are connected and the paths the player can take are all included in the map. The map will be planned out thoroughly and completely before implementation.

Room: The rooms serve as individual events in the adventure. Each room will be connected to other rooms and have specific paths for the player to follow. The room may also have objects to interact with and/or items to add to the inventory.

Item: Objects can be picked up and used by the player throughout the game. Different items will interact with obstacles in different ways, though some may not do anything at all (A key will not do anything to a tree, for example.). We will have to plan out the various item functions as development progresses.

Inventory: The player will have an inventory of the items that they collect throughout the game; the player can check their inventory at any time during the game. Any item that is in the user inventory can be used at any time, and in any room. An item that is added to the inventory leaves the room, with the user, and cannot be picked up from that room again during the game, unless the player drops said item.

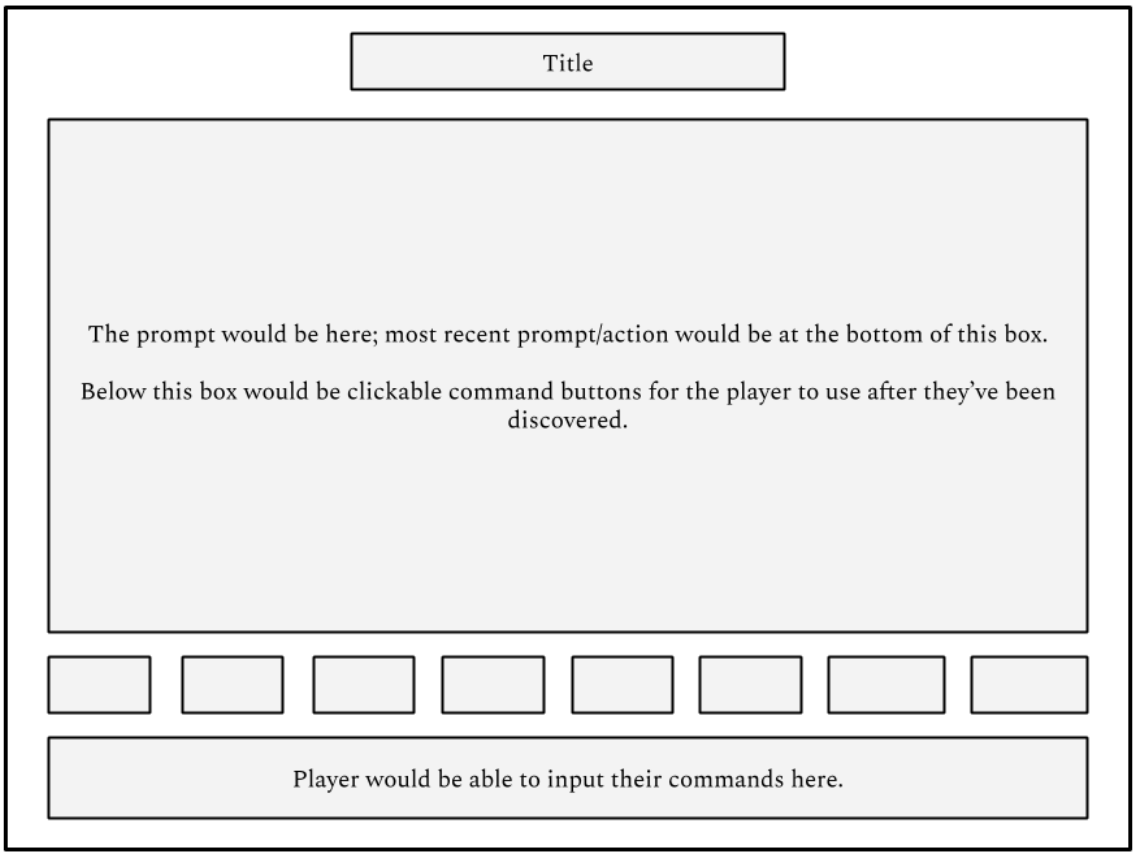
Shortcut buttons: As commands are discovered by the player, shortcut buttons will appear over the input box. The player can then simply click these buttons in order to use the command, rather than typing out the command every time she chooses to use it.

Combat: *This is a stretch goal.* The player may run into enemies throughout the map and engage in combat with them. The player and the enemies will have their own independent health parameters and stats.

Multiplayer: *This is a stretch goal.* The game can be designed for multiple players to play at the same time, and on the same map. This is an especially challenging task; we would have to plan what sort of interactions players can have with each other, as well as making sure that if a player picks up an item in a room, nobody else can take that item.

**Sketches:**

UI:




**Responsibilities:**

Kai: Repository management; class design, implementation

Ralph: UI/UX; ASCII graphics

Brooke: Class design; map, room design

Will: Class design; JSPs; unit tests

**Challenges:**

This is a complex project that requires much planning. We can't begin until we've plotted out the map, come up with puzzles, and have a base setting to work around. We also need to decide what sort of items the player can add to their inventory and how those items can interact with other objects. Down the line, we can begin the implementation of the more complex stretch goals. The multiplayer mode, of course, could prove to be particularly challenging.

**Development Environment:**

Java; Eclipse; CSS, HTML