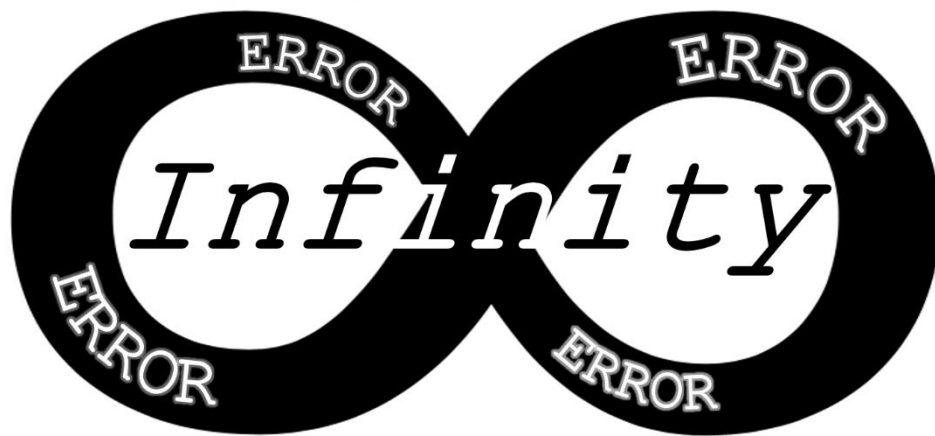


# Escape the Hotel



## Team Infinity Final Report

By: Jackson Druhan, Jacob Lachowski, Muhammad Saleh, & James Smith

Dec 2, 2020

# Table of Contents

• Introduction-----	3
• Project Management-----	3
○ Team Organization-----	3
○ Risk Management-----	3
• Development Process-----	4
• Software Design-----	5
○ Design Rationale-----	5
• Appendix-----	6

# Introduction

This project will be a text-based adventure game that involves escaping from a hotel. This game will be controlled and displayed through the terminal window. This game will include environments and spaces where each space will contain objects and NPC's that must be interacted with to solve puzzles that will allow the player to progress and win. The plan for management, development, and design of the program will be outlined below.

## Project Management

Management of this project will include the organization of the team (roles and responsibilities), as well as risk management.

- Team Roles

Team Member	Design	Implementation	Testing Maintenance
Jackson	Phase Lead	QA Lead	N/A
James	QA Lead	Assurance Manager	N/A
Jacob	Assurance Manager	Librarian	N/A
Muhammad	Librarian	Phase Lead	N/A

Assurance Manager is responsible for making sure that everyone is on task, checking for quality, standards and specifications of the project. (Code, documentation, etc.)

- Risk Management

- Underestimated the workload and the project is too large.

Starting with a smaller and simpler project that can be expanded on later will allow us to manage the workload and change the size of the game as we go through it.

- A team-member is unable to participate

The size of the project will be changed to accommodate the lack of a team member (three puzzles could be lowered to two).

- Team members are unfamiliar or inexperienced with the software/tools.

Team members will bring this up in scheduled weekly meetings and another member can assist/ train user with tool. Alternatively, the team member will use resources and get help to be able to use those tools.

- Microsoft Teams (application) stops working properly and communication is affected.

Alternate forms of communication are established so that if Teams fails, we can still communicate and share information (i.e., Discord).

## Development Process

- Code Review Process

- All changes to main branch or branches created by another member will be communicated and checked before merging or pushing. Tasks will be distributed such that each member will have separate sections of code to work on so that merge conflicts don't occur as often.

- Communications

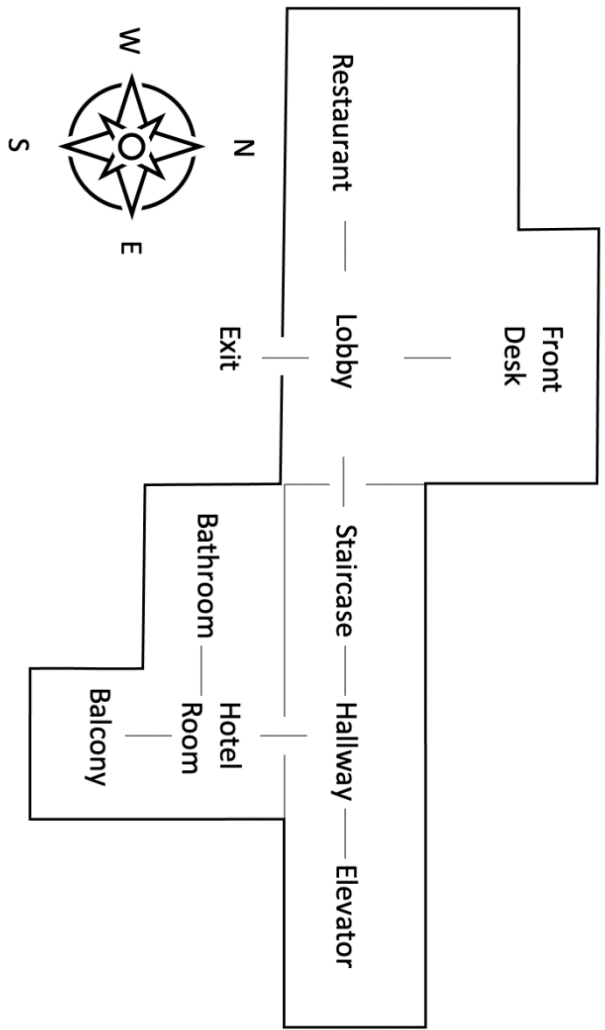
- Team members will primarily use Microsoft Teams as a means of communication and file sharing, besides Gitlab which will be used for pushing, pulling, merge requests and issues. Weekly meetings have been set and best times for additional meetings have been located.

- Change Management
  - Bugs will be handled by the team member responsible for the section that the bug was found in. Bugs can either be communicated to the other team members in teams and they will be added to the issues section of the repository on Gitlab.

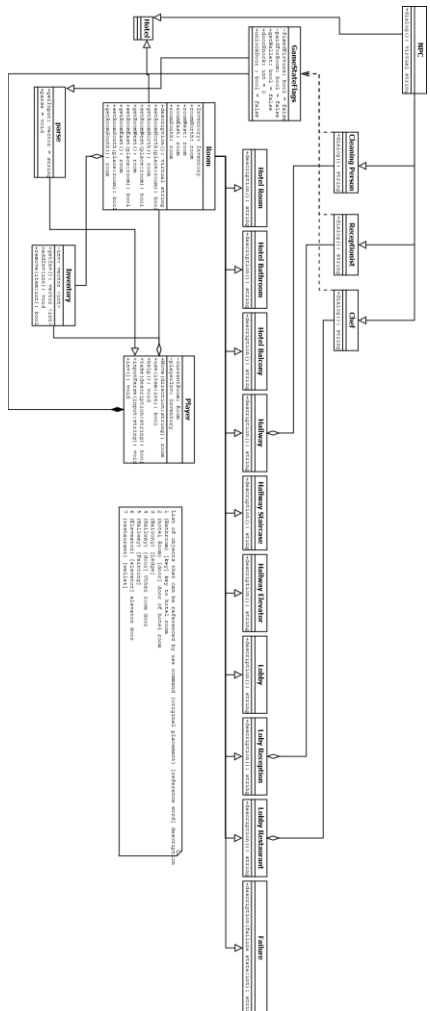
## Software Design

- Design Rationale
  - Game will be split up into three environments, the hotel room, the hallway, and the lobby each with three separate spaces within them. The room will include the players hotel room, the adjoining bathroom, and the balcony. The hallway will include the main hallway, the staircase, and the elevator. The lobby will include the main lobby area, the reception desk, and the lobby restaurant. Each environment is separated by a puzzle that requires the use of objects that can be found in each room and interactions with NPCs that can be found throughout. Progressing and winning will rely on where the player goes, what items they have collected and used, and what NPCs have been interacted with.
- Design
  - See Appendix.

# Appendix

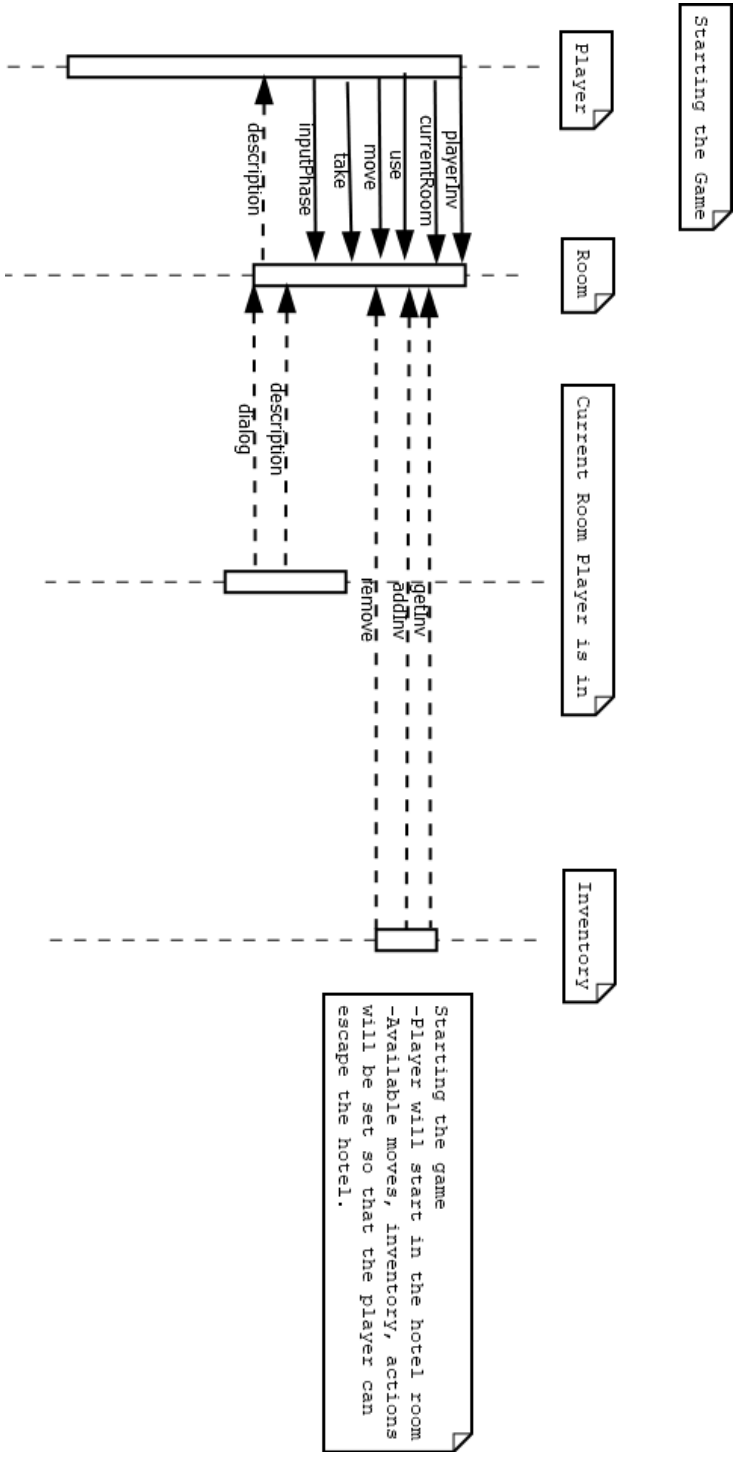


**Figure1:** The proposed layout of the hotel separated into its environments.

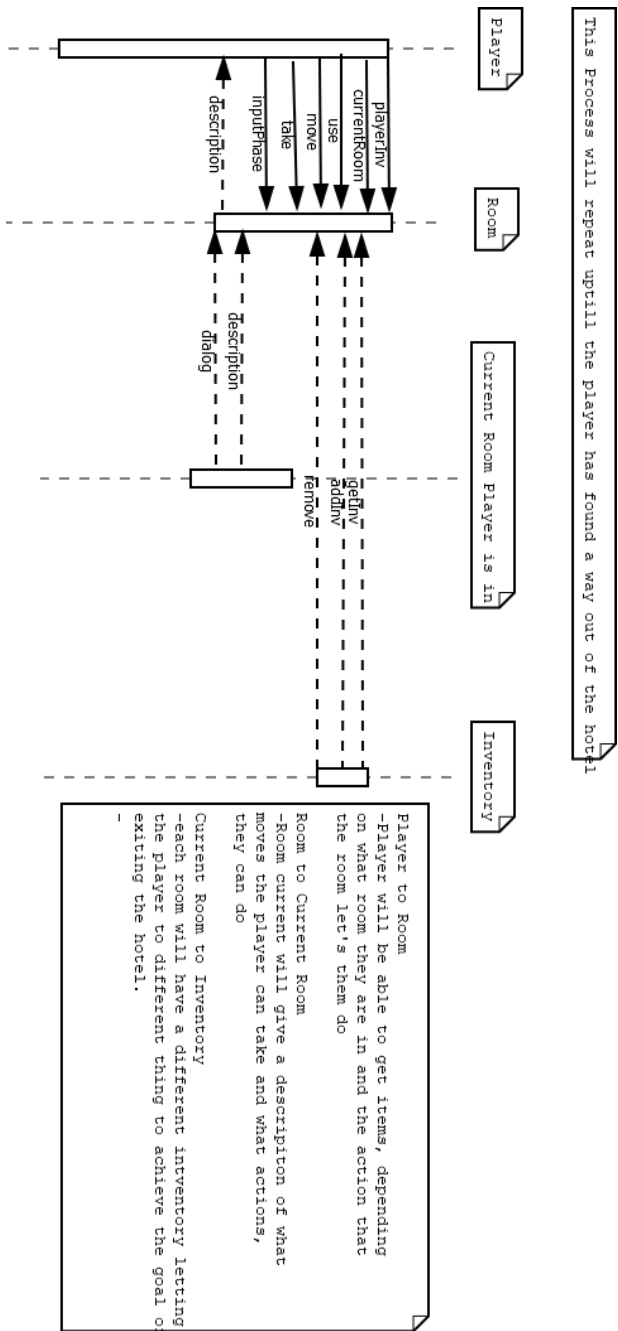




**Figure2:** A UML diagram showing the proposed layout of the classes, their inheritances and dependencies



**Figure3:** A State diagram for starting the game.



**Figure4:** A State diagram for an individual turn.