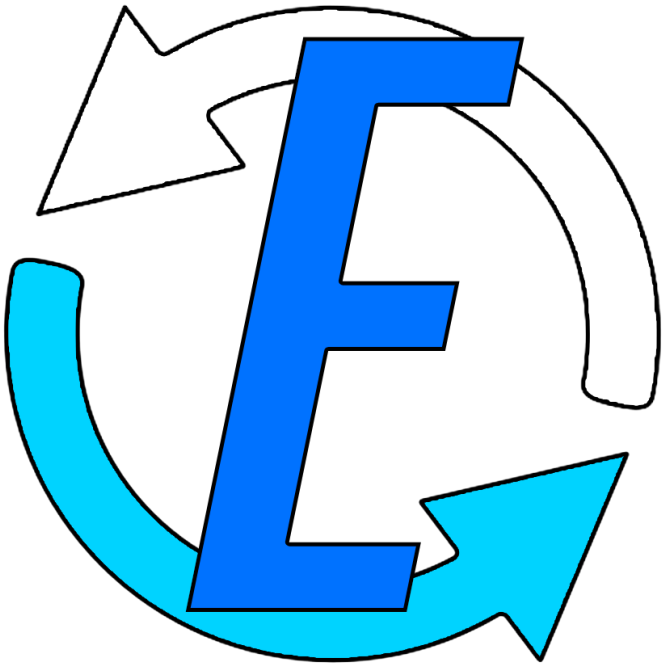
SeaSick

Project Document



Team E: *Electra*

Dustin Ward

Jared Sisco

Neville Leung

Due Date: October 20th, 2019

Table of Contents

Introduction

Welcome to the project document for the horror-thriller text-based adventure game SeaSick, created by Team Electra.

SeaSick is a command line experience set on a cargo ship in the middle of the Atlantic ocean. You play as the captain that has finally regained consciousness after an incident involving an intense storm. The storm appears to have damaged your ship, and the crew has mysteriously vanished. After some investigation you learn that pirates have taken advantage of the situation and have boarded your ship to steal your valuable cargo. Your objective is to stop the pirates, and save the crew that they have held captive.

The player will use a command line interface to progress through a series of rooms towards their objective. Along the way, the player will encounter a variety of items and NPC's that serve to track progress through challenges and puzzles. Some NPC'c will be crew members that require a certain item to progress, and others will be pirates that engage in combat with the player. There is also a temperature system that will challenge the player to find items and rooms to keep warm. If the player fails to stay warm, they will lose the game. If a combat scenario is lost at any point, it will also cause the player to lose. The game is won by successfully completing all the required steps involved in sending a radio message for help.

In this document, you will find the following sections:

A Project Management section that details the organization and responsibilities of all the team members along with a list of foreseeable risks and issues the team may face, and how we plan on dealing with them.

A Development Process section that details the coding processes and practices that the team will follow, along with the teams methods of communication and issue tracking. This section also contains the necessary design diagrams for the game.

Project Management

Team Organization

- Dustin Ward (Team Lead)
  - Responsible for ensuring that code from other members is successfully integrated into the project.
  - Approving pull requests and resolving merge larger merge conflicts.
  - Assigning bugs to developers for resolution.
  - General team management responsibilities, like keeping track of other developers progress and assigning priority to certain tasks.
  - Leading creation of proper team and design documentation.
  - Contributing to smaller/easier classes and implementations, along with the related unit tests.
  - Contributing to design changes and ideas.

- Jared Sisco, Neville Leung
  - Responsible for the bulk of class development.
  - Individually working on the larger/harder classes and implementations.
  - Creating proper and rigorous unit testing procedures that ensure the code is bug-free.
  - Fixing bugs that have been they have been assigned.
  - Contributing to proper team and design documentation
  - Contributing to design changes and ideas.

Risk Management

- **The team planned a project that is too large:**
  - ➢ The team will cut secondary/unnecessary ideas to save on time in case the project becomes too large to handle.
  - ➢ i.e. Ascii art, extra rooms and characters, extra in-game items and events, etc.

- **The team underestimated how long parts of the project would take:**
  - ➢ Members will be temporarily reassigned tasks if one is taking too long.
  - ➢ Team lead will move first to assist one of the class developers with implementation, then the other developer will move too if there is a serious problem.

- **Changes to design are needed during implementation:**
  - ➢ Initial discussion will begin in one of the determined communication forums. (Discord or Gitlab)
  - ➢ If there is a serious flaw in the design, then the team can call a meeting to discuss any modifications to the design.
  - ➢ The team lead will then update UML diagrams to be consistent with the new design, and identify the required changes that need to be made by each developer.

- **Addition or loss of team member:**
  - ➢ If one of the class developers suddenly leaves the group, the lead will take over their responsibilities, and begin to implement the larger and more difficult portions.
  - ➢ If the lead were to suddenly leave the group, one of the class developers will step up and absorb the responsibilities after a discussion between them.
  - ➢ If a new member joins the team, they will take on some of the secondary tasks from other members (Testing/Documenting) or they could begin to work on secondary features that had been previously cut from the project.

- **Unproductive team member(s):**
  - ➢ Bring it to the attention of the team lead.

- ➢ If the issue cannot be resolved between the lead and the problem member, then it will be elevated and taken to Dr. Anvik.

- **Team member(s) lacking expected technical background:**
  - ➢ It is expected that if a team member does not have the required technical information, they will put in the time to learn necessary skills or ask another member for resources that will help.
  - ➢ If there isn't enough time for the member to learn specific skills, then the task can be given to another member of the team in exchange for a task the member will be able to complete.

- **Major life events:**
  - ➢ Will be treated in a similar fashion to losing the team member.
  - ➢ If the member can still handle some workload, then it will be up to them to decide how much they will contribute.

- **Inexperience with new tools:**
  - ➢ Treated in a similar fashion to a member lacking expected technical background.

Development Process

Procedures And Practices

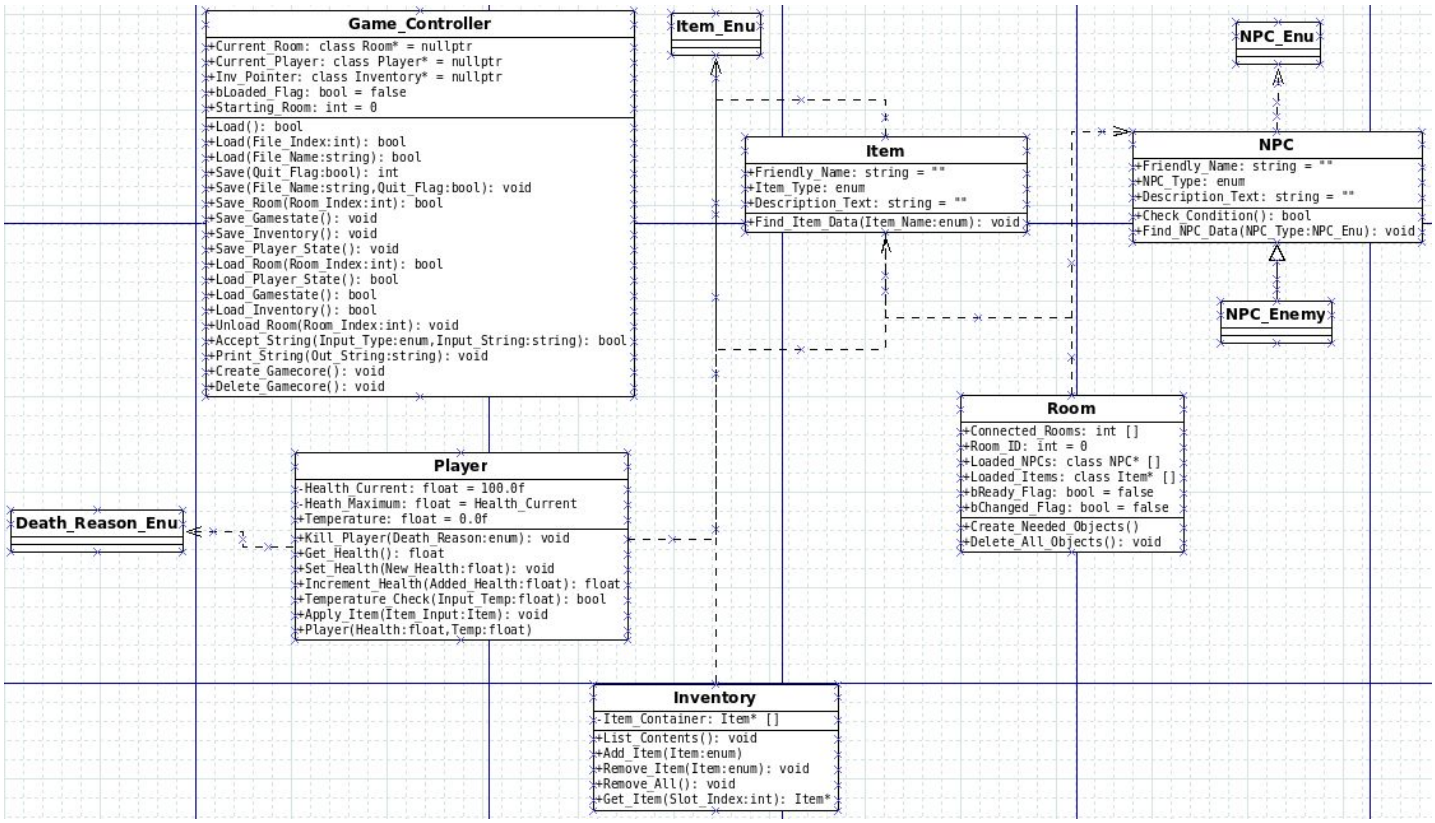
- Code Review
  - ➢ When any developer issues a pull request, the team lead will be responsible for approving and resolving the merge.
  - ➢ If the lead decides that there are issues with code being merged, the request will be denied, and the developer responsible for the faulty code will need to resolve the issue first.
- Communication Tools
  - ➢ The team has a private discord server that will be used for text or voice based communication regarding progress, deadlines, design, etc.
  - ➢ Errors and more specific code related issues can be communicated via bug reports in the repository
- Change Management
  - ➢ Any issues found and reported will be handled by the developer responsible for the majority of the related class/implementation.
  - ➢ If that developer is too busy or has more important issues to handle, and the issue needs to be resolved immediately, the team lead can contribute to fixing the bug.

Software Design

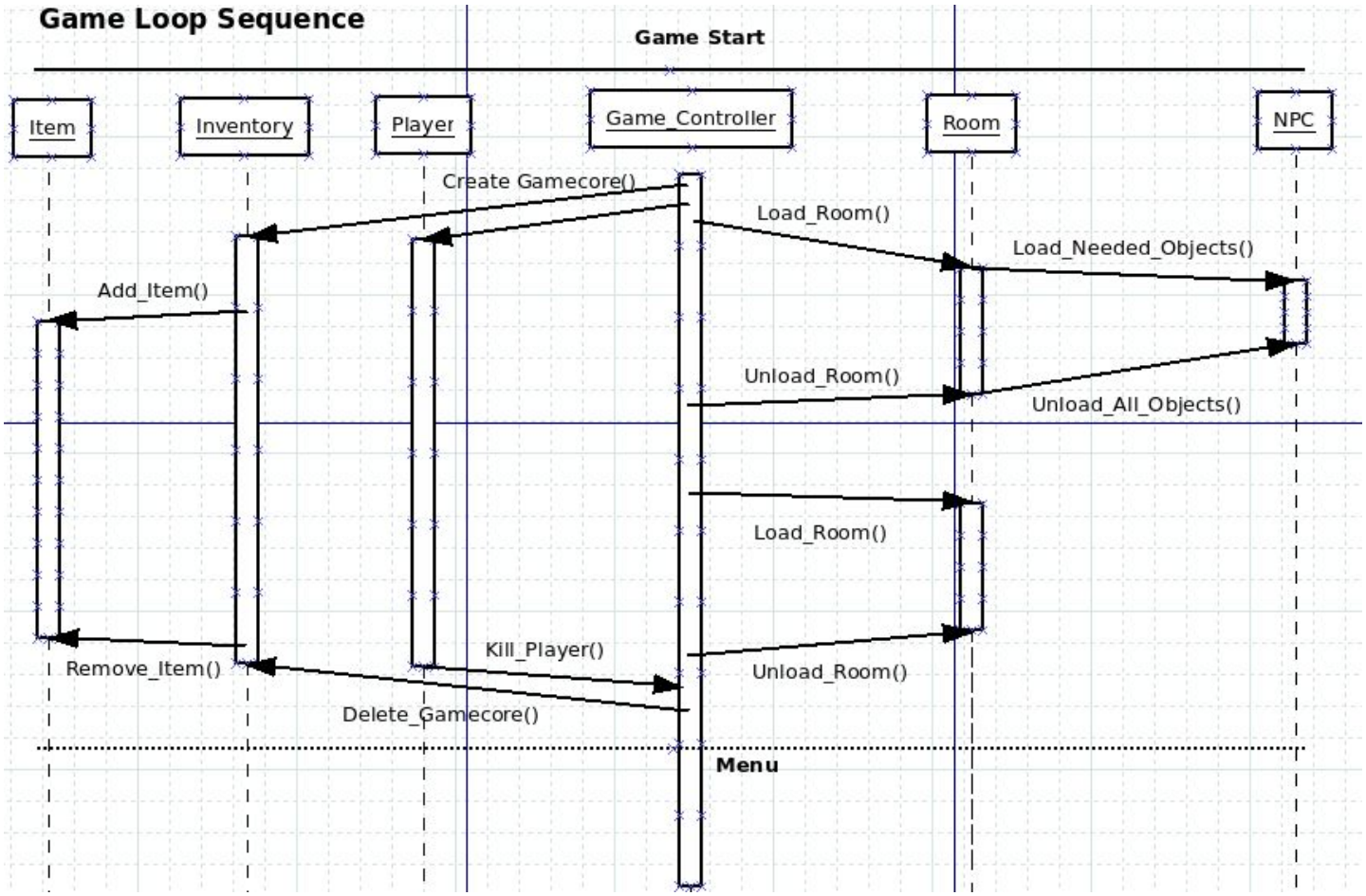
- Class Diagram
  - ➢ Appendix A
- Game State Diagram
  - ➢ Appendix B
- Saving The Game
  - ➢ Appendix C
- Loading The Game
  - ➢ Appendix D

Appendices

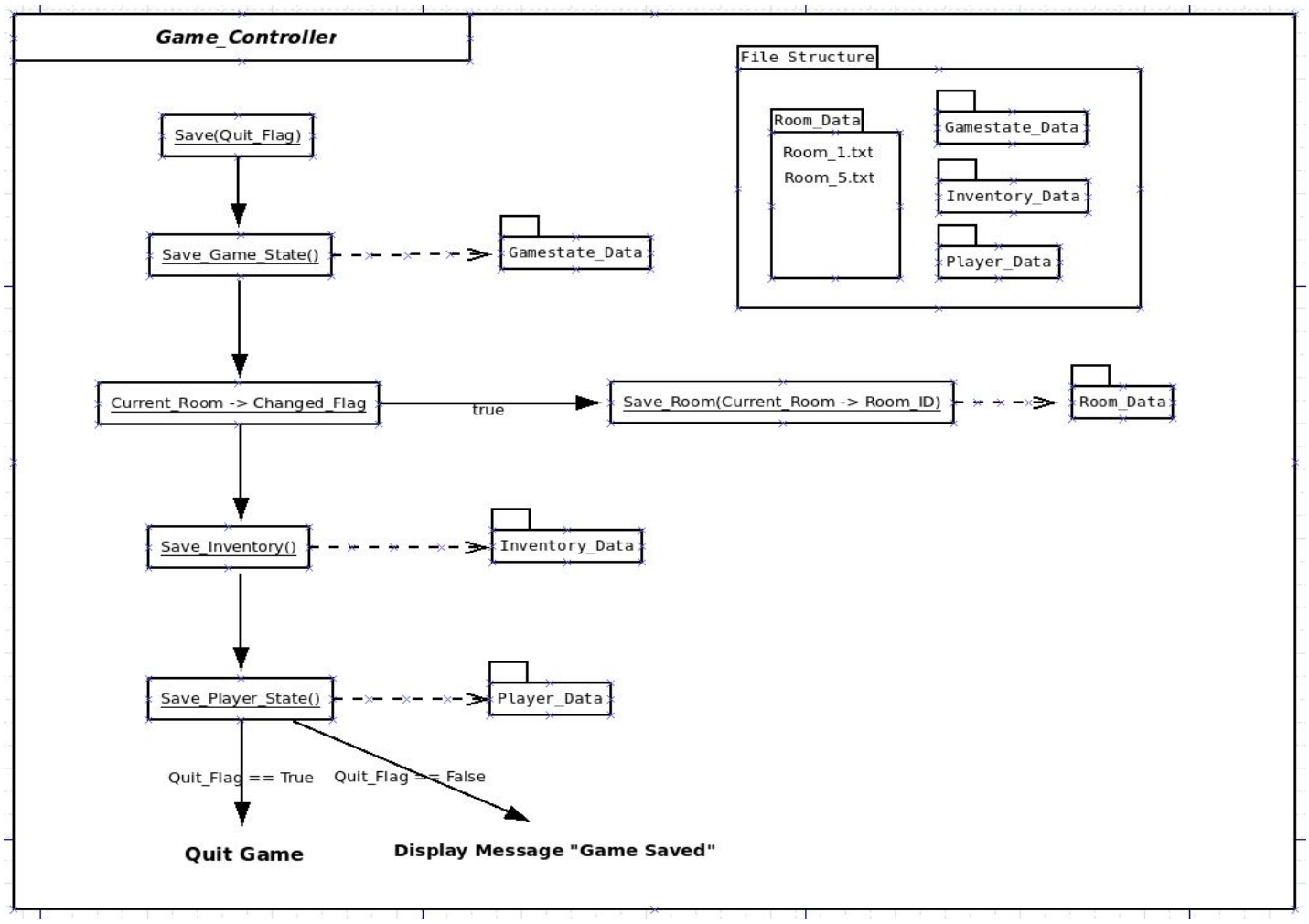
- A (Class Diagram)



- B (Game State Diagram)



- C (Saving The Game)



- D (Loading The Game)