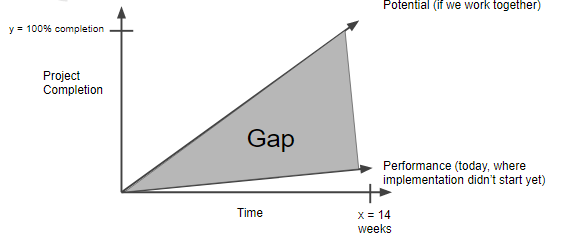
**“Capstone Defense” Project Documentation**

1. **Project vision** 
   1. **Backgrounds**

* Mitchell O’Hair: Computer Science, Java, C#, Unity, MySQL, Blender
* Doug Hartley: Computer Science, Java, quality assurance testing, diagrams, HTML
* James Bridges: Information Technology, Java, C#, Unity, MySQL, Html, Php, CSS, JavaScript
* Jagjot Singh: Computer Science, MySQL, Html, PHP, CSS, Java and JavaScript
* Jenna Mackool: Information Technology, Web Development, HTML, CSS,
* Stevan Rajkovic: Information Technology, QA, Java, PHP, CSS, HTML, C++, Shell scripting
  1. **Socio-economic Impact, Business Objectives, and Gap Analysis**

The socio-economic impact of this project is minimal. It will likely do little to change the lives of anyone except maybe put a smile on their face for an hour or so. This game should be enjoyable to young children and the elderly alike. Our Business Objective is to have monetization from either in-game purchases and/or Unity ads.



* 1. **Security and ethical concerns**

The database must be secure to prevent data breach

In order to view player data from the database, a login is required via the accessible website

Player time: how much time is too much?

Will not monetize via “loot boxes” as they are predatory in nature towards people with gambling addictions.

* 1. **Glossary of Key Terms**

Tower Defense: A genre of game in which the player places structures that hold off increasingly challenging discrete waves of virtual opponents. These games are typically endless and associated with a survival score.

1. **Project Execution and Planning** 
   1. **Team Information**

Mitchell O’Hair: [mlohair@oakland.edu](mailto:mlohair@oakland.edu) 248-408-0485

* 1. **Tools and Technology**

GitHub for file sharing

Trello for Sprint management

Visual Studio for IDE

Unity as the Game Engine

Discord and email for communication

Blender for 3D asset creation (and unity asset store)

* 1. **Project Plan**

Tower Defense - a player places structures to hold off discrete waves of opponents; the longer the player stays alive, the better the score

3D point defense game using C# and Unity gaming engine

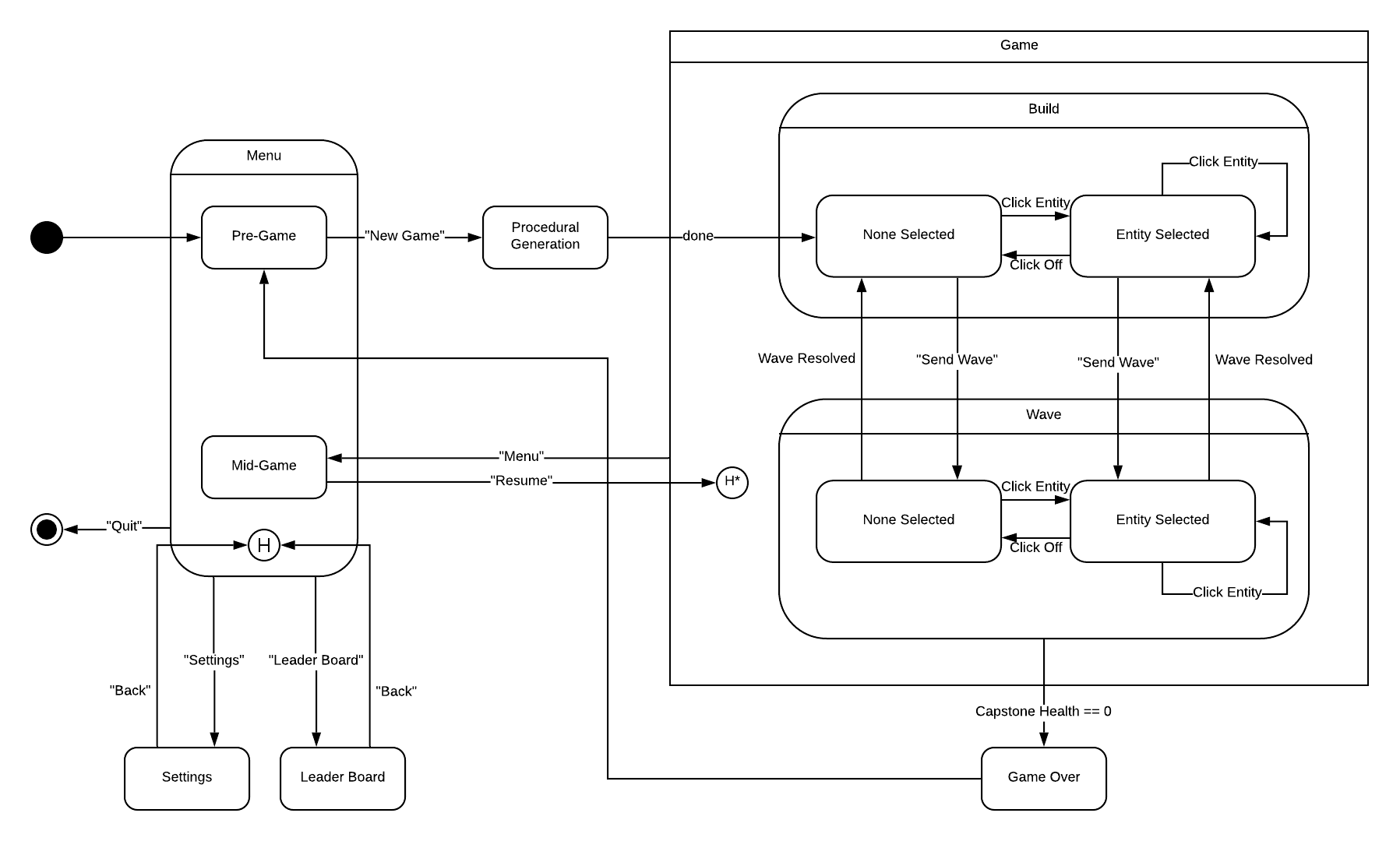
Audio will also be implemented using Unity

High scores and login information stored in SQL database

Website protected by username/password shows high scores

* 1. **Best standards and Practices**

1. **System Requirement Analysis** 
   1. **Function Requirements**
   2. **Non-functional Requirements**
   3. **On-Screen Appearance of landing and other pages requirements.**
   4. **Wireframe designs**
2. **Functional Requirements Specification** 
   1. **Stakeholders**
   2. **Actors and Goals**
   3. **User stories, scenarios and Use Cases**
   4. **System Sequence / Activity Diagrams**
3. **User Interface Specifications** 
   1. **Preliminary Design**
   2. **User Effort Estimation**
4. **Static Design** 
   1. **Class Model**
   2. **System Operation Contracts**
   3. **Mathematical Model**
   4. **Entity Relation**
5. **Dynamic Design 7.1. Sequence Diagrams.** 
   1. **Interface Specification**
   2. **State Diagrams**



1. **System Architecture and System Design** 
   1. **Subsystems / Component / Design Pattern Identification**
   2. **Mapping Subsystems to Hardware (Deployment Diagram)**
   3. **Persistent Data Storage**
   4. **Network Protocol**
   5. **Global Control Flow**
   6. **Hardware Requirement**
2. **Algorithms and Data Structures** 
   1. **Algorithms**
   2. **Data Structures**
3. **User Interface Design and Implementation** 
   1. **User Interface Design**
   2. **User Interface Implementation**
4. **Testing** 
   1. **Unit Test Architecture and Strategy/Framework**
   2. **Unit test definition, test data selection**
   3. **System Test Specification**
   4. **Test Reports per Spring**
5. **Project Management** 
   1. **11.1 Project Plan**
   2. **11.2 Risk management**
6. **References**