

Development Project - Final Report



Make-or-Break

Prepared by

Esat Duman, Jonathan Juarez, Kent Lizardo, Jose Tejeda

for use in CS 440

at the

University of Illinois Chicago

Spring 2024

Table of Contents

| | |
|--|----|
| List of Figures | 1 |
| List of Tables | 2 |
| I Project Description | |
| 1 Project Overview | 3 |
| 2 The Purpose of the Project | 3 |
| 3 The Scope of the Work | 5 |
| 4 The Scope of the Product | 8 |
| 5 Stakeholders | 9 |
| 6 Mandated Constraints | 11 |
| 7 Naming Conventions and Definitions | 13 |
| 8 Relevant Facts and Assumptions | 14 |
| II Requirements | |
| 9 Product Use Cases | 15 |
| 10 Functional Requirements | 22 |
| 11 Data Requirements | 24 |
| 12 Performance Requirements | 24 |
| 13 Dependability Requirements | 26 |
| 14 Maintainability and Supportability Requirements | 28 |
| 15 Security Requirements | 30 |
| 16 Usability and Humanity Requirements | 32 |
| 17 Look and Feel Requirements | 35 |
| 18 Operational and Environmental Requirements | 36 |
| 19 Cultural and Political Requirements | 37 |
| 20 Legal Requirements | 38 |
| 21 Requirements Acceptance Tests | 39 |
| III Design | |
| 22 Design Goals | 46 |
| 23 Current System Design | 46 |
| 24 Proposed System Design | 47 |
| 25 Additional Design Considerations | 51 |
| 26 Final System Design | 57 |
| 27 Object Design | 57 |

IV Project Issues

| | | |
|-----------|-------------------------------------|-----------|
| 28 | Open Issues | 61 |
| 29 | Off-the-Shelf Solutions | 61 |
| 30 | New Problems | 62 |
| 31 | Migration to the New Product | 64 |
| 32 | Risks | 64 |
| 33 | Costs | 65 |
| 34 | Waiting Room | 66 |
| 35 | Ideas for Solutions | 67 |
| 36 | Project Retrospective | 67 |

| | |
|-------------------|-----------|
| V Glossary | 69 |
|-------------------|-----------|

| | |
|----------------------|-----------|
| VI References | 70 |
|----------------------|-----------|

| | |
|------------------|-----------|
| VII Index | 70 |
|------------------|-----------|

List of Figures

| | |
|--|-----------|
| Figure 1 - Context Diagram | 6 |
| Figure 2 - Create Mode Scenario Diagram | 8 |
| Figure 3 - Play Mode Scenario Diagram | 8 |
| Figure 4 - Product Use Case Diagram | 15 |
| Figure 5 - Initial System Design | 47 |
| Figure 6 - Sequence Diagram #1 | 48 |
| Figure 7 - Sequence Diagram #2 | 49 |
| Figure 8 - Class Diagram | 50 |
| Figure 9 - Deployment Diagram | 51 |
| Figure 10 - Data Storage Design | 52 |
| Figure 11 - Home Screen Mockup | 54 |
| Figure 12 - Gameplay Mockup | 55 |
| Figure 13 - Challenge Menu Mockup | 55 |
| Figure 14 - Unity Prototype Design | 56 |
| Figure 15 - Overall System Design | 57 |
| Figure 16 - System Design: App.main | 58 |
| Figure 17 - System Design: App.network | 59 |
| Figure 18 - System Design: App.game.level | 60 |

List of Tables

| | |
|--|-----------|
| Table 1 - Business Event List Table | 6 |
| Table 2 - Product Use Case List | 16 |
| Table 3 - Requirements - Test Correspondence Summary | 39 |
| Table 4 - Project Issues - Estimated Cost of Production | 65 |

I Project Description

1 Project Overview

Make-or-Break is an application utilizing real-time physics simulations in order to test virtual building designs against the forces of nature. Make-or-Break allows users to create, edit, and load real-world recreations of buildings and structures in the 3D environment. Users will be able to test their creations against the forces of nature, such as torrential rain and flooding, earthquakes, and extreme wind/tornadoes. While Make-or-Break is meant to be used for realistic simulations and structural planning, at its core it is still a game so the users will be able to enter a **Challenge Mode**. In Challenge Mode, users will be able to design and create structures in order to survive waves of challenges, varying from real world and fantastical. In addition to the challenge mode, users will also enjoy a **Sandbox Mode**, granting them the freedom to design their buildings without budget limitations or disaster constraints, allowing for the formulation of personalized gameplay strategies. Make or Break is meant to inspire creativity, foster strategic thinking, and empower users to navigate the dynamic interaction of nature and architecture, Make-or-Break promises an immersive experience where innovation meets challenge, shaping the future of virtual construction and simulation gaming.

2 Purpose of the Project

Make-or-Break provides a platform that empowers users to simulate and test the structural integrity of buildings against each of the various natural disasters. By offering a real-time physics simulation, Make or Break aims to **assist designers, engineers, and architects** in designing safer and more resilient structures.

2a The User Business or Background of the Project Effort

Make-or-Break operates within the domain of structural engineering and architecture, focusing on ensuring the resilience of buildings against natural disasters. The platform serves as a simulation tool empowering users, primarily designers, engineers, and architects, to assess and enhance the structural integrity of buildings in the face of diverse natural calamities.

The reason behind Make-or-Break stems from the pressing need to mitigate the catastrophic impact of natural disasters on buildings. With the escalating frequency and severity of such events globally, there is an urgency to fortify

structures against earthquakes, hurricanes, floods, and other natural hazards. A smaller scale example that happened recently was the heavy snowfall in Texas last winter, which left many houses heavily damaged due to poor design and preparation planning. Make-or-Break strives to address this by providing a platform that enables professionals to anticipate potential vulnerabilities and reinforce structures accordingly. Ultimately, the motivation is to foster safer and more resilient communities by promoting best practices in architectural and engineering design.

2b Goals of the Project

The goal behind Make or Break is to help create safer environments by promoting safety and sustainability in construction and design practices. Through our platform, users can explore different construction materials, building designs, and disaster scenarios to optimize their projects for maximum safety and durability. By offering a real-time physics simulation, Make or Break aims to assist designers, engineers, and architects in designing safer and more resilient structures. Make or Break empowers professionals to make informed decisions that mitigate risks and enhance the resilience of their structures in the face of all kinds of environmental hardships. Our mission is to revolutionize the way buildings are conceptualized and constructed, fostering a culture of innovation and resilience in the architectural and engineering communities worldwide.

2c Measurement

- 1. Increase in Safety Standards:** Measure the reduction in structural failures and casualties in real-world construction projects after architects and engineers have utilized Make-or-Break simulations to optimize their designs.
- 2. Innovation in Design:** Assess the number of architectural concepts and engineering solutions generated by users in the various game modes, indicating the platform's success in inspiring creativity and fostering innovation.
- 3. Resilience Improvement:** Evaluate the ability of structures designed using Make-or-Break to withstand extreme weather events and natural disasters, measured by the reduction in damage and downtime experienced during such occurrences.
- 4. Customer Satisfaction:** Measure user satisfaction through surveys or ratings, assessing the platform's usability, effectiveness, and value in aiding design processes and decision-making.

3 The Scope of the Work

When considering the development and release of a game in today's mature gaming market, standing out is paramount. Many contemporary games rely on standard physics engines, but Make or Break seeks to innovate in this regard. We plan to utilize a physics engine that prioritizes realism, determinism, and accuracy, such as Havok or Jolt, or even explore the possibility of crafting our own engine specifically tailored for architecture and engineering. By focusing on authenticity and precision in our physics simulation, Make or Break is poised to offer players an unparalleled gaming experience. Moreover, understanding the current landscape and context of gaming, as well as the nuances of the working activities performed by players or teams, is crucial. This involves examining how our game interacts with its surroundings, partitioning work by business events and their responses, and understanding competing products to ensure our offering stands out in the market.

3a The Current Situation

The kind of person who would buy this product enjoys physics-based and puzzle-based gameplay. They like to view a challenge, think of different ways to solve it, and be able to accomplish that challenge in a timely manner. This player also would have a focus on realism while also being able to enjoy interesting reactions emerging from the game state. This kind of player is a problem solver, and is willing to go through a series of increasing challenges. This player also enjoys being able to compete with others, such as finding the easiest way to pass a level, or even creating the most outlandish set of inputs that can complete a level. They also like to compete with other players, while creating and sharing content for their community. Another kind of player we are aiming to create the product for is the creative and analytical kind. They like to learn about different options they can use, research them, and find out more about them through more gameplay.

3b The Context of the Work

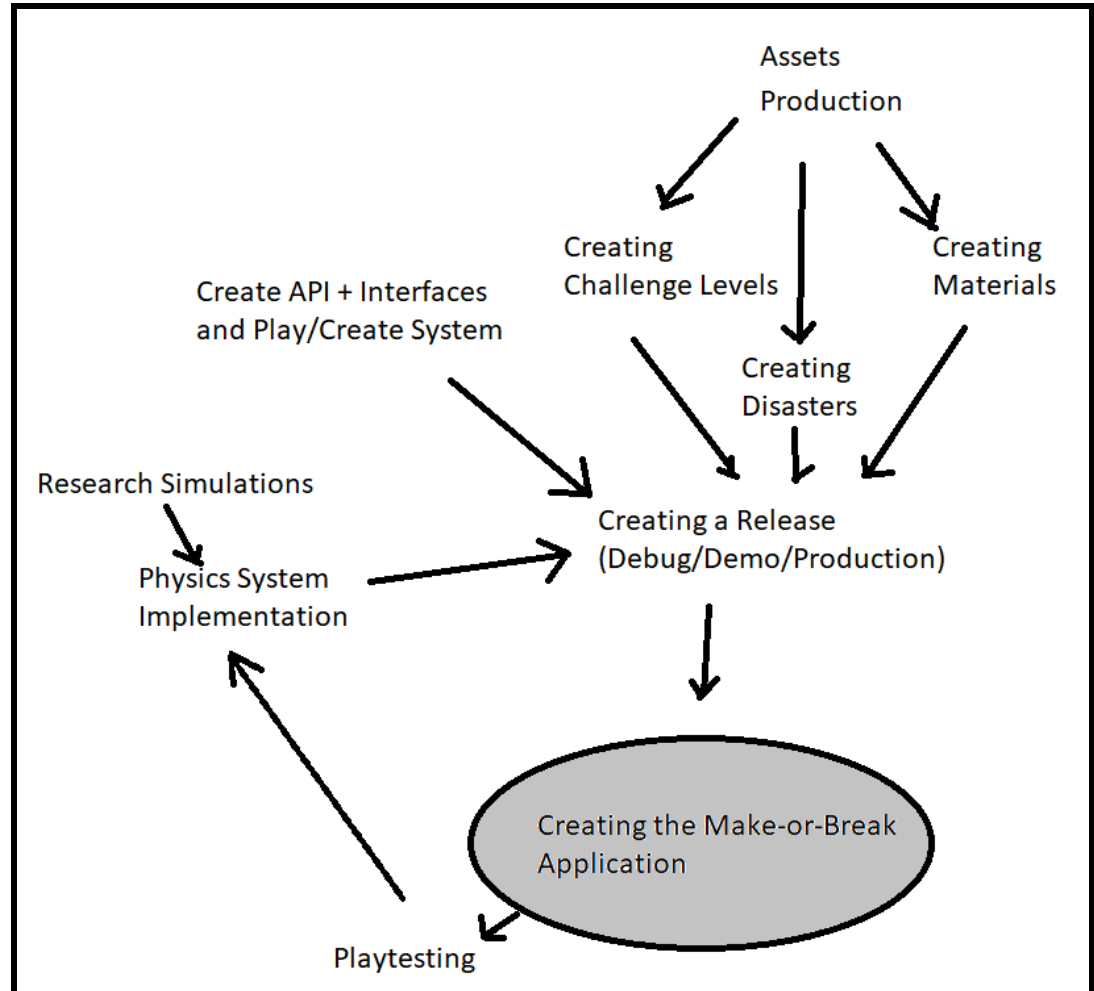


Figure #1 - Context Diagram

3c Work Partitioning

Business Event List

| Event Name | Input and Output | Summary |
|--------------------------------|---|---|
| Update to application code. | Create API + Interfaces and Play/Create System (in) | The general structure of the game is implemented/extended due to the creation or updating of new systems. |
| A production build is created. | Creating a Release (out) | A new stable major release is created. Either due to a large amount of milestones being implemented, or a |

| | | |
|--|---|--|
| | | stable point of development is found. |
| Continuous deployment releases production build. | Creating a Release (in) Playtesting (out) | A release of the game is distributed to playtesters or players through a CI/CD pipeline. |
| Playtest finds physics bugs | Playtesting (in) Physics System Implementation (out) | Playtesters or players find issues with how the physics behaviors work |
| Playtest finds application bugs. | Playtesting (in) Create API + Interfaces and Play/Create System (in) | Playtesters or players find a bug or issue with the game's code, either resulting in a crash or unplayable game. |
| Assets received from production | Assets production (in) Creating Challenge Levels (out) Creating Materials (out) | Production team sends new assets to development team, which team creates new content for next release. |
| Challenge level updated/added to release. | Creating Challenge Levels (in) Creating a Release (out) | Development team publishes new challenge levels to campaign |
| New material added to release | Creating Materials (in) Creating a Release (out) | Development team publishes new materials for players to construct with. |
| New disasters implemented | Creating Disasters (in) Creating a Release (out) | Development team publishes new disasters for challenges to use. |

Table #1 - Business Event List Table

3d Competing Products

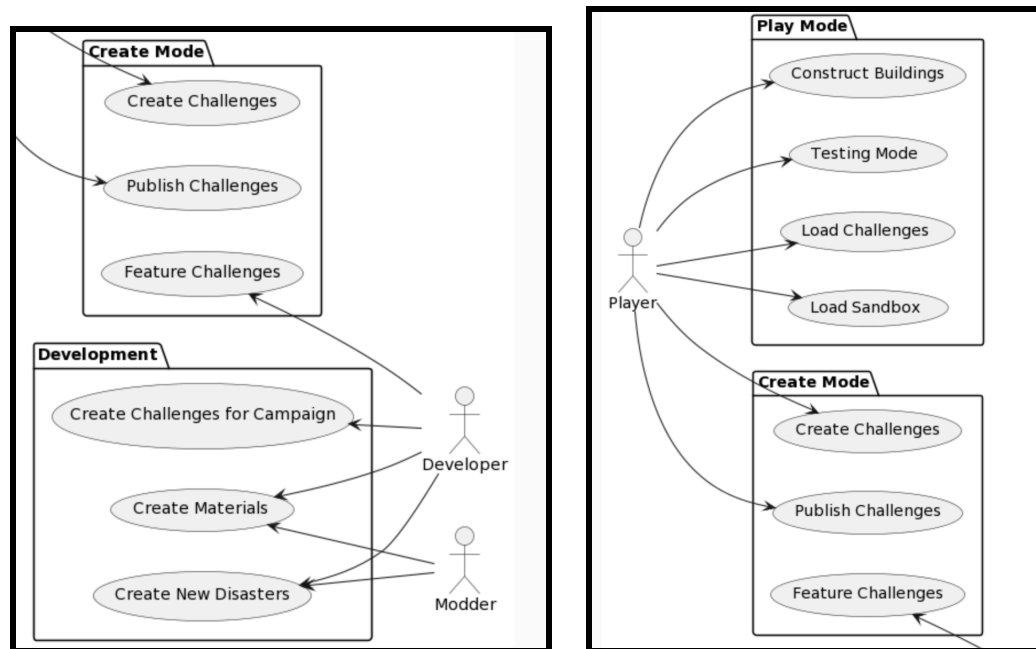
1. **Cities: Skylines:** While not solely focused on testing buildings against natural disasters, Cities: Skylines offers a city-building simulation where players can design and manage entire cities, including infrastructure and urban planning. Some expansions and mods also introduce elements of disaster management.

2. **Poly Bridge:** Poly Bridge focuses on bridge construction and simulation, challenging players to design bridges that can withstand various loads and scenarios. While it doesn't directly simulate natural disasters, it offers a similar experience of testing structures against physical forces.
3. **SpaceEngine:** Although more focused on space exploration and simulation, SpaceEngine allows users to create and explore vast, procedurally generated galaxies and planetary systems. While not directly competing in terms of construction or disaster simulation, it offers a different avenue for creativity and exploration.

4 The Scope of the Product

Post-release, the game should be able to be supported by Developers and from the community by Modders. Developers can Create Challenges for Campaign, Create Materials, and Create new Disasters, as well as Feature Challenges that will be seen in Create Mode. Modders can Create Materials and Create New Disasters.

4a Scenario Diagram(s)



Figures # 2, 3 - Create Mode / Play Mode Scenario Diagrams

5 Stakeholders

WHO is interested in a service like this? Interest for Make or Break will span various sectors: from architecture, engineering, construction, government agencies, to even educational Institutions.

5a The Client

The main clients we expect Make or Break to pull are:

1. **Architect & Engineering Firms:** Find value in Make or Break due to its capability to enhance the design process, improve structural integrity, and simultaneously minimizing risks associated with their projects.
2. **Construction Companies:** See potential for reduced waste in projects, streamlined construction process, and the ability to ensure the durability and safety in their structures.
3. **Government Agencies:** Offers a way to enforce structures to uphold building codes and regulations. Would also promote sustainability in the development of infrastructure.

5b The Customer

The main customers Make or Break will target are:

1. **Educational Institutions:** Provides an immersive tool for students to learn about the principles of structural engineering and sustainable construction practices through hands-on simulation activities.
2. **Insurance Companies:** Potential to mitigate risks associated with structural failures and enables companies to make more informed decisions, potentially reduce claim payouts, while promoting durability from their insured properties.
3. **Players:** Aspiring architects, designers, and general enthusiasts would be drawn to Make or Break for its engaging gameplay experience. The game would allow for users to explore their creativity, test their design skills, all while learning about the complexities of architectural/structural engineering in a fun virtual environment.

5c Hands-On Users of the Product

Students from high school/university learning intro to engineering courses will use our games hands-on for their projects to introduce them to the field. Other

potential hands-on users consist of building architects who create and develop blueprints for structures and buildings and can submit their designs to our game, seeing the results of their work against the simulations of natural disasters. This allows them to prepare for the worst and save the environment from destruction and the lives of many. Gathering information based on the results can save time and money for companies when engineering designs to create natural disaster-proof structures and buildings.

5d Other Stakeholders

1. **Urban Planners:** Allows planners to pick a stable location for building physical layouts of urban spaces. Can test the layout for urban spaces by simulating natural disasters to make the correct decisions for safe and stable urban spaces.
2. **NonProfit Organizations:** Disaster relief organizations can provide input on the game for the impact on the social and environmental effects of the disaster that has occurred.
3. **Contractors:** Contractors that build or renovate homes can use the game to see the results of their renovations and potentially have more clients for their business because of the results that they bring to the table without game.

5e User Participation

While developing the game, users will participate in surveys and add comments, allowing the developers to make changes accordingly. Playtesting will be available for architects, structural engineers, and other professionals to simulate the game and give us feedback on the realism and physics aspects of the game to be as accurate and realistic as possible. Developers will engage with the community of players through various platforms such as YouTube, Discord, and TikTok. Updates and the game's development process will be shared with the community. Users will also be open to sharing their ideas through mods and engagements with the developers through these social media platforms.

5f Priorities Assigned to Users

For our **key users**, we found that the most important users for our product would include **content creators**, as they are the one of the vital parts in growing a community for a game. Another key user would be **structural/civil engineers**, or other construction professionals, as their opinion on how our physics system is

able to deliver on realism can determine our success in regards to the educational aspect of our game. Another key user would be **curators**, in regards to players who participate in the community by creating content such as challenge levels or mods. They also are much more likely to engage with any forums and sites related to our game.

Secondary users would be players that buy our product without the intent of spreading it to others. Their opinion is still valuable, we want most players to enjoy the game even if they don't decide to participate in the community. Another secondary user would be players who aren't knowledgeable about how buildings work in real life, as they would value more outlandish results of games, rather than the realism aspect.

Unimportant users would be players who buy our game just to collect it, as part of their collections of games on stores such as Steam or Epic Games. Another unimportant user would be users who spread malicious or inappropriate content in centers of our player community. This also includes players who download the game from unauthorized sources as they don't have access to full support of our continued development without an official release build.

6 Mandated Constraints

6a Solution Constraints

Description: The product will be compatible with Windows, Linux, Mac-OS.

Rationale: Make or Break is a PC only game and its user base will most likely be running one of the three listed operating systems.

Fit Criterion: Make or Break will be tested to work with all three of the listed operating systems

Description: Moderated and secure distribution of Challenge levels to all players.

Rationale: Game updates will introduce new levels to the user which should include any player no matter their difficulty level.

Fit Criterion: Game testers will play and verify each level is fit to match its designated difficulty level.

Description: All levels are forwards-compatible with production versions released.

Rationale: Users may want to access older completed levels to replay.

Fit Criterion: Game testers will play each level in new version releases to ensure they function properly.

Description: Virtual buildings and behavior of the simulation must try to accurately reflect how they react in real life disasters.

Rationale: Users may want to create and visualize real life disasters inside Make or Break.

Fit Criterion: Some example scenarios will be created and run by game testers used to determine the realness of the simulation.

Description: Periodical moderation and security for published content.

Rationale: To prevent large amounts of inappropriate/malicious content published from players in the community.

Fit Criterion: Challenges and mods created from the community must go through a verification process in order to be accessible to the public user base.

6b Implementation Environment of the Current System

Accessible on PC to low-end hardware. Data can be retrieved using external sources such as Steam Hardware Survey. The site lists and sorts hardware used by its users. Make or Break testing will be used on low-end hardware gathered from external sources to ensure the experience is optimal for users.

6c Partner or Collaborative Applications

There are no external applications Make-or-Break must be compatible with; however, an API could be made to extend functionality to other apps. Such apps might be online database rankings for challenges, or a webhook that shows statistics of a player's profile.

6d Off-the-Shelf Software

The game engine in which Make or Break will be created in (ex. Unreal Engine 5, Unity, etc...).

6e Anticipated Workplace Environment

Make or Break is a PC only game, which means the user's environment will more than likely be their home or outdoor environments where a PC is of acceptable use. Some outdoor environments include: Library, Gaming cafe. Conditions vary

from user to user and location which will require settings to allow users to customize sound volume and graphics quality depending on their computers power.

6f Schedule Constraints

Planning: 3 months

Pre-Production: 4 months

Production: 8 months

Testing: 2 months

Pre-Launch: 1-2 months

Launch: 1-2 months

Post-Launch: 2-5 years of long term support

(Rough estimation of schedule timelines, will vary depending on team size)

6g Budget Constraints

Content Creation: \$10,000 - \$100,000+

Software Development: \$100,000 - \$120,000+

Production: \$80,000 - \$100,000+

QA Analysis and Testing: \$10,000 - \$12,000+

Hardware and Infrastructure: \$1,000 - \$20,000+

Staffing: \$50,000 - \$500,000 (yearly)

Server and Hosting: \$1,000 - \$10,000 (monthly)

Legal and Compliance: \$5,000 - \$20,000+

Game Insurance: \$2,000 - \$10,000+

7 Naming Conventions and Definitions

Player: person who buys and plays a release version of the game on their system.

Deterministic physics: physics where the outcome is solely determined by the inputs. Same inputs and variables give you the same results with no randomness.

Updates and Mods: modifications made to the game post-release, (mods usually by players, and updates/patches by developers)

Forwards Compatibility: any version released after specific content was made can load and access that content through conversion or good data practices.

7a Definitions of Key Terms

Assets: Resources used during development such as textures, audio, 3d models, and UI graphics. Generally released through an executable distribution.

CI/CD: Continuous integration/deployment. A regulated pipeline for seamlessly releasing production builds for playtesters with an emphasis on frequent incremental updates.

8 Relevant Facts and Assumptions

8a Facts

1. Make-or-Break is a simulation application utilizing real time physics simulations for testing virtual building designs against natural forces.
2. The application allows users to create, edit, and load real-world recreations of buildings and structures in a 3D environment.
3. Users can test their creations against various natural forces such as torrential rain and flooding, earthquakes, extreme wind/tornadoes, among others.
4. An estimated 38% of “dedicated PC gamers” use laptops as their main gaming device.

8b Assumptions

1. Users have access to a PC with specifications meeting or exceeding the minimum requirements for an optimal gaming experience.
2. Make-or-Break does not depict real structures or contain any living entities; it is purely a simulation and gaming experience.

II Requirements

9 Product Use Cases

9a Use Case Diagrams

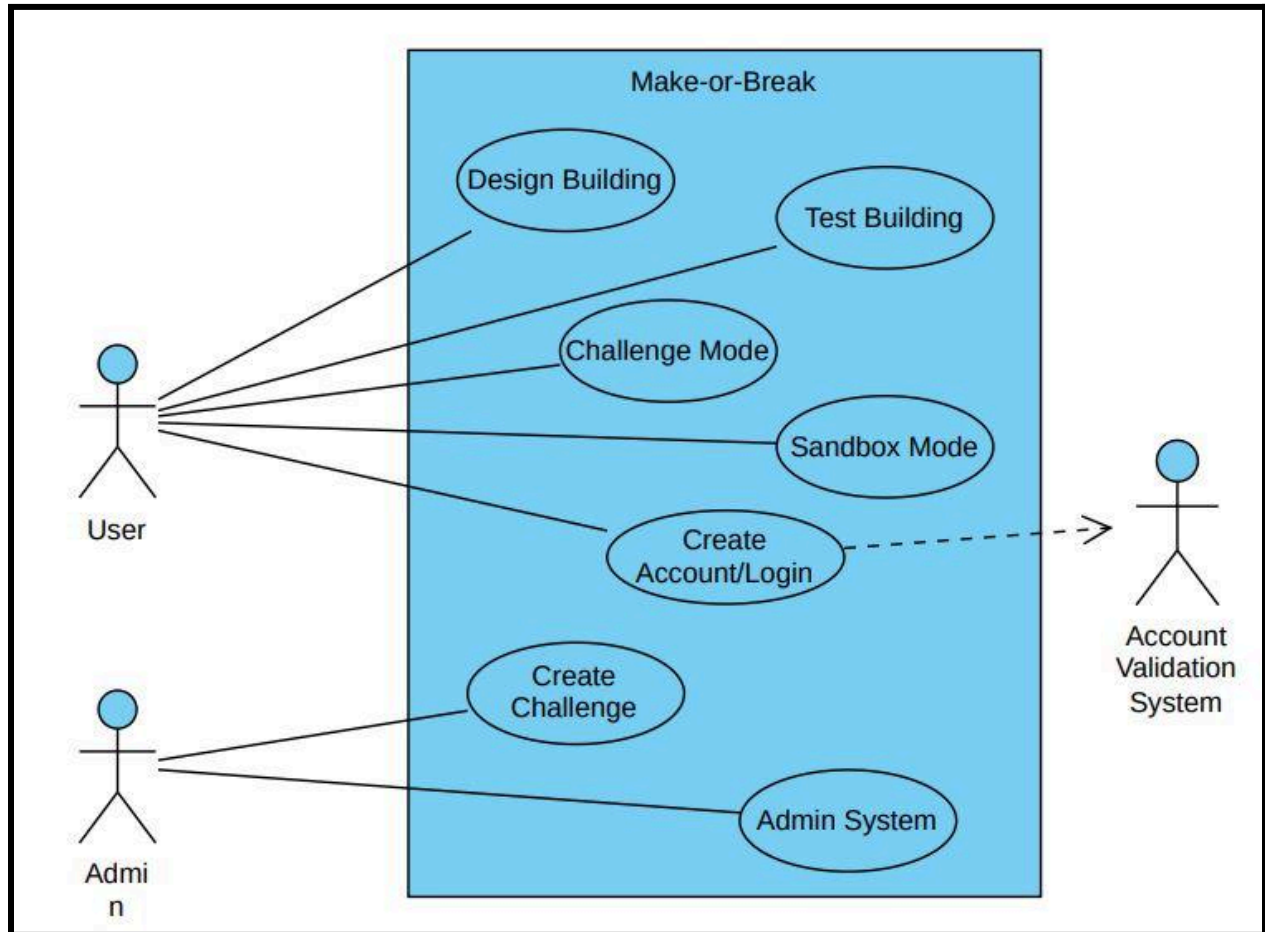


Figure #4 - Product Use Case Diagram

9b Product Use Case List

| Use Cases | Description | Actors | Preconditions |
|------------------------|--|--------|--|
| Design Building | Users can create, edit, and load real-world recreations of buildings and structures in the 3D environment. | User | User is logged into the Make-or-Break application. |

| | | | |
|------------------------------|---|---------------|--|
| Test Building | Users can test their building designs against various forces of nature, such as rain, flooding, earthquakes, extreme wind/tornadoes, etc. | User | User has designed or loaded a building into the Make-or-Break application. |
| Challenge Mode | Users can enter a mode where they design and create structures to survive waves of challenges, varying from real-world and fantastical scenarios. | User | User is logged into the Make-or-Break application. |
| Sandbox Mode | Users can design their buildings without budget limitations or disaster constraints, allowing for personalized gameplay strategies. | User | User is logged into the Make-or-Break application. |
| Create Challenge | Challenge creators can create new challenges for the Challenge Mode, specifying the parameters and objectives for the challenges. | Administrator | Administrator is logged into the Make-or-Break application. |
| Administrative System | Administrators can perform system administration tasks such as user management, data management, maintaining the platform, etc. | Administrator | Administrator is logged into the Make-or-Break application. |

Table #2 - Product Use Case List

9c Individual Product Use Cases

| |
|---|
| <p>Use Case ID: 001 Name: Create Account</p> <p>Preconditions: Users cannot have a preexisting account with the information they plan to use to sign up.</p> <p>Postconditions: User successfully creates a new account in the Make-or-Break application.</p> <p>Initiated By: User</p> |
|---|

Trigger Event: User selects the option to create a new account.

Additional Actors: None

Sequence of Events:

1. User accesses the Make-or-Break application.
2. User selects the option to create a new account.
3. System presents the user with a registration form. User fills out the registration form with required information (e.g., username, email, password). System validates the information provided by the user.
 - a. If the information is valid, the system creates a new account for the user.
 - b. If the information is invalid or incomplete, the system displays error messages and prompts the user to correct the information.
4. Once the account is successfully created, the system automatically logs the user into their new account.

Alternatives: If the user does not want to create a new account, they can choose to proceed as a guest user with limited functionality. Users may have the option to sign up for an account using a mobile application instead of the web interface.

Exceptions: If the username or email provided by the user is already associated with an existing account: System displays an error message and prompts the user to choose a different username or email.

Use Case ID: 002

Name: Login

Preconditions: User has registered an account with Make-or-Break. User is not currently logged into an account.

Postconditions: User successfully logs into the Make-or-Break application and gains access to their account. User's client is logged into the account.

Initiated By: User

Triggering Event: User accesses the Make-or-Break application with the intent of accessing online features.

Additional Actors: None

Sequence of Events:

1. A user uses the Make-or-Break application in a way that requires authentication, e.g. accessing or uploading published content.
2. System presents the login screen to the user.
3. User enters their username and password then sends the query online.

4. System verifies login credentials and provides the user with a valid session token (or any other way of secure temporary access).
 - a. If the credentials are valid, the system grants access to the user's account.
 - b. If the credentials are invalid, the system displays an error message and prompts the user to re-enter their credentials.

Alternatives:

Forgotten Password

1. User clicks on the "Forgot Password?" link.
2. System prompts the user to enter their email address.
3. System sends a password reset link to the user's email address.
4. User follows the link to reset their password.
5. The user may then follow the normal login procedure after the password is reset.

Exceptions: If the system encounters an error during the login process: System displays an error message to the user and prompts them to retry or contact support.

Use Case ID: 003

Name: Design Building

Preconditions: User has the Make-or-Break game application running.

Postconditions: The user's building is saved into a collection stored on the client..

Initiated By: User

Triggering Event: User selects the option to design a new building.

Additional Actors: None

Sequence of Events:

1. User enters Create Mode in the main menu.
2. User selects the option to design a new building. System presents the user with tools and options for designing the building.
3. The User is provided with a suite of creation tools to design a building.
4. Once the design is complete, the user saves the building design for future use.

Alternatives: Users may choose to start with a pre-existing template or blueprint instead of creating a building design from scratch. Users may have the option to collaborate with other users in real-time to design a building together.

Importing

1. User enters Create Mode in the main menu.
2. User selects "import building"
3. File dialog is shown, only selecting certain file types.

4. The game then imports a supported file type into a building design in the building editor.
5. The user may then proceed to edit the building as normal.

Exceptions: None

Use Case ID: 004

Name: Test Building

Preconditions: User has designed or loaded a building into the Make-or-Break application.

Postconditions: User observes the results of the building test simulation.

Initiated By: User

Triggering Event: User selects the option to test the building.

Additional Actors: None

Sequence of Events:

1. User selects an option to test their building.
2. User chooses the type and intensity of the natural disaster to simulate.
3. Make-or-Break simulates the effects of the selected disaster on the building.
User observes the results and adjusts the building design as necessary.

Alternatives: Users may have the option to simulate different scenarios or combinations of disasters to test the building's resilience under various conditions. Users may choose to share the results of their building tests with others for feedback or collaboration purposes.

Exceptions: None

Use Case ID: 005

Name: Challenge Mode

Preconditions: The application is currently running on the user's device.

Postconditions: User completes a challenge scenario successfully or unsuccessfully.

Initiated By: User

Triggering Event: User selects the Challenge Mode from the main menu.

Additional Actors: None

Sequence of Events:

1. User selects the Challenge Mode from the main menu.
 - a. User browses available challenges or creates a new challenge.

2. User designs a structure to meet the objectives of the selected challenge.
3. User submits the structure for evaluation and plays through the challenge scenario.

Alternatives: Users may choose to participate in challenge scenarios individually or as part of a team. Users may have the option to skip certain challenges or return to them later if they are unable to complete them at the moment.

Exceptions: None

Use Case ID: 006

Name: Sandbox Mode

Preconditions: User is logged into the Make-or-Break application.

Postconditions: User creates a building design in the sandbox mode.

Initiated By: User

Triggering Event: User selects the Sandbox Mode from the main menu.

Additional Actors: None

Sequence of Events:

1. User accesses the Make-or-Break application.
2. User selects the Sandbox Mode from the main menu.
3. User begins building without any restrictions.
 - a. Users can experiment with different designs, materials, and strategies.
4. User saves or discards the sandbox project as desired.

Alternatives: Users may choose to collaborate with others in real-time while building in the sandbox mode, allowing for group creativity and brainstorming. Users may have the option to load and modify existing building designs from the sandbox mode instead of starting from scratch.

Exceptions: None

Use Case ID: 007

Name: Create Challenge

Preconditions: Challenge Creator is logged into the Make-or-Break application.

Postconditions: A new challenge is available for users to participate in Challenge Mode.

Initiated By: Administrator / Challenge Creator

Triggering Event: Challenge Creator selects the option to create a new challenge.

Additional Actors: None

Sequence of Events:

1. Challenge Creator accesses the Make-or-Break application.
2. Challenge Creator selects the option to create a new challenge.
3. Challenge Creator defines the objectives, constraints, and parameters of the challenge.
4. Challenge Creator sets the criteria for successful completion of the challenge.
5. Challenge Creator saves the new challenge for users to access in Challenge Mode.

Alternatives: Challenge creators may have the option to collaborate with others in creating challenges, allowing for joint efforts in challenge design. Challenge creators may choose to modify existing challenges instead of creating entirely new ones, providing variations on existing scenarios.

Exceptions: None

Use Case ID: 008

Name: Administrative System

Preconditions: Administrator is logged into the Make-or-Break application.

Postconditions: System administration tasks are completed successfully.

Initiated By: Administrator

Triggering Event: Administrator accesses the system administration panel.

Additional Actors: None

Sequence of Events:

1. Administrator accesses the Make-or-Break application.
2. Administrator navigates to the system administration panel.
3. Administrator manages user accounts, including creating, updating, or deleting accounts.
 - a. Administrator manages data within the application, including backups, restores, and data integrity checks.
4. Administrator performs maintenance tasks to ensure the smooth operation of the Make-or-Break platform.

Alternatives: The system may provide automated tools for system maintenance tasks, allowing the administrator to schedule routine operations such as backups or data integrity checks.

Exceptions: None

10 Functional Requirements

ID#F-1: View Campaign Challenges

Description: Provide a progression-based view for a set of designed challenges with incremental difficulty.

Rationale: The user should be able to experience the game without relying on other players for content.

Fit Criterion: The progression-based view should display a series of challenges with increasing difficulty levels, clearly indicating the completion status of each challenge and providing a way for the user to navigate through them.

Acceptance Tests: 1 (Levels) F-2

ID#F-2: View Online Published Challenges

Description: Provide a list of descriptions and previews for challenges uploaded by players.

Rationale: This feature allows users to access a variety of user-generated content, increasing the replay value and community engagement of the game.

Fit Criterion: The list of online published challenges should include concise descriptions and previews of each challenge, allowing users to make informed decisions about which challenges to play.

Acceptance Tests: 1 (Levels) F-1

ID#F-3: Building Creation

Description: Users can create, edit, and load real-world recreations of buildings and structures within the 3D environment of Make-or-Break.

Rationale: This functionality allows users to engage in the simulation process by designing and importing structures for testing against natural forces.

Fit Criterion: The user should be able to import 3D models of buildings, edit them within the application, and save/load their creations successfully.

Acceptance Tests: 2 (Game Loop)

ID#F-4: Real-Time Physics Simulation

Description: Make-or-Break utilizes real-time physics simulations to test virtual building designs against natural forces like torrential rain, flooding, earthquakes, and extreme wind/tornadoes.

Rationale: This feature ensures that the simulation accurately reflects how structures would respond to real-world forces, aiding in the assessment of their structural integrity.

Fit Criterion: Buildings within the simulation should realistically react to various natural disasters, demonstrating appropriate structural behaviors and responses.

Acceptance Tests: 2 (Game Loop)

ID#F-5: Challenge Mode

Description: Make-or-Break offers a Challenge Mode where users design and create structures to survive waves of challenges, both real-world and fantastical.

Rationale: Challenge Mode adds a gamified aspect to the application, encouraging strategic thinking and problem-solving in designing resilient structures.

Fit Criterion: Users should be able to access Challenge Mode, design structures within it, and receive feedback on their performance against challenges.

Acceptance Tests: 3 (Modes) F-2

ID#F-6: Sandbox Mode

Description: Make-or-Break includes a Sandbox Mode where users can design buildings without budget limitations or disaster constraints.

Rationale: Sandbox Mode provides users with the freedom to experiment with designs without constraints, fostering creativity and exploration.

Fit Criterion: Users should be able to access Sandbox Mode, design buildings freely, and save their creations.

Acceptance Tests: 3 (Modes)

11 Data Requirements

ID#D-1: User Account Data Secure and Encrypted

Description: User account data must be stored in a way, either through third-party OAuth providers, or other service providers such that leak of account data will not leak sensitive information.

Rationale: Confidence in secure storage of user data will encourage users to engage with the online services of the game.

Fit Criterion: Sensitive data must not be stored in the database, only relevant game data.

Acceptance Tests: 4 (User Data)

12 Performance Requirements

12a Speed and Latency Requirements

ID#P-1: Visible Frame Rate at Passable Threshold

Description: The rendering frame rate (not physics) should display above a certain threshold for animations, effects, and UI.

Rationale: Despite the focus on the game's systems rather than style, a low fps would provide a slow and clunky experience.

Fit Criterion: Visible frame rate must be at a minimum 30 FPS or above for most systems. 60 FPS would also make the game feel even smoother.

Acceptance Tests: 5 (Rendering)

12b Precision or Accuracy Requirements

ID#P-2: Maintain Physics Calculations Across Different Frame Rates

Description: Most games have specific solutions in order to account for physics calculations across different timesteps. Physics calculations must employ fixed physics timestep solutions.

Rationale: We want to achieve as much determinism as we can across different users' machines and performances.

Fit Criterion: The game's physics engine should employ fixed physics solutions consistently across different hardware configurations and frame rates, ensuring that the simulation behaves predictably and accurately on all platforms.

Acceptance Tests: 6 (Physics)

ID#P-3: Deterministic Physics Across Identical Conditions

Description: The physics engine used in the game if running with the exact same conditions and rules should have the same outcome.

Rationale: We want to achieve as much determinism for a realistic approach to physics.

Fit Criterion: The physics engine should demonstrate deterministic behavior across repeated simulations with identical initial conditions and inputs.

Acceptance Tests: 6 (Physics)

12c Capacity Requirements

ID#P-4: Simultaneous Players

Description: The server must be able to handle roughly 5,000 to 10,000 simultaneous players at any given time with peak hours being around 12pm - 8pm.

Rationale: The game has online features which allow players to access community created content and the server must be able to handle upload/download across hundreds if not thousands of users. This should not be a difficult task to do, considering the game runs on physics simulations in the client. The online features only consist of challenge attempt verification, and downloading/uploading challenges.

Fit Criterion: The server's performance metrics should consistently meet or exceed the specified player concurrency requirements during peak hours, with minimal latency and no service disruptions.

Acceptance Tests: 7 (Server Functionality)

13 Dependability Requirements

13a Reliability Requirements

ID#D-1: Crash Occurrence must be Below Certain Threshold

Description: The application shall have a maximum allowable occurrence of crashes, indicating the frequency at which the system can fail.

Rationale: To ensure confidence in the application's reliability for the users, we must define the maximum rate at which software failures are acceptable.

Fit Criterion: A crash should not occur in more than 25% of play sessions.

Acceptance Tests: 8 (Bugs/Errors)

13b Availability Requirements

ID#D-2: Online Server Availability

Description: The online server must be available for public access with a predictable schedule.

Rationale: It is unreasonable to expect 24/7 availability and reliability of public servers without downtime for maintenance and updates. The online availability of the game servers should be online most of the time, and if possible, on a predictable schedule.

Fit Criterion: The servers must be up for at-least 5-6 days a week.

Acceptance Tests: 7 (Server Functionality)

13c Robustness or Fault-Tolerance Requirements

ID#D-3: Local Play

Description: The levels downloaded by the users will still be playable on their system regardless of server connection.

Rationale: No access to the game servers should still let players access their levels downloaded on their system. These levels should have complete functionality besides features that need server validation to complete.

Fit Criterion: Any level installed on the user end should be functional to play with or without server connection.

Acceptance Tests: 2 (Game Loop)

13d Safety-Critical Requirements

ID#D-4: Photosensitive Warning

Description: The game should refrain/reduce any intense flashing of colors, lights and have a warning screen to notify players if by chance any can occur.

Rationale: A significant amount of users may experience epileptic seizures when exposed to certain color patterns or flashing lights, or experience photosensitivity in certain conditions. Our game has no requirement for such, so it would be a good requirement to satisfy in order to include more demographics in our player base.

Fit Criterion: Warning message must be displayed every time at game start.

Acceptance Tests: 9 (Safety/Accessibility)

14 Maintainability and Supportability Requirements

14a Maintenance Requirements

ID#M-1: Game Updates

Description: The user will be notified to update their game when a new version is released.

Rationale: Game updates will allow the users to try new features and levels added through new versions. Game updates may also bring any bug fixes from previous releases.

Fit Criterion: Before game launch, the game will check with the server if there are any updates available and notify the user to update their game in order to play.

Acceptance Tests: 7 (Server Functionality)

14b Supportability Requirements

ID#M-2: Bug reporting

Description: The game shall provide a way to report bugs whether it be directly from the game itself with automated bug reporting or manual bug reporting externally from a website.

Rationale: Users being able to report bug fixes allows for developers to fix any unforeseen bugs to improve the quality of the game for its users.

Fit Criterion: Depending on the severity of the bug, if a crash occurs should give the user an option to send a crash report. For smaller threat bugs, there should be a system that allows users to submit a form of the bug and the description of how to replicate the bug.

Acceptance Tests: 8 (Bugs/Errors)

14c Adaptability Requirements

ID#M-3: Operating Systems

Description: Make or Break is a PC game, so will be compatible with Windows, Mac-OS and Linux. Make or Break will be available from video game distribution services such as Steam and Epic Games.

Rationale: Make or Break being accessible on multiple OS' allows for a greater user base. Allowing Make or Break to be released on multiple game services will also allow it to be explored by more users.

Fit Criterion: Make or Break should run normal on all listed OS' and pass all checks to be available on Steam/Epic Games.

Acceptance Tests: 10 (Systems Requirements)

14d Scalability or Extensibility Requirements

ID#M-4: Player Base

Description: Make or Break should be able to support an influx of at least 10,000 players on launch or anytime during its lifespan.

Rationale: With the unpredictability of game popularity, Make or Break should be able to handle an increase of players at any given moment.

Fit Criterion: Server monitoring should be in place to track the current amount of players and other metrics such as active current players. Tests should also be in place to check network and server performance.

Acceptance Tests: 7 (Server Functionality)

14e Longevity Requirements

ID#M-5: Game Support Lifespan

Description: Make or Break should operate for a minimum of 2 years with a probable max of 5 years as to not exceed any budgets related to game upkeep.

Rationale: With Make or Breaks community features, the game should expect to operate and maintain upkeep to assist the user base with any problems that may arise and keep the game operating smoothly.

Fit Criterion: Users should be able to report bugs that arise during the game's lifespan to be fixed. There also should be a way for users to send feedback on things other than bugs for maintenance.

Acceptance Tests: 11 (Post-Release Support)

15 Security Requirements

15a Access Requirements

ID#S-1: User Accounts

Description: Personal information provided by users will not be shared by other users. Make or Break will only show the user's username.

Rationale: Make or Break's community integration allows users to create or use other user's creations. In order to use these features, the user must create a profile in which only their username will be shown to other users when publishing content.

Fit Criterion: Any profile information should only be accessible to the developers with the only public information being the user's username.

Acceptance Tests: 4 (User Data) 9 (Safety/Accessibility)

15b Integrity Requirements

ID#S-2: System Integrity

Description: Make or Break shall make sure that any information is protected from any authorized source.

Rationale: To prevent any sensitive data from being stolen or any data being corrupted.

Fit Criterion: Follow security protocols from cyber security experts to ensure system integrity.

Acceptance Tests: 4 (User Data) 7 (Server Functionality)

15c Privacy Requirements

ID#S-3: Terms and Conditions

Description: Make or Break shall provide the user to look over and agree to the terms and conditions before the users have any interaction with the game.

Rationale: In order to provide the user with explicit terms of the game and the assets' usage. This will provide a legal foundation for our game to stand as we plan on selling this.

Fit Criterion: Consult with legal team to ensure terms and conditions comply with current laws and regulations.

Acceptance Tests: 4 (Secure Practices)

15d Audit Requirements

None at this point in time.

15e Immunity Requirements

ID#S-4: Immunity to Malicious Software

Description: The system must implement security measures to protect itself from unauthorized or malicious software programs..

Rationale: It is important for the system to have built-in defenses to prevent unauthorized access, data breaches, and system compromise.

Fit Criterion: The system's security architecture should include features such as firewalls, intrusion detection systems, encryption protocols, and regular security updates to mitigate the risk of infection by malicious software.

Acceptance Tests: 4 (Safe Practices) 9 (Safety/Accessibility)

16 Usability and Humanity Requirements

16a Ease of Use Requirements

ID#U-1: User-Friendly Interface

Description: The system's user interface (UI) should be intuitive and easy to navigate, allowing users to accomplish tasks efficiently and effectively.

Rationale: A user-friendly interface enhances user satisfaction, reduces training time, and minimizes errors, ultimately leading to improved user enjoyment and engagement.

Fit Criterion: The UI design should adhere to usability principles, including efficiency of use, ease of remembering, low error rates, overall user satisfaction, and appropriate feedback mechanisms.

Acceptance Tests: 9 (Safety/Accessibility)

16b Personalization and Internationalization Requirements

ID#U-2: Customization Options

Description: The system should offer users customization options allowing them to tailor their experience and adapt the system to different cultural and regional preferences.

Rationale: Providing customization features enhances user satisfaction and inclusivity by accommodating diverse linguistic, cultural, and regional aspects.

Fit Criterion: The system interface should include configurable settings menus enabling users to personalize language, currency, units of measurement, date/time formats, and other relevant options. These settings should be seamlessly applied across the system.

Acceptance Tests: 10 (Systems Requirements) 9 (Safety/Accessibility)

16c Learning Requirements

ID#U-3: Tutorial

Description: A tutorial level is introduced when the user first plays the game to understand the tools and features of the game.

Rationale: Tutorial level helps the user familiarize themselves with the tools and features of the game and how to use them effectively.

Fit Criterion: The tutorial should introduce the user with each main feature/tool and provide a check that makes sure they interacted with each step before being able to complete the tutorial level.

Acceptance Tests: 2 (Game Loop) 9 (Safety/Accessibility)

16d Understandability and Politeness Requirements

ID#U-4: Application Understandability

Description: Make or Break shall use simple symbols that are most common in other devices and applications that user's are familiar with accompanied by simple words.

Rationale: Using simple and familiar words/symbols allows for the users to have a more user-friendly experience which they can better understand what they are doing within the application.

Fit Criterion: Use of global symbols such as a home icon to represent the main page or the erase icon to present the deletion of a selected item, etc..

Acceptance Tests: 9 (Safety/Accessibility)

16e Accessibility Requirements

ID#U-5: Accessibility Features

Description: The system should incorporate features and design considerations to ensure accessibility for individuals with common disabilities, including physical disabilities, visual impairments, hearing impairments, cognitive challenges, and other abilities.

Rationale: Ensuring accessibility for individuals with disabilities not only fulfills legal requirements but also promotes inclusivity and allows a wider range of users to access and benefit from the product.

Fit Criterion: The system's accessibility features should adhere to established accessibility standards and guidelines, such as relevant legal requirements in the jurisdiction of use. These features should enable users with disabilities to navigate, interact with, and understand the product effectively.

Acceptance Tests: 9 (Safety/Accessibility)

16f User Documentation Requirements

ID#U-6: User Documentation

Description: This entails compiling a comprehensive list of user documentation to be provided as part of the product package.

Rationale: User documentation is crucial for assisting users in understanding, installing, operating, and maintaining the product effectively.

Fit Criterion: The user documentation list should encompass all necessary documents for different user needs and scenarios, including user manuals, technical specifications, installation guides, and service manuals.

Acceptance Tests: 9 (Safety/Accessibility)

16g Training Requirements

ID#U-7: Interface Help Option

Description: An interface help option should be present to help users understand the different sections of the game what each section entails..

Rationale: An explanation of each section will allow the users to use the game to its full potential.

Fit Criterion: The interface help option would guide users through each button and option on the interface and a description of the purpose of each section.

Acceptance Tests: 9 (Safety/Accessibility)

17 Look and Feel Requirements

17a Appearance Requirements

ID#L-1: Visual Design

Description: The appearance should be friendly to the user, simple, and appealing. The fonts throughout the interface should be easy to read and simple to fit the user interface. Colors of choice will be white and lighter colors, and darker colors will be avoided to make it easy for the client to navigate the application. The game logo must be present in the game. The appearance of the game should represent a sense of engineering, construction, and builder theme.

Rationale: The appearance guidelines prioritize user experience and friendliness. Consumers can easily traverse the UI because of its simple design appeal.

Fit Criterion: The appearance requirements of the game should allow users to locate and navigate through the UI seamlessly so consumers can focus on the game.

Acceptance Tests: 9 (Safety/Accessibility), 12 (Production Workflow), (Font legibly test), (logo presence test) (user feedback).

17b Style Requirements

ID#L-2: Simplified Construction and Architecture Style

Description: Style of game assets must be construction/architecture themed. Should address the impression of professionalism in our clients. Should be simple and easily read by the player, but not too realistic.

Rationale: Making our players feel professional and allowing their thoughts to fit the games helps create a seamless and engaging experience for players.

Fit Criterion: Use the same workflow and process for creation of models, textures, and graphics.

Considerations: A good way to do this would be to create a style and design document for the game that production can refer to.

Acceptance Tests: 12 (Production Workflow)

18 Operational and Environmental Requirements

18b Requirements for Interfacing with Adjacent Systems

Not any at this time.

18c Productization Requirements

ID#O-1: Distribution of Application

Description: Make-Or-Break application must be distributed online through various platforms to our playerbase.

Rationale: Games are typically distributed through websites or platforms such as Steam, or the App Store. These also handle deploying updates and

patches, as well as handle the distribution of production/marketing of the game.

Fit Criterion: Verify that we satisfy the requirements, terms and conditions, and policies required to publish on the platforms needed.

Acceptance Tests: 11 (Post-Release Support)

18d Release Requirements

ID#O-2: Release Cycle

Description: This specifies the intended release cycle for the product, including the frequency of major updates, maintenance releases, and patches.

Rationale: Establishing a clear release cycle helps manage stakeholder expectations, ensures timely delivery of updates and improvements, and facilitates effective planning and coordination across development and deployment teams.

Fit Criterion: The release cycle should be documented, detailing the frequency of major releases, minor updates, and patches. The versioning strategy should specify how version numbers will be incremented for each release type.

Acceptance Tests: 11 (Post-Release Support)

19 Cultural and Political Requirements

19a Cultural Requirements

ID#C-1: Multi-Cultural Sensitive Game Assets

Description: Game assets, including characters, settings, and narrative elements, should be designed with cultural sensitivity and diversity in mind, representing a wide range of cultures and communities in a respectful manner.

Rationale: Our game should hold respect for many different cultures and communities as well as player demographics.

Fit Criterion: A sensitivity consultant could determine whether or not our game satisfies sensitivity requirements for games.

Acceptance Tests: 9 (Safety/Accessibility)

19b Political Requirements

ID#C-2 - Non-Political Iconography

Description: Any icons in game assets must have placeholder or neutral content.

Rationale: The game should depict or show any alignment to any ideology in regards to politics and culture.

Fit Criterion: Possible to refer to a professional sensitivity consultant to look into it, however, employing Q/A and Testing companies would be a more fitting role to test this.

Acceptance Tests: 9 (Safety/Accessibility)

20 Legal Requirements

20a Compliance Requirements

ID#L-1: Data Protection Compliance

Description: Ensure that the Make-or-Break system handles personal data in compliance with relevant data protection laws and regulations.

Rationale: Protecting user privacy and data security is essential for maintaining trust and legal compliance.

Fit Criterion: The Make-or-Break system is designed and implemented with data protection principles in mind, including data minimization, and user consent mechanisms.

Acceptance Tests: 4 (Secure Practices) 9 (Safety/Accessibility)

20b Standards Requirements

ID#L-2: Agile Software Development Methodology

Description: The Make-or-Break system shall be developed according to Agile software development principles and practices.

Rationale: Agile methodologies promote iterative development, collaboration, and flexibility, allowing for quicker response to changes in requirements and market demands.

Fit Criterion: The Make-or-Break development team follows Agile principles such as iterative development, continuous feedback, and adaptive planning throughout the software development lifecycle.

Acceptance Tests: 12 (Development)

21 Requirements Acceptance Tests

21a Requirements - Test Correspondence Summary

| Test | Req 1 | Req 2 | Req 3 | Req 4 | Req 5 | Req 6 | Req 7 | Req 8 | Req 9 | Req 10 | Req 11 | Req 12 | Req 13 | Req 14 | Req 15 | Req 16 | Req 17 | Req 18 | Req 19 | Req 20 |
|---------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Test 1 | | X | | | | | | | | | | | | | | | | | | |
| Test 2 | X | | X | X | | | | | | | | | | X | | | | | | |
| Test 3 | | | | | X | X | | | | | | | | | | | | | | |
| Test 4 | | | | | | | X | | | | | | | | | | | | | |
| Test 5 | | | | | | | | X | | | | | | | | | | | | |
| Test 6 | | | | | | | | | X | X | | | | | | | | | | |
| Test 7 | | | | | | | | | | | X | | X | | | X | | | X | |
| Test 8 | | | | | | | | | | | | X | | | | | X | | | |
| Test 9 | | | | | | | | | | | | | | | X | | | | | |
| Test 10 | | | | | | | | | | | | | | | | | | X | | |
| Test 11 | | | | | | | | | | | | | | | | | | | | X |
| Test 12 | | | | | | | | | | | | | | | | | | | | |
| Test 13 | | | | | | | | | | | | | | | | | | | | |
| Test 14 | | | | | | | | | | | | | | | | | | | | |
| Test 15 | | | | | | | | | | | | | | | | | | | | |

| | | Req 21 | Req 22 | Req 23 | Req 24 | Req 25 | Req 26 | Req 27 | Req 28 | Req 29 | Req 30 | Req 31 | Req 32 | Req 33 | Req 34 | Req 35 | Req 36 | Req 37 | Req 38 | Req 39 | Req 40 |
|----|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 41 | | | | | | | | | | | | | | | | | | | | | |
| 42 | Test 1 | | | | | | | | | | | | | | | | | | | | |
| 43 | Test 2 | | | | | | | X | | | | | | | | | | | | | |
| 44 | Test 3 | | | | | | | | | | | | | | | | | | | | |
| 45 | Test 4 | X | | X | | | | | | | | | | | | | | | X | | |
| 46 | Test 5 | | | | | | | | | | | | | | | | | | | | |
| 47 | Test 6 | | | | | | | | | | | | | | | | | | | | |
| 48 | Test 7 | | X | | | | | | | | | | | | | | | | | | |
| 49 | Test 8 | | | | | | | | | | | | | | | | | | | | |
| 50 | Test 9 | | | X | X | | | | X | X | X | X | | | | | X | X | | | |
| 51 | Test 10 | | | | | | X | | | | | | | | | | | | | | |
| 52 | Test 11 | | | | | | | | | | | | | | | | | | | | |
| 53 | Test 12 | | | | | | | | | | | | X | X | X | X | | | | X | |
| 54 | Test 13 | | | | | | | | | | | | | | | | | | | | |
| 55 | Test 14 | | | | | | | | | | | | | | | | | | | | |
| 56 | Test 15 | | | | | | | | | | | | | | | | | | | | |

Table 3 - Requirements - Test Correspondence Summary

21b Acceptance Test Descriptions

All Requirements Tests with brief descriptions:

ID#F-1: View Campaign Challenges

Description: Provide escalating challenge levels for user progression.

ID#F-2: View Online Published Challenges

Description: List of descriptions and previews for challenges.

ID#F-3: Building Creation

Description: Create, edit, and load recreations of buildings and structures

ID#F-4: Real-Time Physics Simulation

Description: Real-time physics simulations to test virtual building designs against forces of nature.

ID#F-5: Challenge Mode

Description: Users design and create structures to survive waves of challenges.

ID#F-6: Sandbox Mode

Description: Users can design buildings without budget limitations.

ID#D-1: User Account Data Secure and Encrypted

Description: User data must be stored such that leak of account data will not leak sensitive information.

ID#P-1: Visible Frame Rate at Passable Threshold

Description: Frame rate should display above a certain threshold.

ID#P-2: Maintain Physics Calculations Across Different Frame Rates

Description: Calculations must employ fixed physics timestep solutions.

ID#P-3: Deterministic Physics Across Identical Conditions

Description: Running with the exact same conditions and rules should have the same outcome.

ID#P-4: Simultaneous Players

Description: Server must be able to handle roughly 5,000 to 10,000 simultaneous players

ID#D-1: Crash Occurrence must be Below Certain Threshold

Description: Application shall have a maximum allowable occurrence of crashes

ID#D-2: Online Server Availability

Description: The online server must be available for public access.

ID#D-3: Local Play

Description: Levels downloaded by the users will still be playable on their system regardless of connection.

ID#D-4: Photosensitive Warning

Description: Refrain/reduce any intense flashing of colors, lights and have a warning screen to notify players

ID#M-1: Game Updates

Description: User will be notified to update their game when a new version is released

ID#M-2: Bug reporting

Description: Provide a way to report bugs automatically from the game itself or manually.

ID#M-3: Operating Systems

Description: PC game, so needs to be compatible with Windows, Mac-OS and Linux.

ID#M-4: Player Base

Description: Support an influx of at least 10,000 players on launch.

ID#M-5: Game Support Lifespan

Description: Make or Break should operate for a minimum of 2 years.

ID#S-1: User Accounts

Description: Personal information provided by users will not be shared by other users.

ID#S-2: System Integrity

Description: Make sure that any information is protected from any authorized source.

ID#S-3: Terms and Conditions

Description: Look over and agree to the terms and conditions before users play the game.

ID#S-4: Immunity to Malicious Software

Description: System must implement security measures to protect itself from unauthorized access.

ID#U-1: User-Friendly Interface

Description: UI should be intuitive and easy to navigate.

ID#U-2: Customization and Globalization Options

Description: System should offer users customization options and support globalization.

ID#U-3: Tutorial

Description: Tutorial level is introduced when the user first plays the game.

ID#U-4: Application Understandability

Description: Use simple symbols that are most common in other devices and applications.

ID#U-5: Accessibility Features

Description: The system should incorporate features and design considerations to ensure accessibility for individuals with common disabilities.

ID#U-6: User Documentation

Description: Comprehensive list of user documentation to be provided as part of the product.

ID#U-7: Interface Help Option

Description: Help option should be present to help users understand the different sections of the game.

ID#L-1: Visual Design

Description: The appearance should be friendly to the user, simple, and appealing.

ID#L-2: Simplified Construction and Architecture Style

Description: Style of game assets must be construction/architecture themed.

ID#O-1: Distribution of Application

Description: Application must be distributed online through various platforms.

ID#O-2: Release Cycle

Description: Intended release cycle for the product, including the frequency of major updates

ID#C-1: Multi-Cultural Sensitive Game Assets

Description: Game assets, including characters, settings, and narrative elements, should be designed with cultural sensitivity and diversity in mind.

ID#C-2 - Non-Political Iconography

Description: Any icons in game assets must have placeholder or neutral content.

ID#L-1: Data Protection Compliance

Description: System handles personal data in compliance with relevant laws.

ID#L-2: Agile Software Development Methodology

Description: System shall be developed according to Agile software development principles and practices.

III Design

22 Design Goals

The design of Make-or-Break aims for a balance between realism and entertainment. Keeping this in mind the main design objectives for Make or break are:

- **Physics Simulations:** Realistic depictions of natural forces on structures
- **Scalability:** Wide range of building sizes / varying magnitudes of disasters
- **Performance Optimization:** Smooth and responsive simulations on varying hardware
- **Education Value:** Introduces engineering principles, and disaster preparedness / awareness

23 Current Design System

There are no pre existing systems for Make or Break currently on the market, although there are games with similar aspects which are similar to what Make-or-Break is trying to accomplish currently available such as the previously mentioned: Cities: Skylines, Poly Bridge, and Space Engine.

24 Proposed System Design

24a Initial System Analysis and Class Identification

After performing grammatical analysis on the following, “Make-or-Break allows users to create, edit, and load real-world recreations of buildings and structures in the 3D environment. Users will be able to test their creations against the forces of nature, such as torrential rain and flooding, earthquakes, and extreme wind/tornadoes...” we determined that the following diagram closely summarized our application’s main objects and relationships at the topmost level.

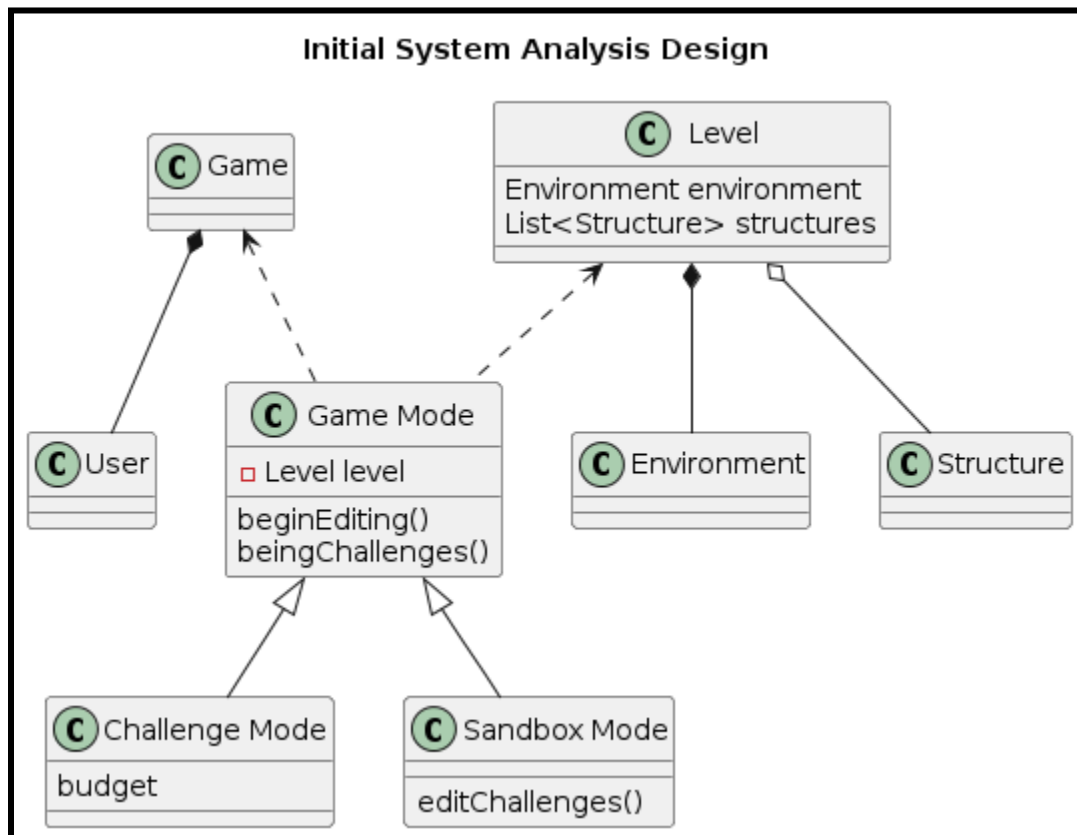


Figure #5 - Initial System Design

24b Dynamic Modeling of Use-Cases

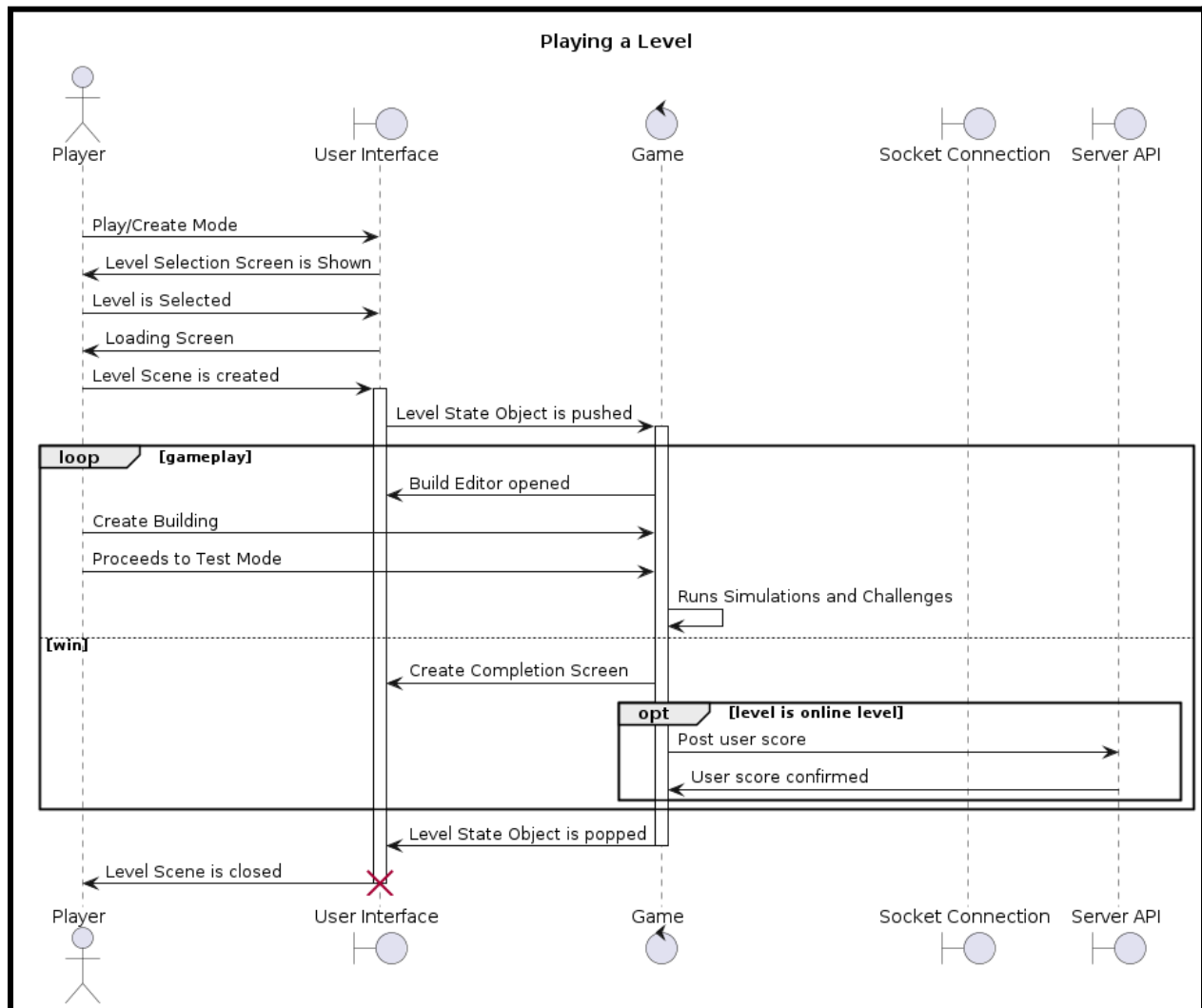


Figure #6 - Sequence Diagram #1

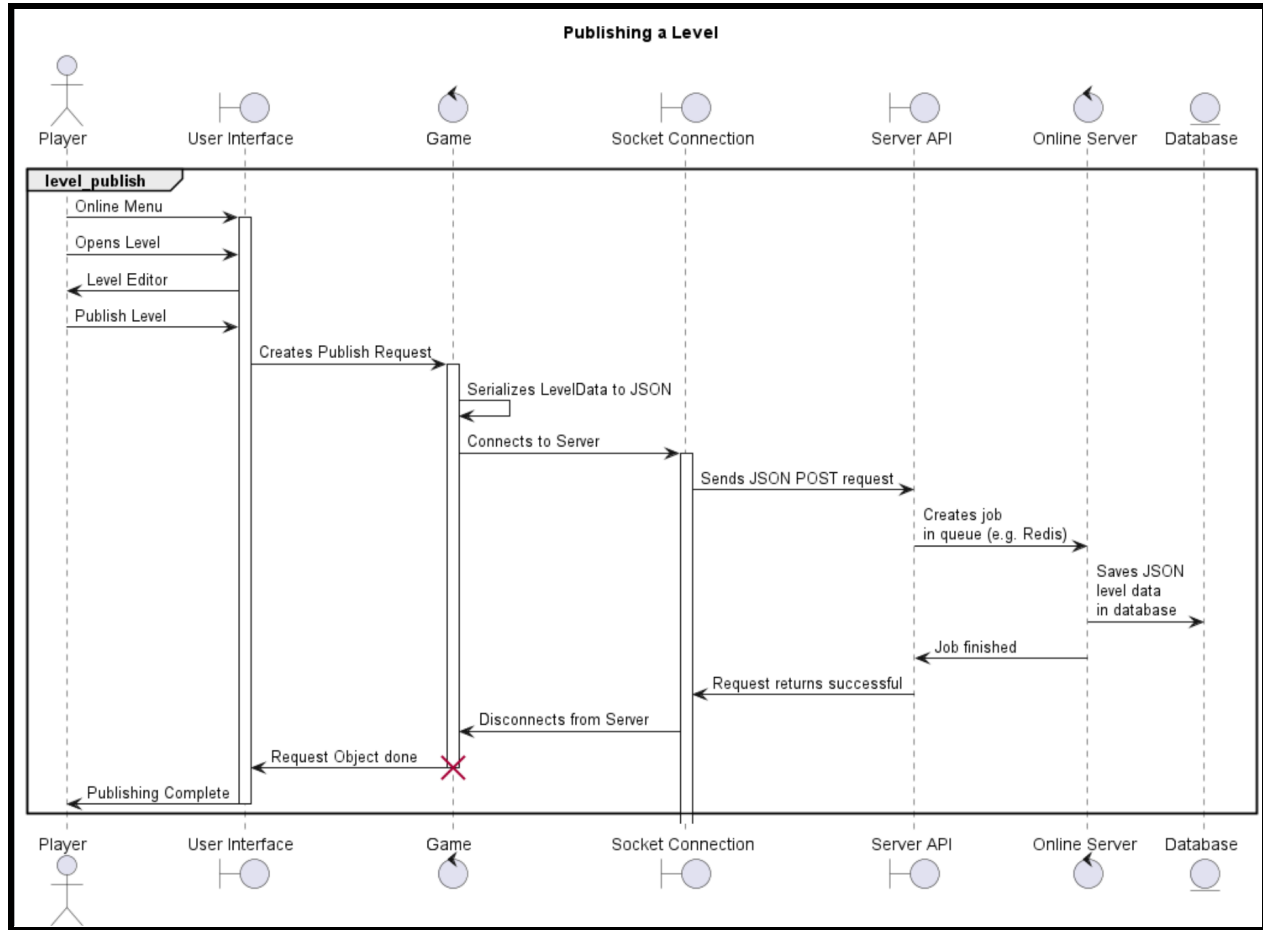


Figure #7 - Sequence Diagram #2

24c Proposed System Architecture

For the "Make-or-Break" application, a suitable software architecture would be the Model-View-Controller (MVC) architecture.

1. **Model:** The model represents the core data and logic of the application. For "Make-or-Break," the model would encompass the physics engine, which simulates real-world forces acting on the structures, as well as the database or data structures holding information about buildings, challenges, and user progress.
2. **View:** The view layer handles the presentation of the data to the user. In this context, the view would be the 3D environment where users can see their structures, interact with the simulation, and receive feedback on how their designs fare against natural disasters. It would also include the user

interface elements for creating, editing, and navigating through the application.

3. **Controller:** The controllers would manage user interactions such as creating and editing structures, initiating simulations, and responding to events like challenges or disasters. They would also orchestrate communication between the model and view components.

24d Initial Subsystem Decomposition

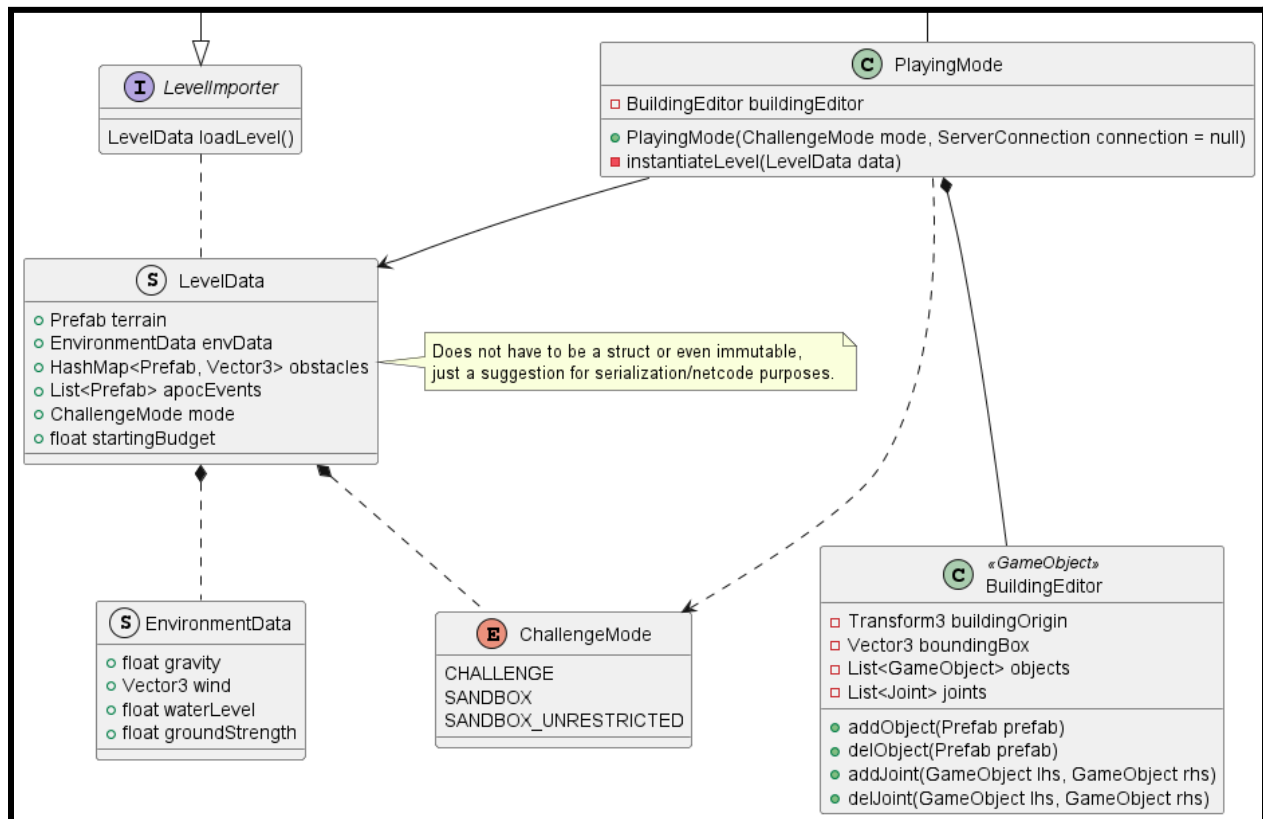


Figure #8 - Class Diagram

25 Additional Design Considerations

25a Hardware / Software Mapping

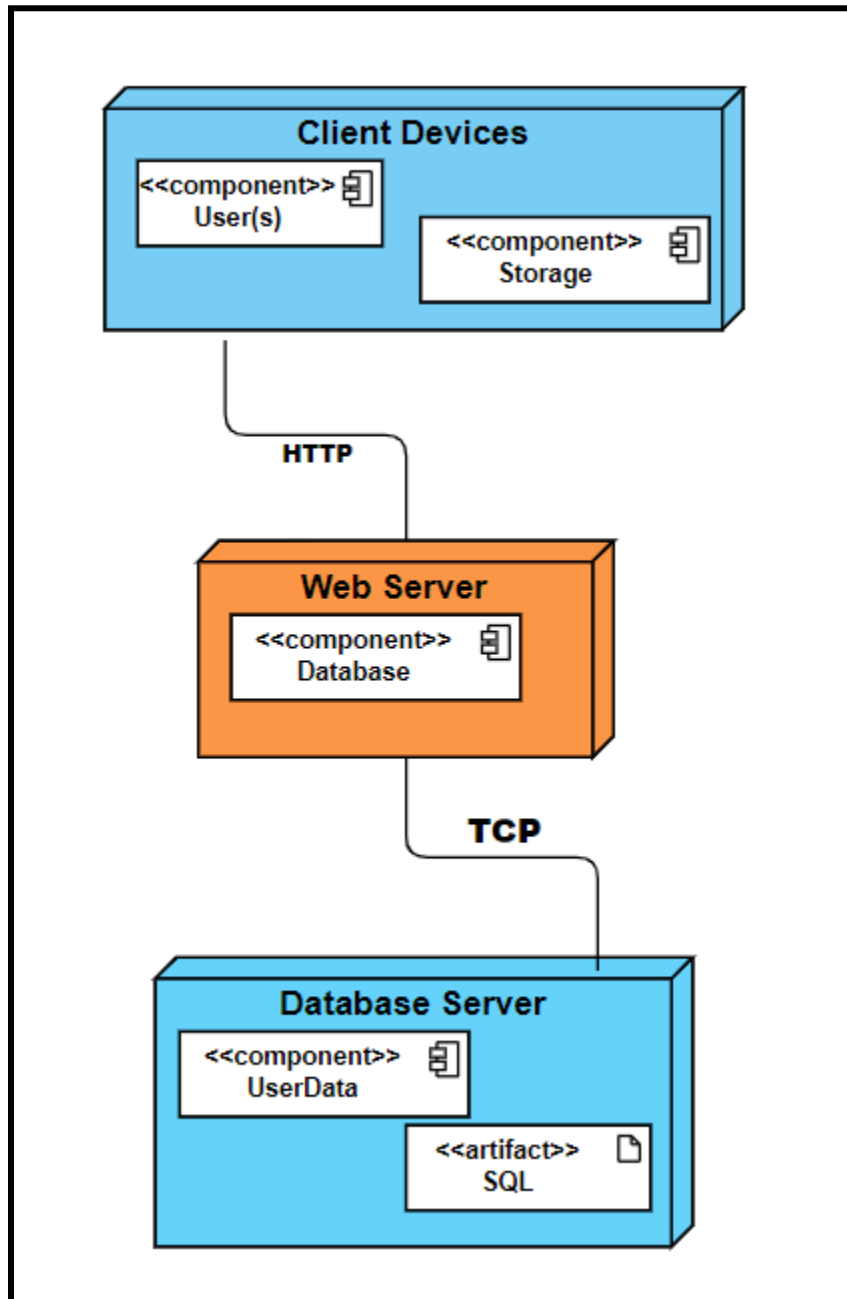


Figure #9 - Deployment Diagram

25b Persistent Data Management

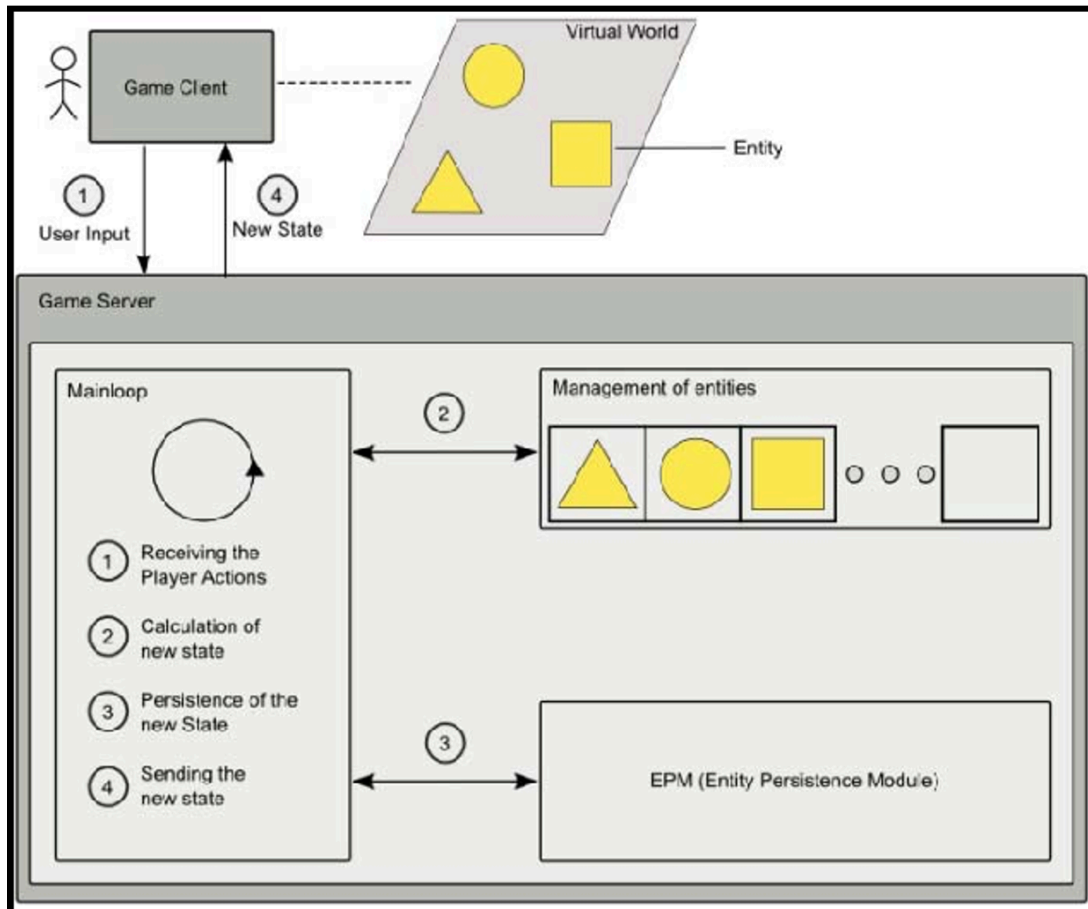


Figure #10 - Data storage design for real-time interactive game

25c Access Control and Security

With games that also employ features such as ranking and score-keeping online, there are vulnerabilities that must be addressed. A malicious actor may be able to send any sort of fake information to the API's servers. This should be handled in a graceful manner, and to be expected when hosting publicly accessed online functionality.

A solution to being sent cheated data could be 1, eliminate the ability for users to directly input in data to the server's database. Instead, a concurrent game could run on the server that will be then interfaced to the player. One flaw about this implementation is the budget, as this severely increases the hosting costs and scaling per player as each player needs to have an online "match" hosted for them

to submit any scores. This does solve the problem of verifying data by only accepting data sent to the master records by legitimate sessions of the game.

Another solution which has not been commonly accepted is leveraging the deterministic physics engine property to our advantage. Since deterministic physics allows for identical results to appear from different machines and runtimes, the server could run a multitude of ways to verify this. The server could run a simulation of the starting conditions and properties and see whether or not it results in the same data sent by the player. This would be a lot more performant than concurrent games that rely on tandem communication between the client, the master server, and/or a containerized server. Developers could also look into researching ways of verifying the data directly through modular arithmetic, checksums, and more which would be a significant decrease in cost.

We also must keep in mind security related issues concerning user data. A few ways to maintain secure data compliance would be to store data that is only required by the user. Our user database only needs to store a minimal amount of data, such as usernames/ids, display names, completion of levels, and created levels. Other things such as profile picture, real names, device information, etc. are either unnecessary, or can be linked to other providers.

Another solution that would solve most issues would be to use a third-party OAuth provider or other user authentication such as the Steamworks API. This would mean our database does not need to store any sensitive information. With the Steamworks API, players can also easily link with other features such as achievements, a friends list, or workshop.

25d Global Software Control

Performance, Maintainability, and Scalability are very important with the development of games especially if they rely on the server to be functional. There is a possibility that an overload of users can affect the Scalability of the game. To keep maintainability as simple and efficient as possible, organizing our codebase is the solution for avoiding maintenance problems like bugs or system updates sent to users. Performance of the game can be affected by the slow algorithms or bad data indexing. If the user's machine is not up to par with the game's specifications those users might have to deal with low performance when opening the system. However these problems can be dealt with by using efficient and low runtime algorithms as well as caching. Users who don't have access to machines that can't handle our system can access it via cloud gaming services.

25e Boundary Conditions

During the initial startup of the game, the server loads up smoothly and loads up game assets from the local storage. If the user can't connect to the server, the system will load up the user's most recent progress from the local storage or database. When an update arrives for a new version of the game or bug fixes, the system needs to connect to the server and restart smoothly once the game assets have been updated from the server. The class and subsystem responsible for this is the game startup manager. When the system receives a signal to shut down, the game system will save or store changed files in the local storage before preparing for shutdown. Saves to the game state will be processed to a safe and efficient game storage. For any abnormal system shutdown due to system errors or failures, the game server will save the player's progress and any other important files before shutting down. The class game shutdown manager will notify the user of what has happened and the reason for the shutdown. Lastly, the games data manager class will save player progress, game progress, and any relevant data into a database under the player's system account.

25f User Interface



Figure #11 : Home Screen Mockup



Figure #12 : Gameplay Mockup

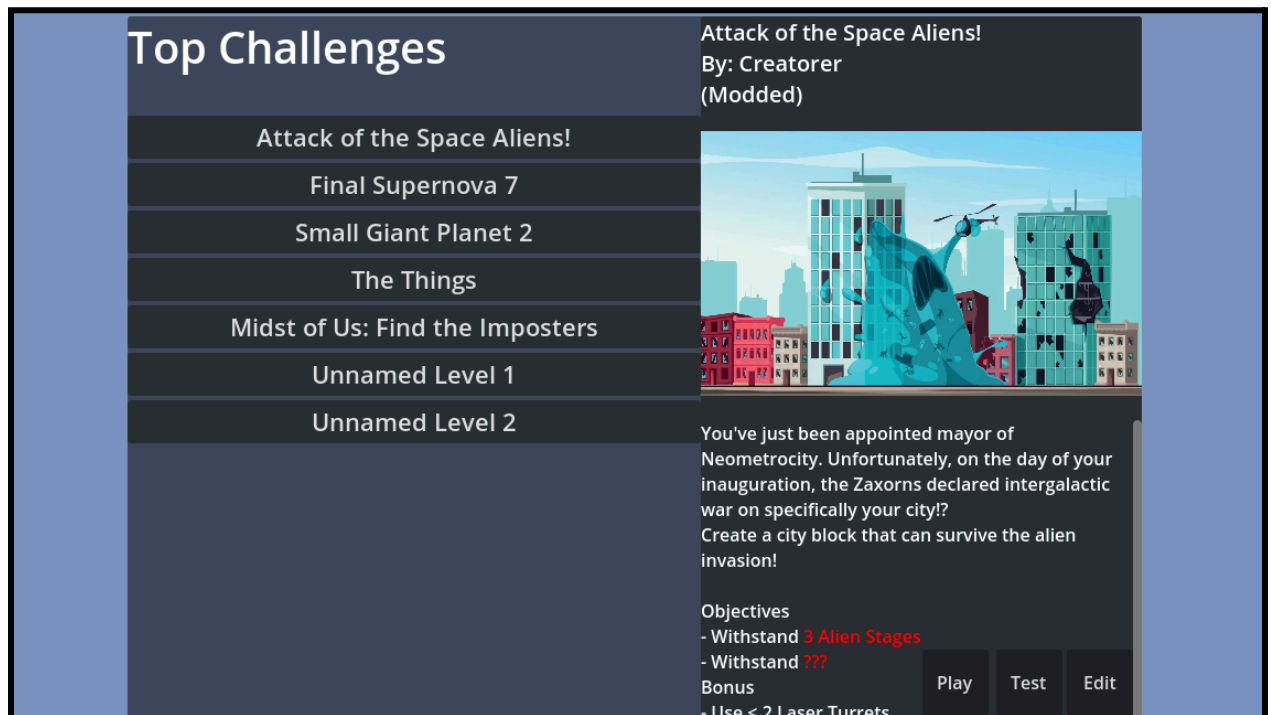


Figure #13 : Challenge Menu Mockup

25g Application of Design Patterns

Finite State Machine design pattern for game modes:

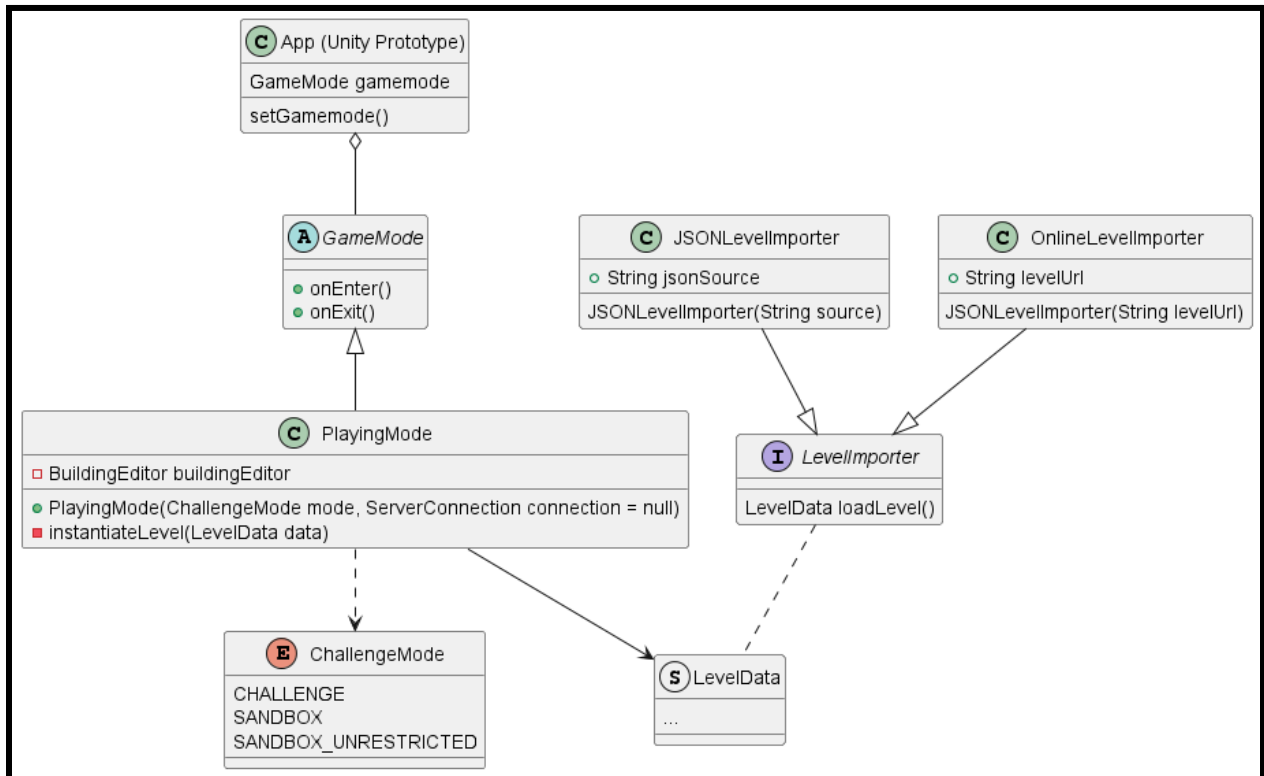


Figure #14 - Unity Prototype Design

Proxy for the server connection:

Service Locator

Aggregation/Composition: for game objects and entities

Factory for importing levels

26 Final System Design

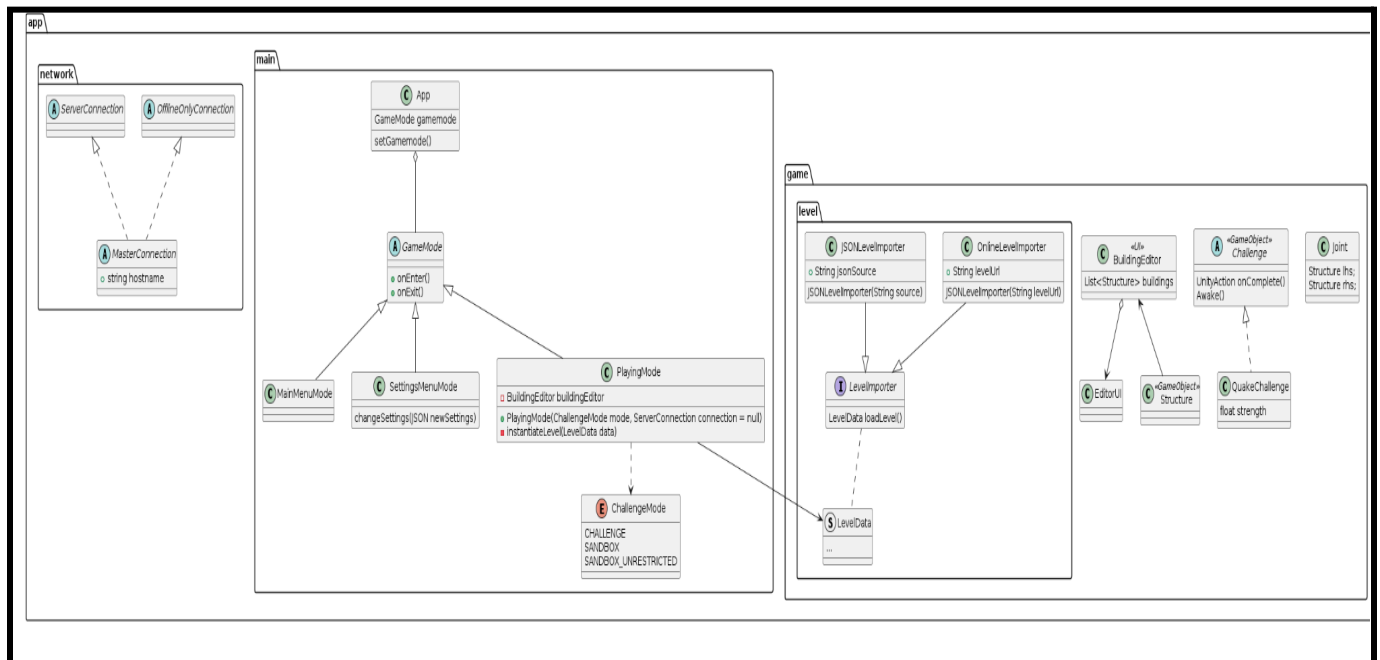


Figure #15 - Overall System Design

27 Object Design

27a Packages

App:

Root namespace related to the application, holds all other application related code and assembly.

App.main:

Entry point and classes related to most of the application's entry point and main systems.

App.game:

Package related to assets and game state inside the game, and other active entities.

App.network:

Package related to online functionality, connecting, and handling for lack of network access.

App.game.level:

Sub-package of app.game related to creating and instantiation of levels and environments.

27b Subsystem I

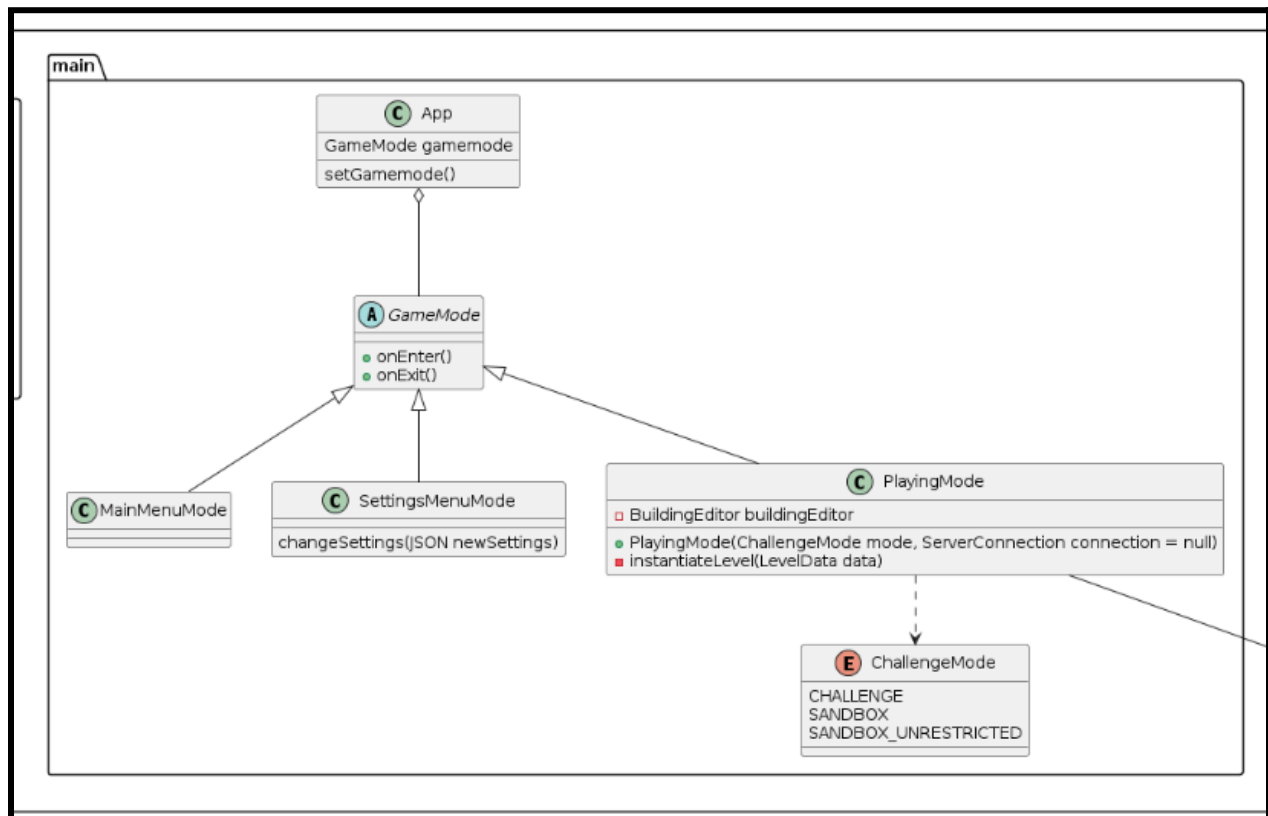


Figure #16 - Overall System Design (App.main)

27c Subsystem II

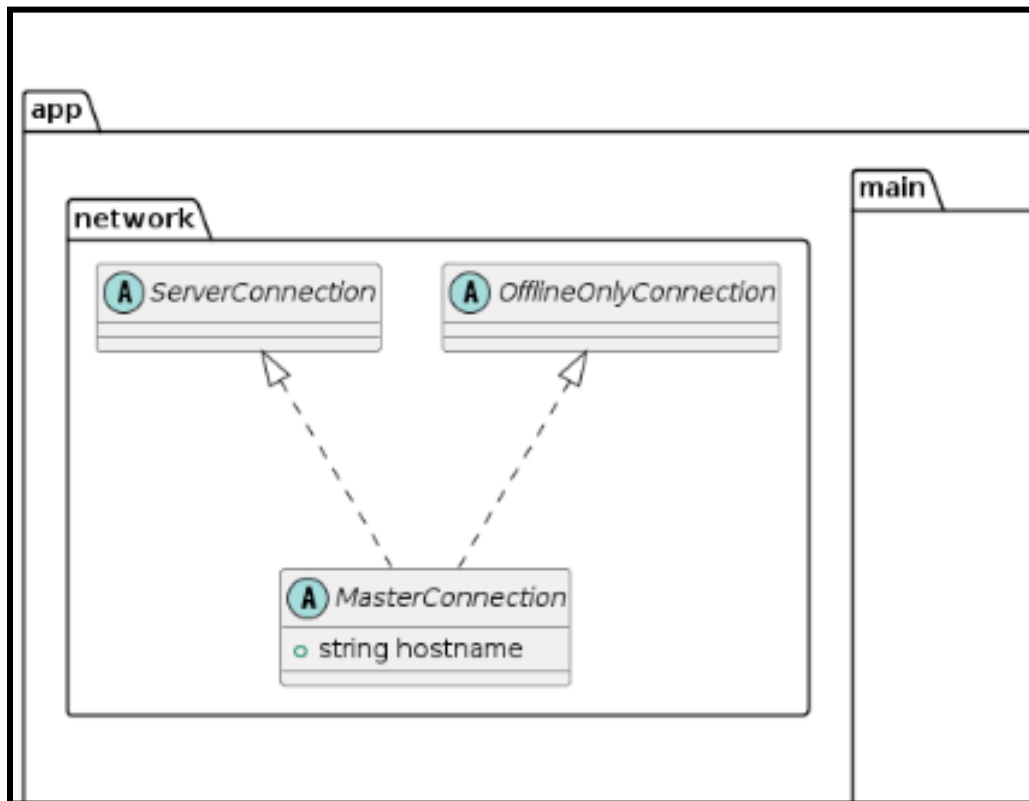


Figure #17 - Overall System Design (App.network)

28d Subsystem III

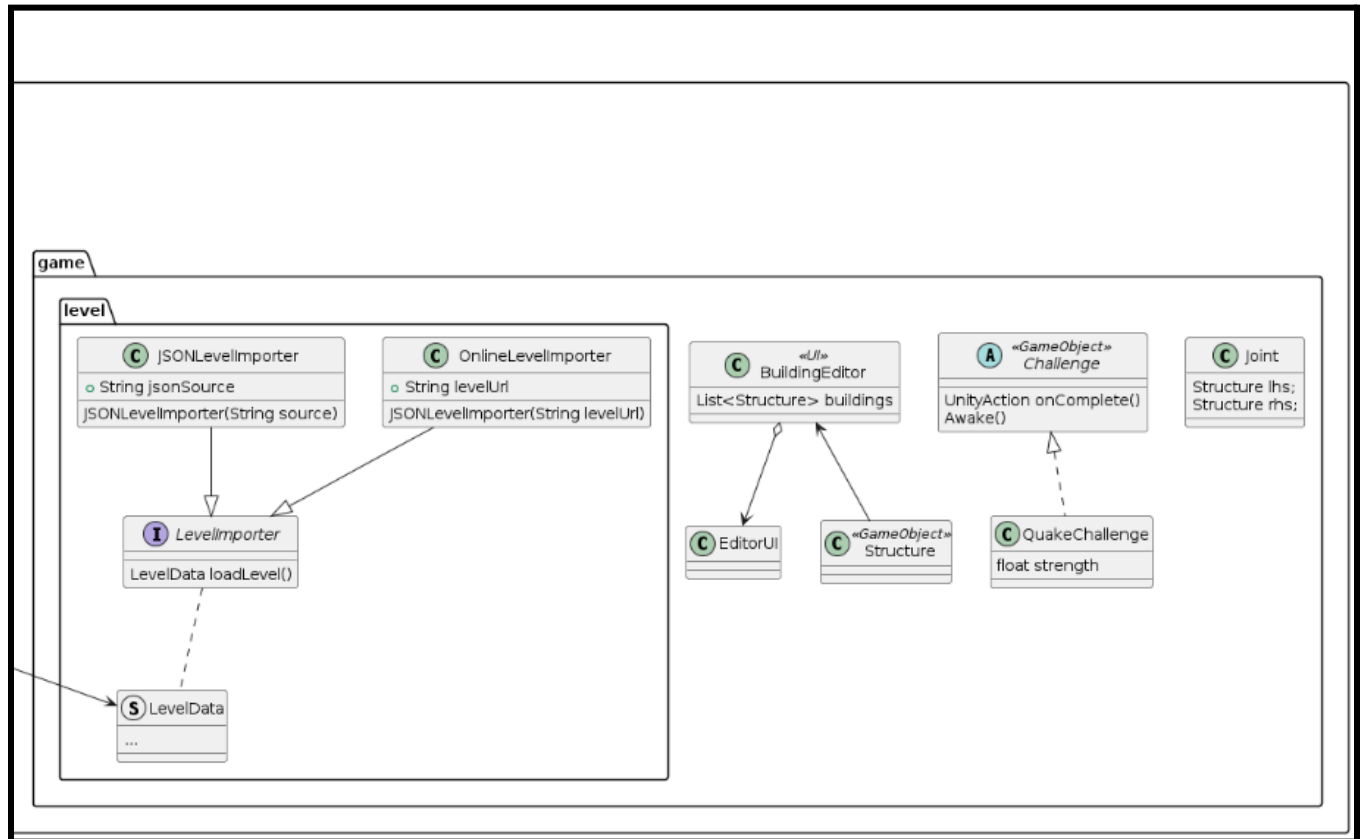


Figure #18 - Overall System Design (App.game.level)

IV Project Issues

28 Open Issues

Issues that Make or Break could face upon launch were determined to be:

- **Hardware Compatibility:** Consistent performance across varying hardware configurations
- **Resource Constraints:** Limited budgets, time or personnel
- **Technological Dependencies:** Reliance on third party technologies for physics simulations, rendering, multiplayer support, etc.
- **User Feedback:** Affects future updates, features, marketing strategies, etc.

29 Off-the-Shelf Solutions

29a Ready-Made Products

Existing products and components like Unity3D Engine and Unreal Engine offer solutions for hardware compatibility, providing cross-platform support and optimization tools. Asset store assets and freelance services can help alleviate resource constraints by offering pre-made assets and access to skilled professionals. Technologies like PhysX and Photon Engine reduce technological dependencies by providing advanced physics simulations and multiplayer networking solutions. Additionally, user feedback platforms and analytics tools offer insights into user preferences and behavior, aiding in future updates and marketing strategies. These existing options enable developers to address key challenges efficiently and effectively, potentially saving time and resources in the development process.

29b Reusable Components

Reusable components for use in Make or Break development:

- Physics libraries such as Box2D and Bullet Physics for physics simulation.
- Networking libraries such as Lidgren.Network and Mirror Networking for multiplayer functionality.
- OpenGL for graphic rendering.
- UI frameworks like ImGui and Dear ImGui offer efficient solutions for creating user interfaces.

By leveraging these existing libraries and toolkits, Make or Break will benefit from accelerated development, reduced complexity, and improved quality without every component having to be developed from the ground up.

29c Products That Can Be Copied

Make or Break could draw heavy inspiration from the very popular Bloons Tower Defense franchise. In their latest games, they have included a sandbox mode which offers players a similar level of freedom and experimentation. Make or Break can emulate a similar design in which a player can come up with endless combinations of materials, designs, and disaster simulation.

Here are some aspects that Make or Break could potentially emulate:

- **Unlimited Resources:** Unlimited resources to build and test structures without financial constraints.
- **Customizable Settings:** Allows players to customize various parameters such as disaster frequency, intensity, and terrain conditions to create unique scenarios and challenges.
- **Endless Waves or Challenges:** Provides players with continuous gameplay and the opportunity to refine their strategies over time.

By adopting these elements from Bloons Tower Defense's sandbox mode, Make or Break can provide players with a dynamic sandbox experience where creativity and experimentation are encouraged.

30 New Problems

30a Effects on the Current Environment

The implementation and maintenance cost could fluctuate heavily depending on the terms of software licenses, hardware infrastructure, and training programs. Additionally, ongoing maintenance and updates to the system may incur additional costs over time, which need to be carefully managed to ensure the long-term viability of the solution.

Also, users might spend too much time playing the "Make-or-Break" game due to its engaging gameplay and nature. The ability of designing, testing, and refining virtual structures against various natural disasters could lead to extended gaming sessions as users strive to pass increasing levels.

Additionally, the challenge mode and sandbox mode offer endless possibilities for experimentation and creativity, enticing users to invest significant time in perfecting their building designs.

30b Effects on the Installed Systems

Make-or-Break may potentially have adverse effects on existing hardware or software systems, particularly if it requires integration with or access to other systems within the organization. For example, if the "Make-or-Break" system relies on specific hardware configurations or operating systems that are incompatible with existing infrastructure, it could lead to compatibility issues and disruptions in workflow.

30c Potential User Problems

Users of the software may potentially have a negative response to the "Make-or-Break" system if they encounter usability challenges, such as significant changes to the system or UI, leaving some users may feel frustrated or resistant to adopting the new system. Also, if the game has too steep a learning curve or difficulty when it comes to completing levels / tasks that could leave user's with a frustrating experience as they strive to improve at the game. Furthermore, if the system's performance is subpar, leading to lag or glitches during simulations, users may become frustrated and lose interest in the game.

30d Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

Limitations that could inhibit effectiveness: if the system requires too high of a level of computational power for real-time physics simulations, but the existing hardware infrastructure lacks sufficient processing capabilities, it may lead to performance bottlenecks and degraded user experience.

Understanding and addressing these potential limitations early on is crucial for ensuring the successful implementation and performance of the system in its intended environment / configuration.

30e Follow-Up Problems

One problem is an inability to scale up the product if the user base becomes too large too quickly. Inability to play immediately because of lack of capacity, or system instability caused by too many users attempting to play the game at once, may cause users to lose interest and move on to different games.

Inversely, if the player base's attention or interest is unable to be held, users may lose interest quickly after trying out all existing features and modes. This requires new content to be added through updates on a timely basis in order to retain user interest.

31 Migration to the New Product

Not Applicable to Make or Break at this point in time.

32 Risks

1. **Inaccurate Physics Simulations:**
 - a. **Probability:** Medium
 - b. **Impact:** Inaccuracies in the physics simulation engine leading compromises the effectiveness of structural testing.
2. **Technical Compatibility Issues:**
 - a. **Probability:** Medium
 - b. **Impact:** Compatibility issues with existing hardware, software, or operating systems, resulting in delays or disruptions during implementation.
3. **User Adoption Challenges:**
 - a. **Probability:** High
 - b. **Impact:** Unfamiliarity with simulation tools or reluctance to change established way of playstyle / interaction.
4. **Performance Bottlenecks:**
 - a. **Probability:** Medium
 - b. **Impact:** Results in slowdowns, crashes, or degraded user experience.
5. **Resource Constraints:**
 - a. **Probability:** High
 - b. **Impact:** Budgetary constraints, staffing shortages, or time limitations, impact the project's ability to meet deadlines, deliverables, or quality standards.

6. **Feature Bloat:**
 - a. **Probability:** High
 - b. **Impact:** Uncontrolled expansion of project scope, with additional features being added over time without proper assessment of their impact on timeline, budget, and resources.
7. **Dependency on Third-party Technologies:**
 - a. **Probability:** Medium
 - b. **Impact:** Disruptions or limitations due to dependencies on third-party technologies, leading to delays or functional limitations.
8. **Market Competition:**
 - a. **Probability:** Medium
 - b. **Impact:** Similar simulation software or competing solutions, leading to challenges in customer retention, or pricing pressure.

33 Costs

| <u>Activity / Deliverable:</u> | <u>Estimated Cost (Range):</u> |
|---------------------------------------|---------------------------------------|
| Work Context Analysis | \$5,000 - \$7,000 |
| Business Events | \$3,000 - \$5,000 |
| Functional Requirements | \$1,0000 - \$15,000 |
| Nonfunctional Requirements | \$7,000 - \$9,000 |
| Constraints | \$3000 - \$5,000 |
| Development Resources | \$50,000 - \$80,000 |
| Hardware Infrastructure | \$20,000 - \$30,000 |
| Software Licenses | \$10,000 - \$15,000 |
| Training / Support | \$ 5,000 - \$10,000 |
| Contingency / Reserve Budget (10%) | \$10,500 - \$15,500 |
| <u>Total Estimated Cost:</u> | <u>\$114,500 - \$181,500</u> |

Table #4 : Estimated Cost of Production

34 Waiting Room

1. Content: Advanced Simulation Features

- a. **Motivation:** To enhance the capabilities of the simulation platform, providing users with more advanced tools for realistic structural design and testing.
- b. **Consideration:** These features may require significant development effort and technical expertise, making them more suitable for future releases when resources and technology allow.

2. Content: From Desktop to Mobile

- a. **Motivation:** To improve user engagement and satisfaction by incorporating features that allow users to play on the go.
- b. **Consideration:** Implementing these features may require additional hardware support or software optimizations, which may be better suited for future iterations of the product (touchscreen support, etc).

3. Content: Customization Options

- a. **Motivation:** To empower users with greater creativity in their simulations by providing extensive customization options for building materials, environmental conditions, and disaster scenarios (realistic and fantastical).
- b. **Consideration:** Implementation may require significant development effort, making these additions ideal candidates for future iterations of the game.

4. Content: Accessibility Features

- a. **Motivation:** To ensure inclusivity and usability for all users, including those with disabilities, by incorporating accessibility features such as screen reader compatibility, color-blind mode, and keyboard shortcuts.
- b. **Consideration:** Prioritizing accessibility features may require specialized expertise and thorough testing to ensure compliance with accessibility standards and regulations, warranting further exploration in future iterations.

5. Content: Localized Versions

- a. **Motivation:** To cater to users from different regions and language preferences by offering localized versions of the simulation platform (including translations / cultural adaptations).
- b. **Consideration:** Development may involve additional resources and coordination with translators and localization experts, making it a more feasible consideration for future releases.

6. Content: Gamification Elements

- a. **Motivation:** To enhance user engagement and motivation by incorporating gamification elements such as achievements, rewards, and progression systems into the simulation platform.
- b. **Consideration:** Implementing gamification features may require careful balance and iteration to ensure they complement the core game experience without detracting from its overall value, warranting further exploration in future releases.

35 Ideas for Solutions

1. Content: Real-Time Physics Engine:

- a. **Idea:** Use a robust real-time physics engine, such as NVIDIA PhysX or Unity Physics, to accurately simulate dynamic forces and interactions in the virtual environment.
- b. **Motivation:** Enhance the realism and accuracy of simulations by leveraging advanced physics simulation capabilities.
- c. **Considerations:** Ensure compatibility with existing software architecture and optimize performance for smooth and responsive simulations.

2. Content: Standard Peripheral Compatibility

- a. **Idea:** Implement support for popular gaming controllers, such as Xbox and PlayStation controllers, to enable users to interact with the simulation platform using their preferred gaming peripherals.
- b. **Motivation:** Enhance user experience and accessibility by allowing users to utilize familiar controllers for navigating and interacting with the simulation environment.
- c. **Considerations:** Ensure seamless integration with controller input mappings for both Xbox and PlayStation controllers, provide customizable controls to accommodate user preferences,

36 Project Retrospective

As we approach the completion of Make-or-Break's documentation, it's important to reflect on the journey we've taken to efficiently and coherently fill out each section of this development report. Throughout the project, certain approaches proved effective, others not so much. Below we dive into what worked well and areas where improvements could be made.

What Worked Well: The usage of a dedicated discord channel, along with weekly team meetings helped facilitate clear and timely communication among team members. Most of us would check discord fairly regularly so if any questions or concerns/thoughts were raised there would be an easy way to decide the best path moving forward. This helped in sharing progress updates, and dividing work based on what each team member could do given their workload for that week.

What Didn't Work Well: We would typically underestimate the time a task or section of the report would actually take, which led to stressful last minute pushes to make sure deliverables were submitted on time. Additionally, there was sometimes a lack of consideration for the fluctuating schedules of team members due to personal or work commitments. We could have been a bit more proactive in adjusting our plans to accommodate these changes throughout the semester.

Recommendations for Future Improvement: Include better/more detailed diagrams, since diagrams are a great way to facilitate a reader's understanding. Keeping expectations manageable as well would help, seeing as we would come up with big ideas that would sometimes veer us off course of what a project like Make-or-Break is intended to be. Lastly, maintaining flexibility in accommodating conflicting schedules and work sessions will greatly benefit a team in the long run, much like the agile principles we've been following throughout the semester, in adherence to the Scrum methodologies. This ensures smoother collaboration and helps streamline the process of drafting large project reports.

V Glossary

- **Deterministic physics:** Physics where the outcome is solely determined by the inputs. (no randomness)
- **Physics simulations:** simulations inside the game that uses certain physics related formulas to accurately pinpoint forces put onto the structures or buildings from natural disasters.
- **Curators:** collection of users that have formed a community from our game.
- **Player:** person who buys and plays the game on their system.
- **Updates/Mods:** modifications made to the game post-release, (mods usually by players, and updates/patches by developers)
- **Real-Time Simulation:** Changes made by players are immediately reflected in the game's environment

VI References / Bibliography

| | |
|--|--|
| | Temporary Logo: https://www.shutterstock.com/zh/search/disaster-logos |
| | J. Bell, "A Sample Document for Generating Consistent Professional Reports," University of Illinois Chicago, 2018. |
| | M. Fowler, UML Distilled, Third Edition, Boston: Pearson Education, 2004. |
| | B. Bruegge, A. Dutoit, "Object Oriented Software Engineering: Using UML, Patterns, and Java", Third Edition, Pearson Education: 2009 |
| | Diagrams: https://online.visual-paradigm.com/diagrams/features/use-case-diagram-software/ |

VII Index

| | | | |
|----------------------------|-----------------|-----------------------|------------------|
| Project Description | (3 - 14) | Requirements | (15 - 45) |
| Design | (46-60) | Project Issues | (61-68) |
| Glossary | (69) | References | (70) |