

Platform Perils

Detailed Design

Steven Palmer

⟨palmes4⟩

Chao Ye

⟨yec6⟩

April 26, 2016

Contents

| | | |
|----------|---|----------|
| 1 | Introduction | 1 |
| 2 | Module Interface Specification | 1 |
| 3 | Error Handling | 1 |
| 3.1 | Missing Files | 1 |
| 3.2 | Library Initialization Failures | 1 |
| 3.3 | Unexpected Errors | 1 |
| 4 | User Interface Elements | 2 |
| 4.1 | User Interface | 2 |
| 4.1.1 | Menu | 2 |
| 4.1.2 | Game Stages | 3 |
| 4.2 | Object Meshes | 6 |
| 5 | Key Algorithms | 7 |
| 6 | Relational Database Structure | 7 |
| 7 | Communication Protocols | 7 |
| 8 | Implementation Specifics | 7 |
| 8.1 | Language | 7 |
| 8.2 | Supporting Libraries | 7 |
| 9 | Appendix A: Mesh Samples | 8 |

List of Tables

| | | |
|----------|---------------------------------------|----------|
| 1 | Supporting Libraries | 7 |
|----------|---------------------------------------|----------|

List of Figures

| | | |
|----------|------------------------------|----------|
| 1 | Level select screen. | 3 |
| 2 | Stage 1 example. | 4 |

| | | |
|---|------------------------------|----|
| 3 | Stage 2 example. | 5 |
| 4 | Stage 3 example. | 6 |
| 5 | Sample mesh: gate. | 8 |
| 6 | Sample mesh: rock. | 9 |
| 7 | Sample mesh: spear. | 10 |
| 8 | Sample mesh: spikes. | 11 |

Revision History

| Date | Version | Notes |
|------------------|---------|------------------|
| January 10, 2015 | 1.0 | Created document |
| January 11, 2015 | 1.1 | Rev 0 submission |
| April 25, 2016 | 1.2 | Rev 1 submission |

1 Introduction

This document makes up the second part of the design document. The first part is found in the ‘System Architecture’ document, which contains a proper introduction for the document as a whole.

2 Module Interface Specification

A module interface specification was prepared using Doxygen. This is included as a separated document entitled ‘Detailed Design - Module Interface Specification’.

3 Error Handling

3.1 Missing Files

Several files will be included as part of the game, and will be required for the game to function properly. Required file types include sound files, object mesh/texture files, and shader files. Any missing file will produce a message indicating that the file was not found. The game will continue to run if sound files are not found, but will not play any missing sounds. Missing object or shader files, however, is a more serious error and will result in termination of the application.

3.2 Library Initialization Failures

Library initialization failures, especially GLEW initialization errors, may occur when the game is attempted to be run on a system with outdated hardware. These types of errors will produce an error message followed by termination of the game application. All errors of this type will be considered unrecoverable.

3.3 Unexpected Errors

No other errors are expected to occur given that the game code restricts bad states from occurring.

User inputs will be well defined and tested for errors. Other user inputs (i.e. keys that are not used in controlling the game) have no programmed function and as a result should have no effect when pressed. It is still likely, however, that unexpected errors will occur from time to time due to uncaught bugs that rarely execute or external factors (driver error, memory error, etc.). These unexpected errors are not handled and will likely cause the game to crash.

4 User Interface Elements

4.1 User Interface

The game will consist of two general types of user interfaces: menus and game stages/levels. Examples of what these interfaces will look like along with some of their key features are given in the following subsections.

4.1.1 Menu

The main menu of the game will be the level select screen. An example of what this screen will look like is given in [Figure 1](#).

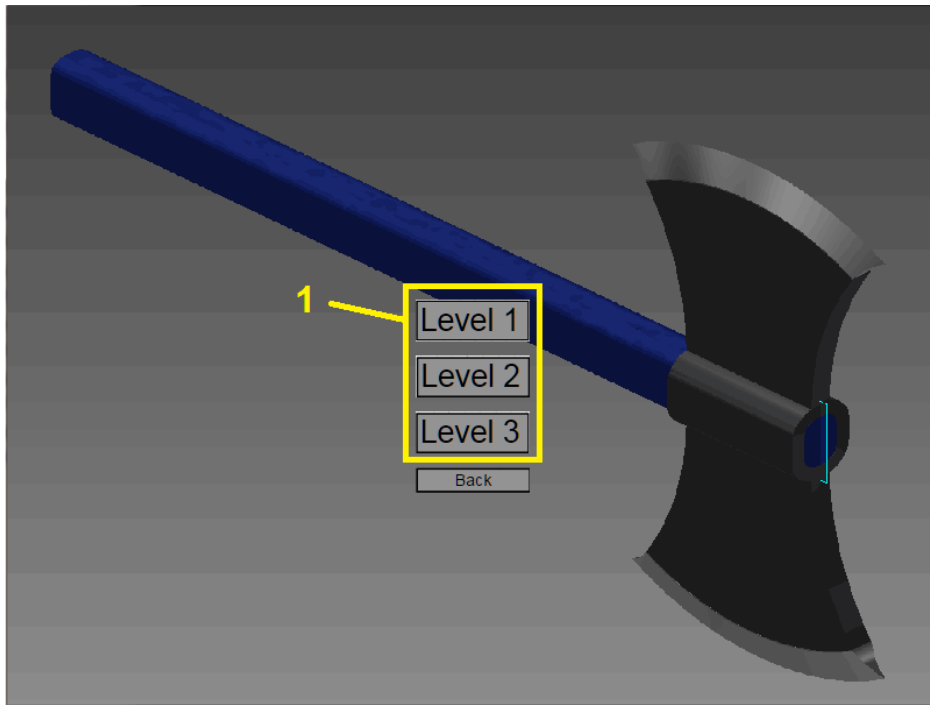


Figure 1: Level select screen.

1. When the game is launched, the user begins at the main menu. From here, a stage may be selected to be played.

4.1.2 Game Stages

Snapshots of the game stages are given in Figures 2 - 4. Each figure is used to highlight some of the features that appear in the game.

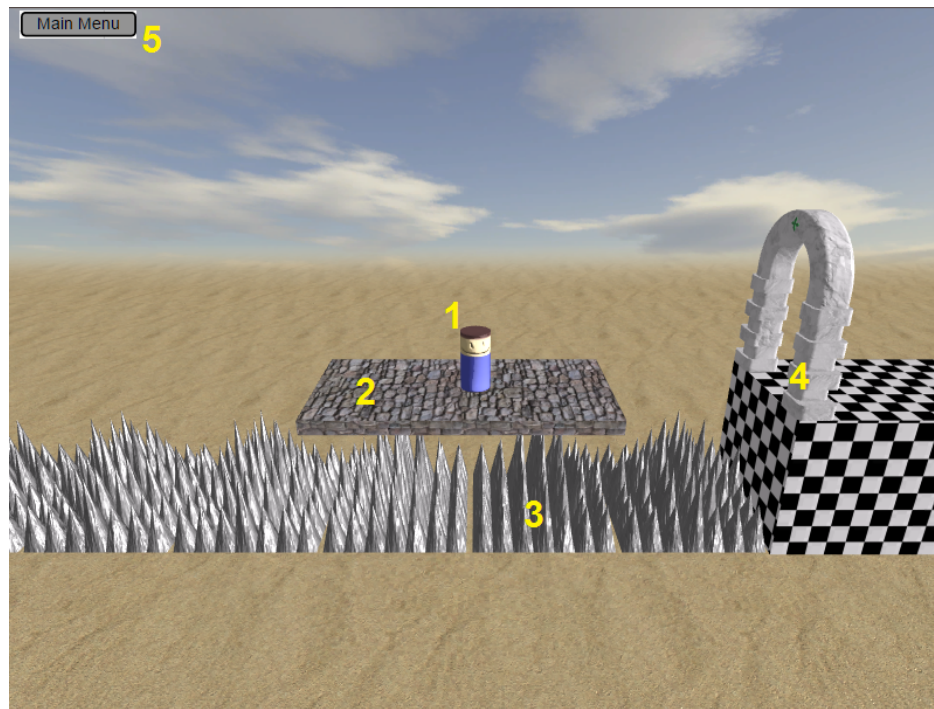


Figure 2: Stage 1 example.

1. The hero character that is controlled by the user.
2. A moving platform: transports the hero on a fixed path.
3. A spike hazard: the hero will be killed if contact is made.
4. The goal: the hero must reach this platform to complete the stage.
5. A button to exit the stage and return to the main menu.



Figure 3: Stage 2 example.

1. A boulder hazard: rolls freely and crushes the hero if caught in its path.
2. A ramp: allows the hero to climb to new heights.

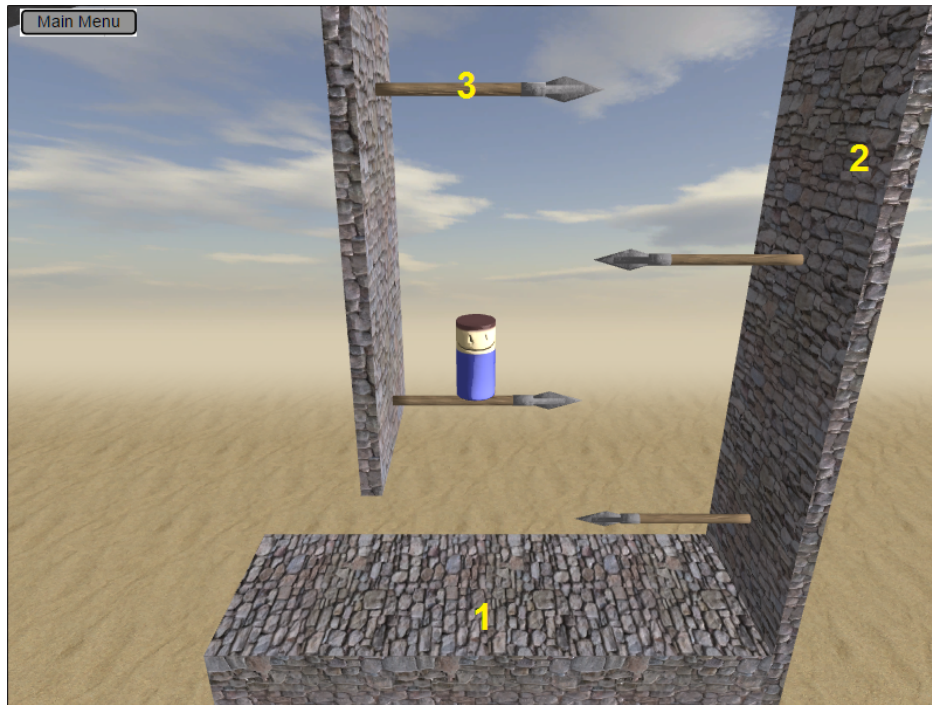


Figure 4: Stage 3 example.

1. A platform: the hero is supported by platforms and can move freely across them.
2. A wall: obstructs the hero's path.
3. A spear hazard: the pointy end impales the hero, but the wood shaft can be used as a platform.

4.2 Object Meshes

Creating meshes that accurately portray the objects they represent is an important consideration related to the user interface. Samples of object meshes that have been created for use in the game are given in Figures 5 - 8 in [Appendix A](#).

5 Key Algorithms

This section is not applicable to this project.

6 Relational Database Structure

This section is not applicable to this project.

7 Communication Protocols

This section is not applicable to this project.

8 Implementation Specifics

8.1 Language

The project will be implemented using C++. The use of C++ combined with OpenGL will allow for simple and efficient cross platform implementation.

8.2 Supporting Libraries

Libraries that will be used in the implementation of the game are listed in [Table 1](#).

Table 1: Supporting Libraries

| Library | Function |
|-----------------------------|--------------------------------|
| Chipmunk 2D | Physics engine |
| OpenGL | Graphics functionality |
| OpenAL | Audio functionality |
| GLFW | Window and IO management |
| GLEW | Advanced graphics capabilities |

9 Appendix A: Mesh Samples

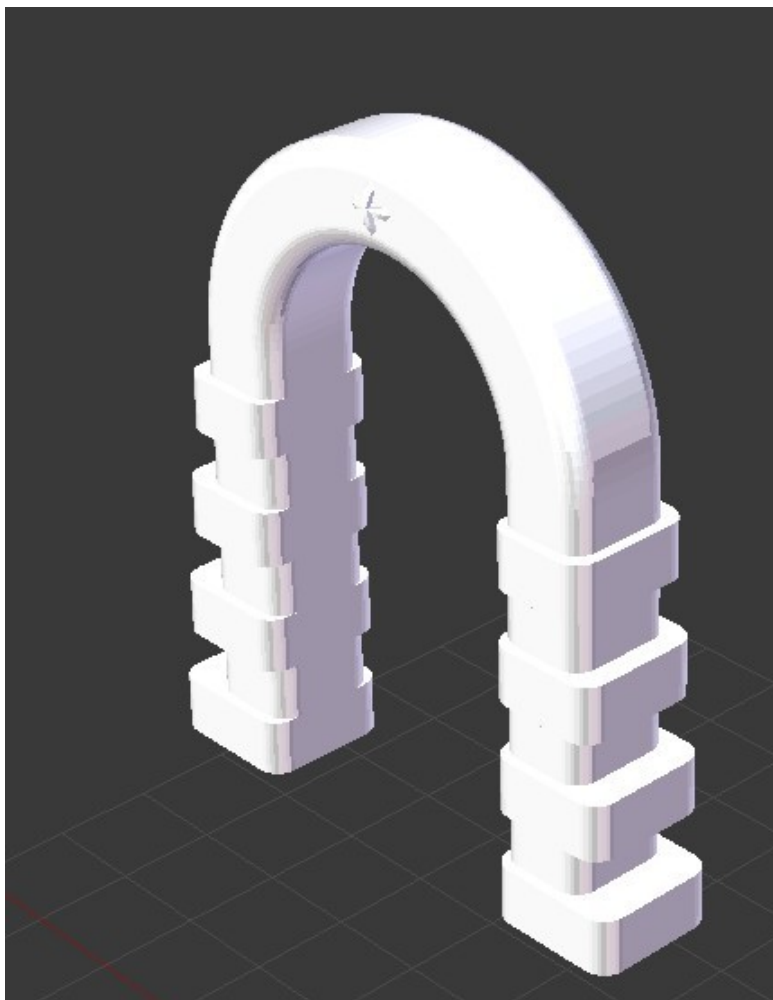


Figure 5: Sample mesh: gate.

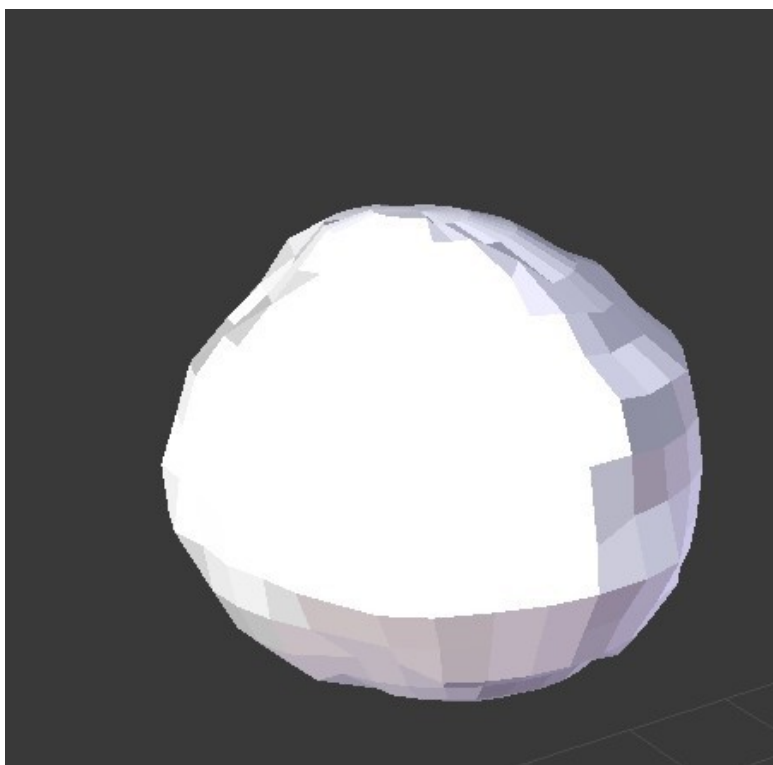


Figure 6: Sample mesh: rock.

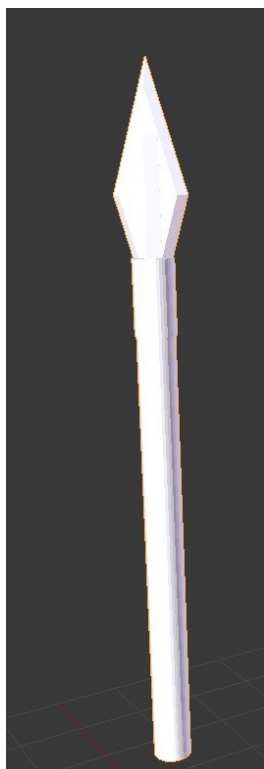


Figure 7: Sample mesh: spear.

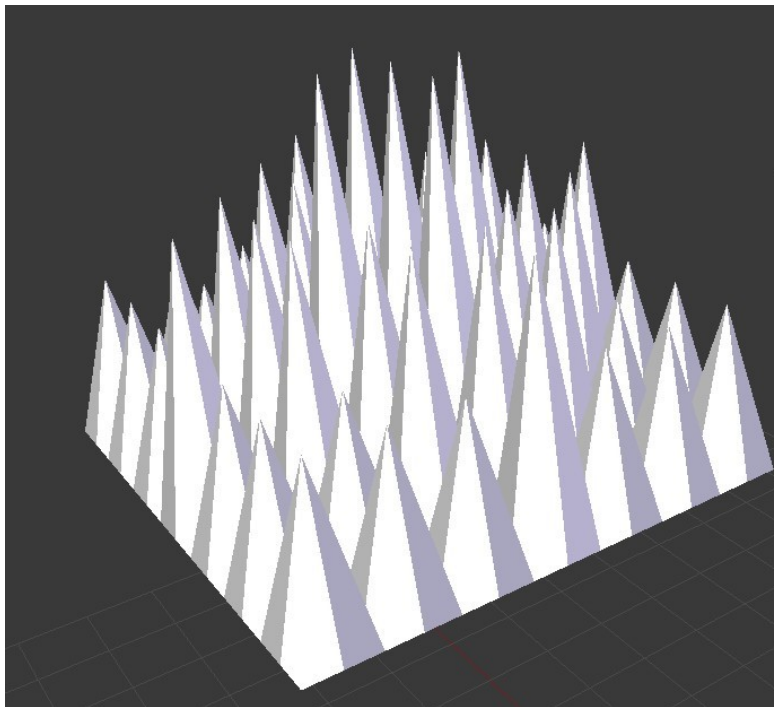


Figure 8: Sample mesh: spikes.