**University of Brighton   
Computer Games**

**CI411 - Introduction to Game Programming 2022-23  
Coursework 2: C++ SDL Game**

**Module Leader: David Dorrington**

**Student: Kennedy Sovine  
Date:**

*This is a template that you can use for CW2; you do not have to follow the suggestions strictly, but make sure you include all information necessary according to the Assignment!*

*Delete all yellow highlighted sections!*

*General Advice*

*Write only in the third person*

*Use formal language – Do not write as you speak*

*You will need to have references from at least 3 sources*

*Do not write unsupported stereotypical generalisations – try and back them up with references*

*If you are making assumptions – State them clearly*

*This is a report not an essay – you will need to introduce and contextualise each section but in a report bullet points can be very useful to list your key points*

*Make the information as easy to read as possible*

*Read what you have written – Get someone else to proof read it*

*Make sure you understand the “logic” of the way that you present the information and ideas.*

## Table of Contents

[Table of Contents 2](#_Toc133580266)

[Introduction 3](#_Toc133580267)

[Game Design 4](#_Toc133580268)

[Type and Style of Game / Genre 4](#_Toc133580269)

[Screen Layout and Format 4](#_Toc133580270)

[Control System 5](#_Toc133580271)

[Gameplay & Core Rules 5](#_Toc133580272)

[Level Designs 5](#_Toc133580273)

[Implementation 6](#_Toc133580274)

[Game Objects Diagram 6](#_Toc133580275)

[Program Structure 6](#_Toc133580276)

[Player Controlled Objects / Character 7](#_Toc133580277)

[NPCs 7](#_Toc133580278)

[Objects 7](#_Toc133580279)

[Interface 7](#_Toc133580280)

[Game Management 7](#_Toc133580281)

[Testing, Problems & Solutions 8](#_Toc133580282)

[Critical Review 9](#_Toc133580283)

[Conclusion 9](#_Toc133580284)

[Bibliography and References 10](#_Toc133580285)

[Graphical Assets 10](#_Toc133580286)

[Audio Assets 10](#_Toc133580287)

[Appendices 11](#_Toc133580288)

[How to play / game Walkthrough 11](#_Toc133580289)

*This is auto generated - Right click to update (Update Field)*

# Introduction

Overview of what this document is and what will be covered by which sections.

This document is a detailed overview of the development of my SDL game. This document covers the design decisions for the game, how ideas and techniques were implemented, testing, and a review of my own work.

The ‘Game Design’ section covers the design of the game. This includes the inspiration for the game and every design choice that went into it. It is essentially the idea behind the game.

The ‘Implementation’ section covers the parts of the game. This includes the NPCs, the PC, the game's management, and the program's structure. The implementation is the next stage of the game’s development as the idea discussed in ‘Game Design’ is realized and put into a physical form.

In ‘Testing, Problems, and Solutions,’ the document details any issues that occurred during development. This will also include solutions that were or were not implemented and solutions that may or may not have been found.

Lastly, the end of the document contains the bibliography and appendices, which includes a walkthrough and gameplay description for the game.

# Game Design

* Introduce this section
* What type of game has been made and what is it called?
* What is it similar to?
* Include a screen shot of the game

This section details the design decisions made for the game. The game is an arcade-style shooter inspired by the mechanics of classics such as *‘Space Invaders.’* This game is called ‘*Prism of Lights.’*

|  |  |
| --- | --- |
| SDL Game Developed | Game Influence |
| A picture containing scatter chart  Description automatically generated  Fig 1. Screen Shot of David’s Game | Space Invaders - Wikipedia  Fig 2. Screenshot of the game *Space Invaders*, developed by Taito. (fame, 2004) |

## Type and Style of Game / Genre

* Describe the genre of the game
* Describe the key genre conventions using a reference
* Describe the Visual style – gothic cartoon, abstract Fantasy, sci-fi, etc

The genre of the game is an arcade-style shooter. The conventions of these types of games are low-res sprites, along with simple mechanics of ‘point-and-shoot’. Since these games were first created for arcades, they are meant simply for entertainment and quick play. There is no story nor deeper meaning behind them like modern video games.  
  
The game's visual style is pixelated and inspired by contemporary fantasy. The world and scene are visually basic, not having any complex colors or details that would be hard to see in a low-resolution setting.

## Screen Layout and Format

* Describe the screen format, resolution and aspect ratio
* Are you using all or part of the screen as a board /play area?
* The view Top down, side, isometric
* fixed, 1 or 2 axis scrolling, Tiled etc
* Include sketch / diagram or screen grab where appropriate

The game's resolution is 1920 x 1080, which is standard 1080 resolution. The entire screen is the play area, and the view is top-down. The game is fixed, the level itself not moving around.

|  |
| --- |
|  |

## Control System

* Describe the Method of control (mouse, KB, Trackball, buttons, joystick, pad)
* Map of Controls

The methods of control are WASD and the mouse. WASD is used for movement controls, whereas the mouse is used for aiming to shoot.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Control | W | A | S | D | First Mouse Button | Mouse |  |  |
| Action | Move Forward | Move to the Left | Move Backwards | Move to the Right | Shoot | Aim |  |  |

## Gameplay & Core Rules

* Intended Style of gameplay
* Goals, Win & Lose states
* Key Gameplay actions
* Objectives, Progression, challenges

The intended gameplay style is a simple point-and-shoot, where the player shoots the enemies and can earn power-ups by moving around the map.

The goal is to get through 10 waves of enemies. A congratulatory message will appear if the player wins, and the game will end. The player can only win if they destroy all the enemies without losing all their health. The player loses if the enemies can attack them and reduce their health to 0. The enemies can only attack if they get too close, meaning the player needs to kill the closest enemies first to ensure their safety. This introduces an element of strategy.

The player must shoot and kill all enemies to proceed to the next wave. In every wave, there is a chance that a power-up will spawn. The powerups are blue, yellow, red, and white prisms, all with their own special ability.

## Level Designs

* Level plans of each level / phase
* Key Game Objects on Screen
* GUI: information - Score, health, lives, progress, location, goals
* Include sketches / diagrams or screen grabs

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

*Screen plan of XX*

# Implementation

Introduce this section

How was the game made

What Technology was used

The implementation section is all about how the ideas and concepts behind the game were implemented through code inside the program. Using techniques that were learned in class, the ideas were efficiently added to the program.

The code base of the program is the code that was provided during class. This was an efficient choice due to the code base already having many of the functions and features desired for the game.

## Game Objects Diagram

Diagram

Description automatically generated

Figure 3 Sample Object Diagram

## Program Structure

* How is the program structured?
* CPP Files and Headers
* Key functions
* Program flow
* Game Loop

|  |  |
| --- | --- |
| Fig X XXXX | Fig x XXXXXX |

## Player Controlled Objects / Character

* Control Method
* PC Key States and Variables
* Functionality and actions
* Important Code snippets

## NPCs

* Overview of NPCs
* Key States
* Key functionality / behaviours
* Important Code snippets

## Objects

* Overview of important objects
* Key States / functionality
* Important Code snippets

## Interface

* Key functionality
* Screen Grabs /Diagram
* Important Code snippets

## Game Management

* Win / lose states
* Progress / Objectives
* What is being managed
* Replay-ability
* Important Code snippets

Anything else you Feel you should mention

# Testing, Problems & Solutions

* Problems that were encountered
* Solutions that were found, implemented, not implemented, not found

|  |  |  |  |
| --- | --- | --- | --- |
| **Priority** | **Problem** | **Solution** | **Implemented** |
| *Low/Med High* | *Nature of the problem e.g.:*   * *Images not correct (transparency, jagged)* * *Collision not working* * *Score won’t reset* * *Sounds plays at wrong moment* * *Bugs* * *NPCs get stuck* * *Bullets pass through walls* * *Too Simple / Too difficult* | *What could be done to rectify the problem if known.  State if you do not know how to fix the problem* | *Solution has been implemented within timescale*  *Could not have been applied in current timescale*  *Need to research problem* |
|  |  |  |  |
|  |  |  |  |

# Critical Review

* identify three reasons why the design and implementation of the game are good.
* Further identify three reasons where the implementation could be improved and a summary of how the improvements could be made.
* What could you have done better
* How would you improve the gameplay
* What have we learnt from this process

# Conclusion

* what are the main take-away messages –
* what are the key concepts that you learned during the development.

# Bibliography

fame, g. h. o., 2004. *gamespy.com.* [Online]   
Available at: https://web.archive.org/web/20040902162808/http://archive.gamespy.com/legacy/halloffame/spaceinvaders.shtm  
[Accessed 6 May 2023].

user17446225, n.d. *Seamless tileable texture of green bush..* [Online]   
Available at: https://www.freepik.com/premium-photo/seamless-tileable-texture-green-bush\_12492565.htm  
[Accessed 6 May 2023].

## Graphical Assets

|  |  |  |  |
| --- | --- | --- | --- |
| **Image** | **name** | **Used for** | **Sourced from** |
|  | PC run | PC | www.tileset.com |
|  | NPC\_smile | NPC1 NPC2, NPC3 | Drawn by Author in flash |
| A picture containing blur  Description automatically generated | BG\_Swirl | Level 1 background | www.abm-enterprises.net/fractals/  rainbowswirlwallpaper.html |
| Seamless tileable texture of green bush. | bush | Tile for shrubbery border | Freepik (user17446225, n.d.) |

## Audio Assets

|  |  |  |
| --- | --- | --- |
| **Sound** | **Used for** | **File & Source** |
| Background Music | Level 1 background | Littlest\_hobo.wav  www.soundforce.net |
| Arrow whoosh | PC Attack | Arrow.wav  Author Recorded in Audacity |
| Pig squeel | PC hit | Squeee.wav www.soundforce.net |
|  |  |  |
|  |  |  |

# 

# Appendices

## How to play / game Walkthrough

Use screen grabs if they help

Make sure we know how to complete the game!