Software Requirements Specification

for

Double Dash Dodge & Deceive

Version 1.0 approved

Prepared by Kieran Firkin

Code In the Shell

Feb 17, 2024

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Kieran Firkin	02/17/24	Initial filling of information	1.0

1. Introduction

1.1 Purpose

This software requirements specification intends to cover the entire scope of Double Dash Dodge & Deceive, a new multiplayer video game under development.

1.2 Document Conventions

This document follows IEEE standards for in-text references. End users should be referred to as “users” and in-game characters should be referred to as “characters” to avoid the ambiguity of the word “player”.

1.3 Intended Audience and Reading Suggestions

This document is intended for developer use. The SRS contains a list of functional and non-functional requirements, grouped by feature. Readers should skim through sections 1, 2, and 3. Section 4 should be read more thoroughly, noting features relevant to the reader’s role and reading sub-sections in their entirety.

1.4 Product Scope

The product, Double Dash Dodge & Deceive, is intended to be a multiplayer video game with competitive elements and rapid round turnover. The full product scope is specified in the product description document [1].

1.5 References

1. K. Firkin, M. Maksymenko, L. Cringle, A. Ahmad, and C. Thurston, “Project Description,” Code In the Shell, Feb. 2024. Accessed: Feb. 17, 2024. [Online]. Available: /Deliverable0/CodeInTheShell_Deliverable_0_ProjectDescription.pdf
2. Unity Technologies, “Unity - Scripting API”:, *Unity3d.com*, 2019. https://docs.unity3d.com/ScriptReference/

2. Overall Description

2.1 Product Perspective

This SRS specifies a brand new 2D platforming video game. The product is to be developed in Unity as a self-contained system, only relying minimally on external resources for web hosting.

2.2 Product Functions

Product functions are as follows:

- Provide an entertaining, competitive experience for two users.
- Provide a method for users to sabotage the opposing user and impede their progress.
- Provide methods for users to progress through the game in satisfying, high skill ceiling ways.

2.3 User Classes and Characteristics

The product shall cater to both casual users, as well as experienced users. Casual users do not use the product often, and so should be able to pick up the game with little to no practice required. Experienced users on the other hand should be able to find enjoyment in perfecting their movement skills. Casual users have a higher priority compared to experienced users as they will likely make up a greater proportion of end users.

2.4 Operating Environment

The product will operate in the Google Chrome version 121 and Firefox version 122 internet browsers on the Windows and Linux operating systems under the WebGL framework offered through Unity.

2.5 Design and Implementation Constraints

The product is designed using the Unity platform targeting WebGL. As such, it must comply with the constraints enforced by Unity's architecture. Additionally, as the product is being developed for web use, it must comply with more stringent performance constraints associated with web development as opposed to standalone programs.

2.6 User Documentation

The product shall include a built-in help screen, easily accessible from the main menu, which describes game mechanics such as how to control one's player character. Additionally, there will be a readme file included with the game describing the operation of the software.

2.7 Assumptions and Dependencies

The product is dependent on Unity's WebGL target working for Firefox and Google Chrome browsers in an acceptable manner. It is assumed that Unity's built in physics components will be sufficient to implement the movement techniques that are planned to their fullest extent.

3. External Interface Requirements

3.1 User Interfaces

The system has three main user interfaces: the main menu, the pause menu, and the main game loop. The main menu should have a button to start the game, and a slider to control game volume. The pause menu should be accessible from the main game loop, and should be a semi-transparent overlay providing a button to exit to the main menu, and a slider to control game volume. The main gameplay loop should feature a split screen where each user can move independently, and each user is provided a level progress bar indicating how close they are to their respective goal.

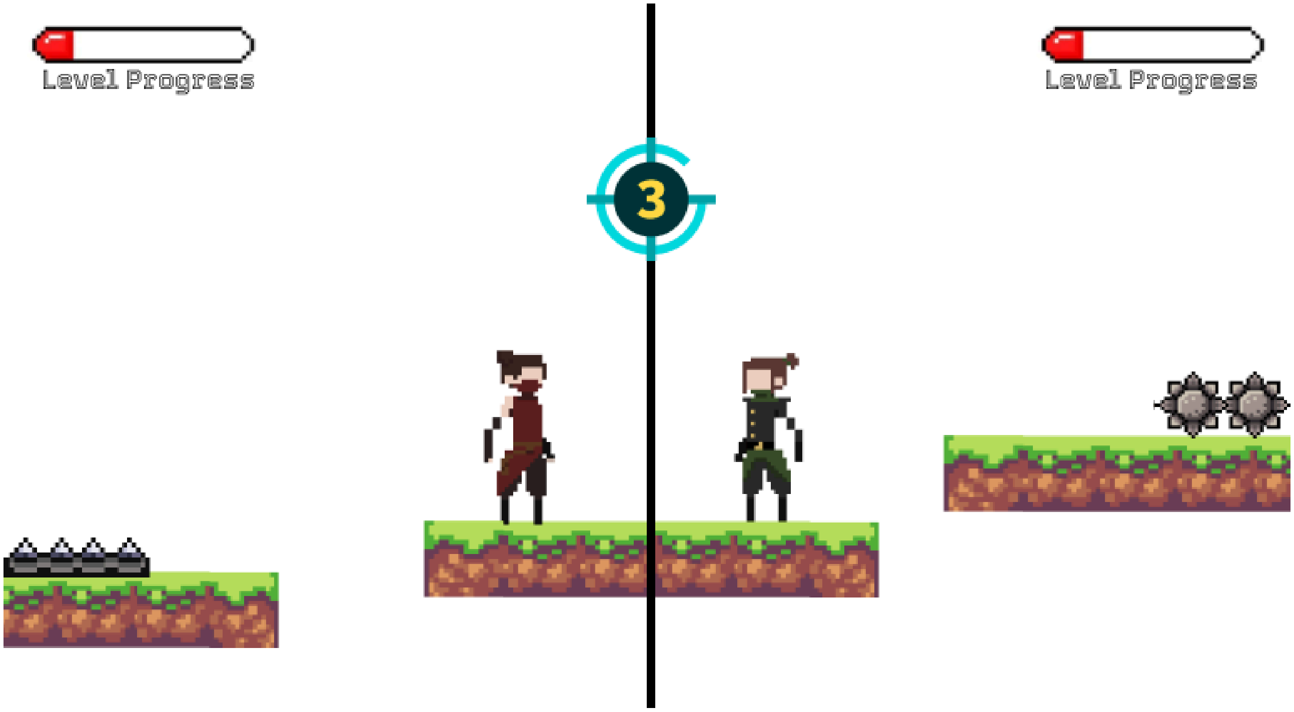

Fig. 1. Main game loop UI mockup.

3.2 Hardware Interfaces

As the product is being developed for web browsers through Unity, there is minimal direct hardware interfacing. However, the product is intended to work for screens of various sizes and aspect ratios, dynamically resizing UI components to fit to the screen.

3.3 Software Interfaces

The product must interface with web browsers to display graphical information to the user, as well as to capture user input. Unity handles this interfacing on behalf of the product, exposing simple API calls like Input.GetButtonDown() to be used in the product source code. These API calls are well documented on Unity's website [2].

3.4 Communications Interfaces

It is not necessary for the product to communicate with external resources outside of initially loading the webpage from a server, as it only operates on local data which is not saved between sessions.

4. System Features

4.1 Player Movement

4.1.1 Description and Priority

The player character should be able to move around the world by walking, running, jumping, grappling, and dashing. This is a high priority feature as it is integral to the user experience.

4.1.2 Stimulus/Response Sequences

This feature comes into play when the user selects play from the main menu.

4.1.3 Functional Requirements

REQ-1: The system shall move the character left or right when the user presses left or right respectively.

REQ-2: The system shall impart a force on the character upwards when the user presses the jump button and the character is currently grounded. This force may be modulated by the amount of time the jump button is held.

REQ-3: The system shall impart a force on the character upwards and to the side when the user presses the jump button and the character is touching a wall. This force may be modulated by the amount of time the jump button is held.

REQ-4: The system shall impart a force on the character in the direction they are facing when the user presses the dash button.

REQ-5: The system shall extend a grappling hook from the character in the direction they are facing when the user pressed the grapple button.

4.2 User Interface

4.2.1 Description and Priority

The user should be able to glean useful information about the system easily in order to effectively use the system. The user should also be able to interact with the system easily. This is a medium priority feature, as while it's necessary to add before public use, it need not be implemented early on as no other subsystems rely on it.

4.2.2 Stimulus/Response Sequences

This feature comes into play when the user selects play from the main menu.

4.1.3 Functional Requirements

REQ-6: The system shall present the user with a progress bar indicating how far they have progressed through the current level.

REQ-7: The system shall present a help screen explaining core game mechanics and controls when the user clicks the help button.

REQ-8: The system shall present the user with a pause button which, when clicked, temporarily halts all in-game action.

REQ-9: The system shall present the user with a volume slider to control how loud the game is.

REQ-10: The system shall present the user with a countdown timer before each round to allow them to prepare themselves.

4.3 Game Structure

4.2.1 Description and Priority

The system should provide a mechanically interesting experience to the user, impeding them while still allowing them to progress by moving towards their goal. This is a high priority feature, as the rest of the features build upon it heavily.

4.2.2 Stimulus/Response Sequences

This feature comes into play when the user selects play from the main menu.

4.1.3 Functional Requirements

REQ-11: The system shall provide a method by which a user can send traps to the opposing user in order to impede their progress. Exact method is TBD [1].

REQ-12: The system shall indicate that a user has won once they progress their character far enough towards their side.

REQ-13: The system shall improve the character’s ability to progress when they collect a power-up. Exact method is TBD [2].

REQ-14: The system shall accept input from two separate users simultaneously, with each being able to control opposing, independent characters.

REQ-15: The system shall present users with an obstacle course that their characters must progress through.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The system shall respond to user input within 5 seconds 95% of the time in menus.
The system shall respond to user input within 0.1 seconds 99% of the time during gameplay.
The system shall launch to the main menu within 30 seconds of pressing play 90% of the time.
The system shall run at a constant 60 frames per second 95% of the time.
The system shall be no bigger than 10 gb of download space.
The system shall not use more than 4 gb of RAM space.

5.2 Safety Requirements

The system shall not produce flashing lights or images which may cause epileptic seizures.

5.3 Security Requirements

The system shall never access personal user data on a user's computer, to minimize security risks associated with the product.

5.4 Software Quality Attributes

The system shall be coded to allow easy modifications to game systems.
The system shall have a consistent 2D art style throughout 99% of the system.
The system shall be developed on a version of unity released within the last 1 year.

5.5 Business Rules

There is no distinction between end users in terms of operator privilege. All users may perform all functions at appropriate times.

Appendix A: To Be Determined List

[1] The exact method by which users are able to impede their opponents is undecided as of yet.

[2] The exact method by which power-ups improve the player's ability to progress through the game is undecided as of yet.