Brick Breaker

Software Requirements Specification

Document

(Group 3)
(Skyler Davis, Jason Hitchcock-Sivak, Nick Wieskamp,
Jacob Davidson)

Version: 1	Initial Document	**Date: 01/30/2017**

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to describe and explain the overall design, features, and requirements that are needed to create the Brick Breaker computer application. The document is also used to describe the App so that if another person were to design this game they would be able to.

1.2 Scope

The Software product Brick Breaker is a single player game for the PC. The levels consist of walls on the sides and top of the screen. The player will have control over a small platform that remains on the bottom of the screen and can only move left and right. The player will use the platform to make contact with a ball that moves continuously around the screen. The ball moves freely until it comes in contact with either the left, right, or top edges of the screen, or the the platform, at which point the ball will ricochet in the opposite direction. If the player fails to contact the ball as it approaches the bottom of the screen, the ball will disappear off of the screen and the player will lose a life. There will be rectangular "bricks" placed towards the upper part of the screen in various patterns. The objective of the game is to bounce that ball off of the player controlled platform and hit the bricks with the ball, breaking them (removing them from the screen), until no bricks remain, at which point, the next level will begin. The player will continue in this sequence until the ball falls of the screen three times, at which point the game will end. This PC app will be created using the Unity Development Platform because it allows the game to be run in different environments fairly easily.

1.3 Definitions, Acronyms, and Abbreviations.

Power up- Bonus abilities that will slowly fall down from a broken brick that affect the gameplay in various ways (see section 3.2.2.2.2)
Unity- Development platform that the pc app will be created on
Platform - Player-controlled piece of the game that the ball bounces off of to keep the game going
Walls- Barriers on the side and top of the screen, so if the ball hits them it will bounce off without them losing a life
Bricks- Rectangular objects that, after the ball hits the brick the required amount of times, will disappear and the ball will bounce off just like for the walls or platform

1.4 References

No references are needed.

1.5 Overview

The rest of the document will cover the overall description of how the app is put together and will also show the specific requirements that the game needs so that it can function properly. The document will also discuss in detail the overall design structure of the PC application.

2. The Overall Description

2.1 Product Perspective

2.1.1 System Interfaces

The Brick Breaker PC application will not be requiring any additional systems to run on PC besides the Unity interface.

2.1.2 Interfaces

The user interface will only consist of the input from the keyboard arrow keys to move the platform left and right and the mouse, which can be used for menu selection, and to start and stop the game.

2.1.3 Hardware Interfaces

The hardware requirements for Brick Breaker will require the user to have a desktop/laptop with a graphics card containing DX9 or DX11 and feature level 9.3 capabilities.

2.1.4 Software Interfaces

The Unity Development Platform will be used when making the game. It will help with portability along with making the game a desktop application.

2.1.5 Communications Interfaces

Brick Breaker is a single player game and will not need/require communication interfaces with other systems to perform properly. Everything will be accessed and stored in local storage.

2.1.6 Memory Constraints

The average amount of RAM that PCs would have are between 4GB and 8GB and our application will be well within those constraints so there will not be any issue with memory. If a user wishes to use the application via laptop, the laptop must have at least 2GB of RAM in order to run our application.

2.1.7 Operations

The only operation the application will need to perform will be running smoothly and keeping track of the user's score.

2.2 Product Functions

The function of Brick Breaker is to serve as a game that can be played on any laptop or computer.

2.3 User Characteristics

There are no specific user characteristics expected or required to use this application. It is intentionally meant to be simple, fun, and accessible to all potential users.

2.4 Constraints

Due to the intended simplicity of the Brick Breaker game, there are not any reasonable constraints on developers except that extra features should not be implemented if it is at the cost of the game's ease of use. Our number one priority at the moment is to have a functional game for users to play before adding unnecessary features.

2.5 Assumptions and Dependencies

For at least the first phase of this application, Brick Breaker will depend on a Windows operating system (7 or above). This means that virtually all PC users will be able to play the game.

2.6 Apportioning of Requirements

Brick Breaker may be played on a Mac operating system during later phases of development.

3. Specific Requirements

3.1 External Interfaces

No external interfaces are required for this application.

3.2 Functions

3.2.1 Phase 1

3.2.1.1 Start

3.2.1.1.1 When users launch the application, a main menu shall be displayed for ease of access.

3.2.1.1.2 During Phase 1, the menu shall only allow the user to begin the game or exit the window. This simplistic first design will be the building blocks for any added features later.

3.2.1.2 Game Play

3.2.1.2.1 After selecting the start option from the main menu, the first level will be loaded to the screen, indicating that the user can begin the game.

3.2.1.2.2 The user may then click anywhere on the screen to send the ball out or hit the left or right arrow keys to position their platform before clicking. After the initial click, the actual game play will start and the arrow keys will be the only required buttons until a life is lost or the level is beaten by destroying all the bricks on the screen.

3.2.1.2.3 The game starts by sending a ball off of a platform. The platform can be moved with the left and right arrow keys, a possible option would be using the popular ‘a’ and ‘d’ keys for side-to-side motion as well.

3.2.1.2.4 The ball will move up the screen until it connects with a rectangle (brick). Upon contact the brick will disappear from the display and the ball will bounce off the brick with the same angle it connected at.

3.2.1.2.5 If the player fails to move the platform to a position where it will redirect the ball upwards, the ball will be destroyed at the bottom of the screen and the user will lose one of their

three initial lives.

3.2.1.2.6 When the user loses all three lives, the user will have the option to begin a new game, or to return to the main menu.

3.2.2 Phase 2

3.2.2.1 Start

3.2.2.1.1 The main menu will now have a third option to view a table of high scores, which will be created based on the point system that will also be implemented in this phase.

3.2.2.2 Game Play

3.2.2.2.1 When the ball breaks a brick, somewhere on the screen (conveniently located) a numerical score will begin to increase as more bricks are broken.

3.2.2.2.2 This phase will also introduce that possibility of bonuses, which can be dropped randomly towards the platform when certain bricks are broken. These power ups will have different effects on gameplay.

3.2.2.2.2.1 Extra Ball: This power up will make another ball appear on the screen. Both balls will behave normally and the user can continue to play with both balls until one falls off the screen, at which point the game will continue with only one ball and the user will not lose a life. As long as one ball is still active, the game will continue.

3.2.2.2.2.2 Fireball: This power up will allow the ball to continue on its same path after contacting a brick. The ball will continuously hit and pass through bricks, breaking each it comes into contact with. The ball will only change direction when it hits a wall or the user's platform. The fireball will most likely only last one or two hits off the platform before changing back to a normal ball.

3.2.2.2.2.3 Extra Life: This power up will give the user an extra life for their current game. Remaining lives at the end of the level will be converted to extra points for a high score. Optionally this power up could be changed to be a point boost rather than an extra life.

3.3 Performance Requirements

The game must be able to perform smoothly and without lag at a comfortable frame rate

that will be decided on through testing. At this point, the application should just handle the basic physics and gameplay of Brick Breaker without frame jumping.

3.4 Logical Database Requirements

This game will not need to save any data until Phase 2, when high scores will be saved. However high scores will be stored in a text file that will be linked with the application folder and stored locally, so no online database will be needed.

3.5 Design Constraints

The only constraint with design is to ensure that the user experience remains simplistic, therefore making overly complex settings or menus is highly discouraged. The game will be centered on running properly first before extra features are added.

3.6 Software System Attributes

3.6.1 Reliability

The design of the game itself will follow all guidelines and functions that are listed. For example, when the ball gets past the platform, a life will be subtracted every time. The bounce of the ball will also be true to its initial angle of approach and bounce off correctly. Each of these functions must be working properly each and every time. It will be mandatory that the game does not lag and is visually smooth so the experience for the user is the best it can be. Once implemented, the high score system must be reliable for saving high scores across multiple system start-ups. In other words, if the game is shutdown and restarted, it will remember the saved data from last session and beyond.

3.6.2 Availability

This game will run infrequently. It only needs to run after the user request. It is not meant to have an autosave feature, since early on it will not have many levels. In other words, if the game crashes, the user will have to restart from the beginning.

3.6.3 Security

This application will not need any security measures because this will be an offline game that will not interface with any other user besides the one that has downloaded the game directly to their PC. Everything will be locally stored and lightly secured.

3.6.4 Maintainability

If bugs are reported the developers will find the bugs and work on a solution. The game will start out basic with little amounts of color, with more being added as time goes on.

The servers will also be monitored in order to ensure that it does not crash. The simplistic original design will allow easy maintenance if any problems arise.

3.6.5 Portability

This system will be initially developed for Windows, but in Phase 2 of this game, it may become available to Macs. The Unity software being used is intended on making the game more portable, so that the code does not have to be entirely rewritten. The way Unity works allows the developers to embed the game in a website if wanted.

3.7 Organizing the Specific Requirements

3.7.1 System Mode

This game will not be set in separate modes, and will only function in one way, therefore no changes in performance or any other categories will take place. Everything will be run in the same environment, every time.

3.7.2 User Class

All users will be grouped together as a collective class. No specific users will have different functionality in any way. Everyone who plays this game will be treated equally and there will be no benefits or disadvantages for any users.

3.7.3 Objects

The only object that will have interaction with the game is the keyboard. The keyboard will be the linking of the user to the game. The mouse is technically included in this input class, but the main point is that the keyboard will be the way the user controls the flow of the game.

3.7.4 Features

Loss of life will be the main feature that ultimately affects the gameplay. If a user lets a ball get past their platform, a life will be taken away. When all lives are gone, the game will end and the level will be failed.

3.8 Additional Comments

No additional comments are needed.

4. Change Management Process

The change management process for the development of the Brick Breaker game will be based upon good communication within the development team. This communication will be facilitated with the use of a GitHub repository, which will easily allow any team member to post changes that need to be made, as well as allow different members to experiment in their own branches before merging into a master branch.