

Table 1: Revision History

Date	Developer(s)	Change
Nov. 10th – Rev0.1	Andrew, Kyriakos, Teodor	first draft
Dec. 6th – Rev1	Andrew, Kyriakos, Teodor	modified every table in the document to match current project. Also updated Gantt charts. red text shows modifications.

SE 3XA3: Test Plan Space Pinball 2017

Team #17, Design Document MG
Andrew Bennett 1319879
Kyriakos Kyprianou 400025691
Teodor Tomescu 400038361

Contents

1	Introduction	2
2	Anticipated and Unlikely Changes	2
2.1	Anticipated Changes	2
2.2	Unlikely Changes	2
3	Module Hierarchy	3
4	Connection Between Requirements and Design	4
5	Module Decomposition	5
6	Traceability Matrix	5
7	Use Hierarchy Between Modules	6
8	Gantt Chart	7

Figures

1	Gantt Chart	10
2	Resources Chart.	10

1 Introduction

Space Pinball 2017 is a software project that aims to revive one of the most classic and well-known arcade games to date. This cross-platform video game will allow the user to use default key controls (or touch input on mobile devices) for various input in order to play the game and navigate through the menu. We will be using C# scripts written for the Unity engine that will enable us to allow user interaction with the game environment. Space Pinball 2017 will be programmed entirely in C# using Visual Studio. C# was the programming language of choice as Unity only gives you the choice of C# and Boo. Some of the object oriented principles that will be implemented in the code will be modularity, single class responsibility and inheritance.

In this document we will break down the various modules used by our software and explain how several requirements will be accomplished. We will also include a module hierarchy of the behavior hiding process, a software decision module, a traceability matrix between modules and requirements, and another traceability matrix between modules and anticipated changes.

2 Anticipated and Unlikely Changes

In this section we will explore some of the possible changes that may occur to our system. We will classify changes as being either likely or unlikely to occur. Anticipated changes are listed in Section 2.1, and unlikely changes are listed in Section 2.2.

2.1 Anticipated Changes

AC1. The operating systems on which Space Pinball 2017 will be running (eg. mobile devices)

AC2. Allow for modification of controls (eg. arrow keys instead of A and D)

AC3. The specific hardware on which Space Pinball 2017 will be running

AC4. The save/load game feature

2.2 Unlikely Changes

UC1. The score obtained through interacting with variable obstacles

UC2. The angle on which the table lies

UC3. The maximum power the ball can be launched into play

3 Module Hierarchy

Because Space Pinball 2017 is not too complex of a program, we do not have significant module depth. Because the GameManager Module manages the entire game, it would be on top in the hierarchy. As a result of that, the module hierarchy for our program is relatively flat.

M1: BallCollision Module

M2: BumperExplosion Module

M3: Flipper Module

M4: MenuButtons Module

M5: OutOfBounds Module

M6: PowerSpring Module

M7: ScoreManager Module

Level 1	Level 2
ScoreManager Module	BumperExplosion
PowerSpring Module	
OutOfBounds Module	
MenuButtons Module	
Flipper Module	
Ball Collision Module	

Table 1: Module Hierarchy for Space Pinball 2017

4 Connection Between Requirements and Design

Requirements	Modules	Explanation
R1- Display score	ScoreManager	The ScoreManager module takes care of displaying the score a player has received
R2- Pinball interaction with bumpers	BumperExplosion	BumperExplosion ensures that the pinball bounces off a bumper, and adds to the score appropriately
R3- User input starts the game	PowerSpring, MenuButtons	PowerSpring takes input from the user (space bar) and applies the appropriate power to the ball as it springs into the playing field. Menu Buttons allows the user to start a new game while navigating the main menu.
R4- Left Flipper activates when 'A' is pressed	Flipper	Flipper module sets the resting position, power, as well as the angle the flipper rotates and activates when the appropriate flipper when input detected (In this case 'A')
R5- Right Flipper activates when 'D' is pressed	Flipper	Flipper module sets the resting position, power, as well as the angle the flipper rotates and activates when the appropriate flipper when input detected (In this case 'B')
R6- Pressing the Menu button while in game will take the user back to the main menu	MenuButtons	MenuButtons module keeps track of all the buttons present in the UI throughout the game

Table 2: Connections between Requirements and Design for Space Pinball 2017

5 Module Decomposition

6 Traceability Matrix

This section shows two traceability matrices: between the modules and the requirements and between the modules and the anticipated changes.

Requirement	Modules
R1	M7
R2	M2
R3	M6, M4
R4	M3
R5	M3
R6	M4

Table 3: Traceability matrix between modules and requirements for Space Pinball 2017

AC	Modules
AC1	N/A
AC2	M3, M4
AC3	N/A
AC4	M4

Table 4: Traceability matrix between modules and anticipated changes for Space Pinball 2017

7 Use Hierarchy Between Modules

Uses Hierarchy

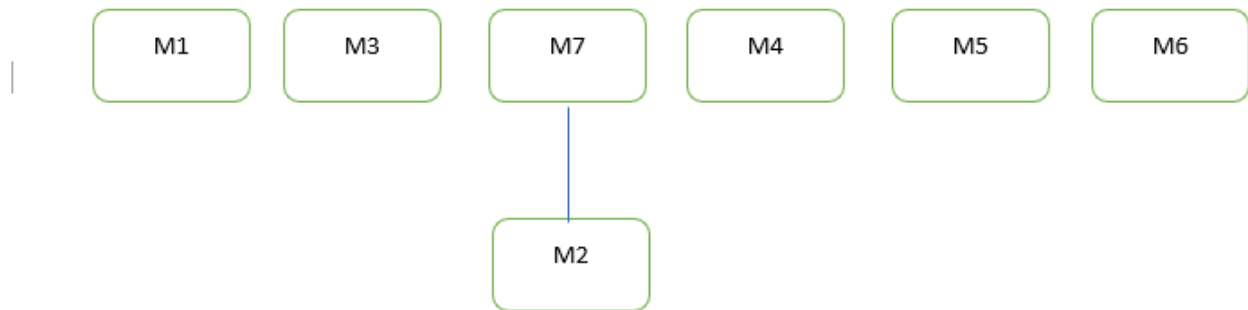


Figure 1: Use Hierarchy Among Modules

8 Gantt Chart

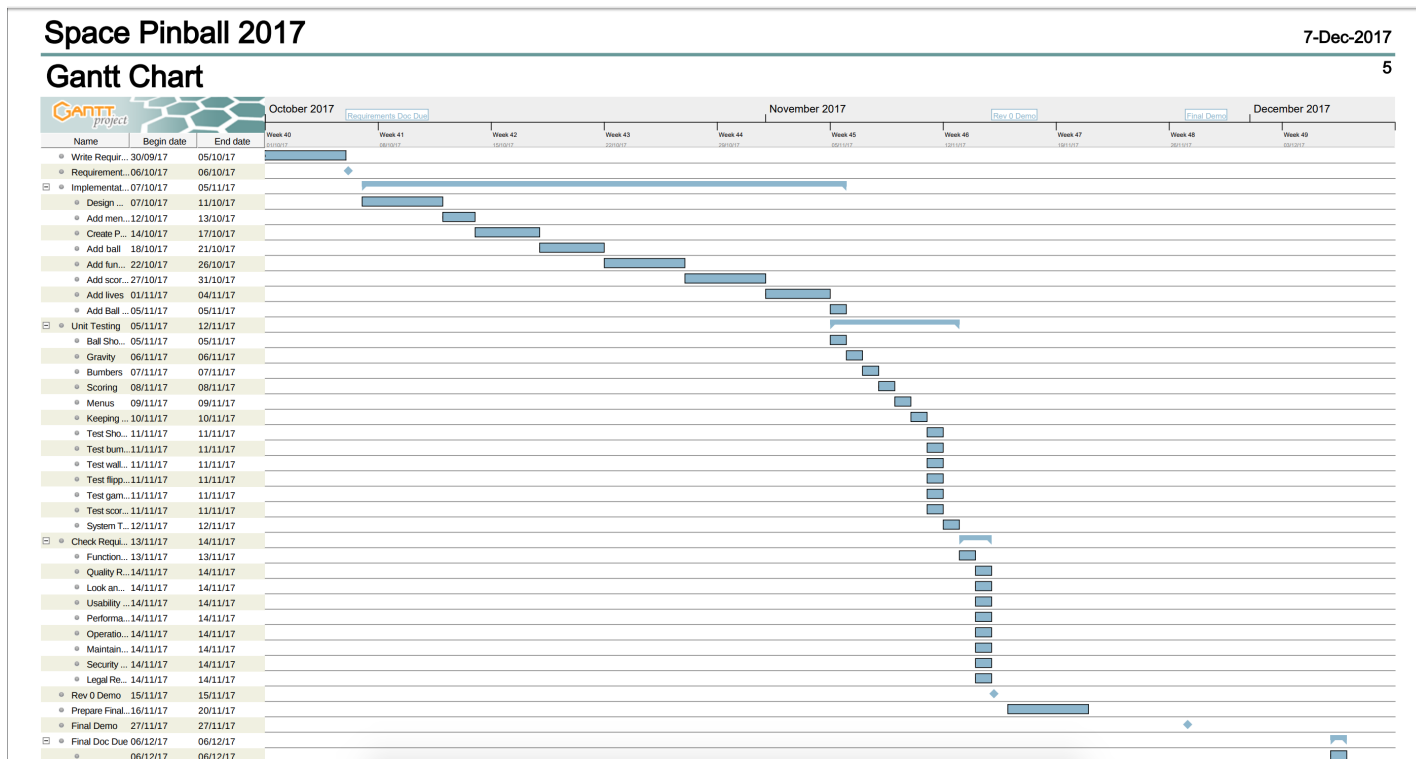


Figure 1: Gantt Chart

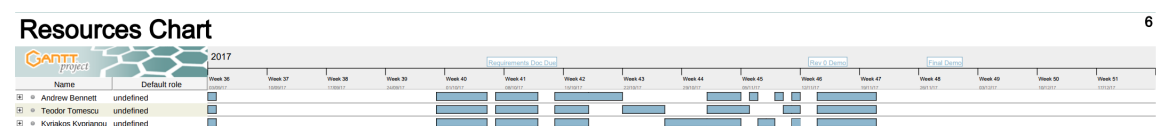


Figure 2: Resources Chart