

Ball In

TECHNICAL DESIGN DOCUMENT

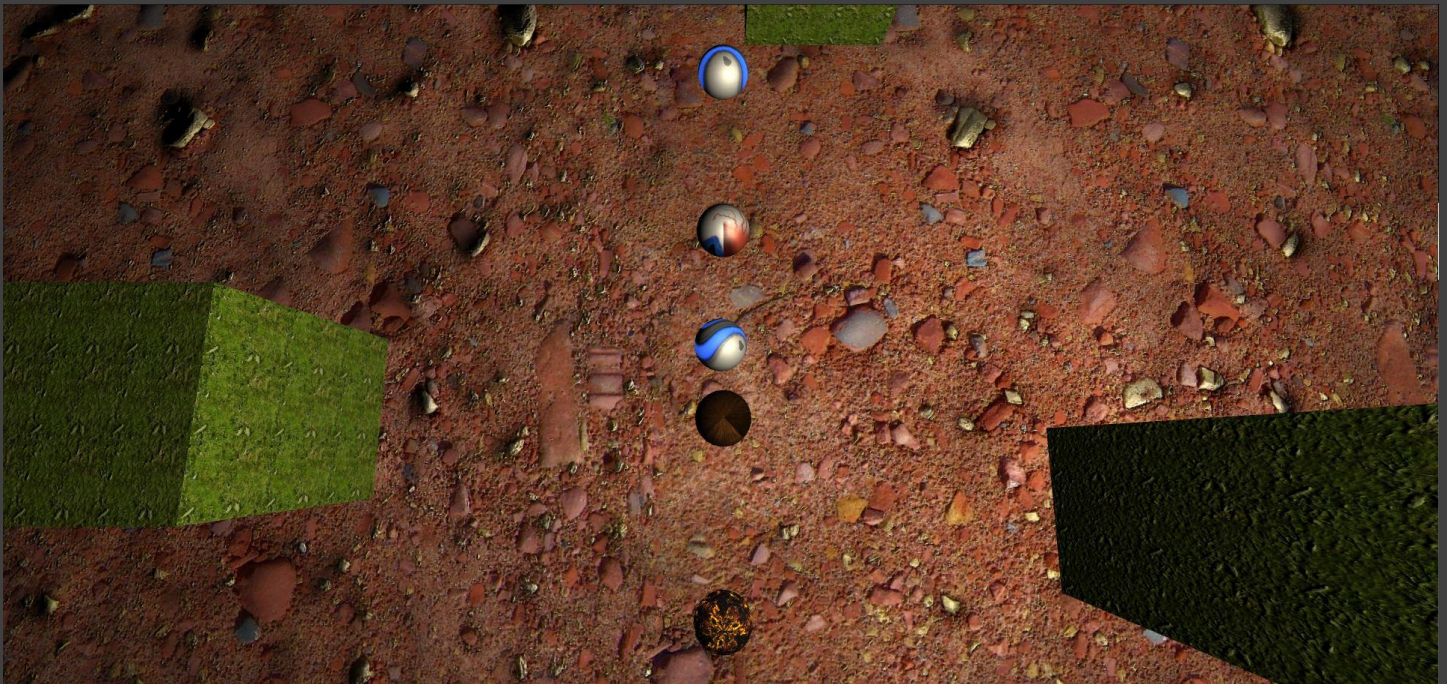
Team 01

Alexa Becerra

Cameron Yokoi

Peter Contant

Rodney Alacon



Contents

1. Executive Summary
 - a. Game Overview
 - b. Technical Summary
2. Equipment
 - a. Hardware
 - b. Software
3. Evaluation
 - a. Game Engine
 - b. Target Platform
4. Scheduling
 - a. Development Plan
 - b. Milestones
 - c. Updates, Maintenance & DLCs
5. Work Environment
 - a. Remote Collaboration
6. File Formats & Naming Convention
7. Levels
 - a. Level 1
 - b. Asset List

Executive Summary

Game Overview

Ball In is a multiplayer game where each player attempts to win as many mini games as possible. Each mini game will result in points earned. Points earned vary between placings. For example, first place receives the most points and last place receives the least. Whoever ends up with the most points after all mini games have been played is the winner.

Technical Summary

The prototype for Ball In will be available for play testing in roughly 1 month by 4 people using the Unity game engine. All assets and prefabs used were either retrieved from the Unity asset store or created using the default scene editor.

The game will be deployed for PC exclusively.

The minimum requirements include:

PC, MAC & LINUX STANDALONE

OS: Windows, Mac OS, Linux

Equipment

Hardware

Members of the team will utilize their personal workstations to develop Ball In. If time allows, usage of mobile phones may also take place to test for mobile platform support.

Software

The software used to develop Ball In will be the following:

- Main Development Platform: Unity editor & Visual Studio Code
- Communication & Collaboration: Discord
- Source Control & SDLC: GitLab

Evaluation

Game Engine

Unity was chosen by default as the primary tool and platform for game development as directed by the course syllabus. The prerequisite course, CS583, also mandated Unity so this advanced course follows suit accordingly. Lastly, Unity is a free to use game development platform up to a certain point of which we should not need to be concerned of reaching.

Target Platform

For the prototype, Ball In's target environment will be PC. By default, PC is the easiest platform to develop for since it also serves as the development environment. If time permits, alternatives such as mobile will be explored. Since majority of the team is using Windows, Android would be addressed first.

Scheduling

Development Plan & Milestones

Event	Date
Scrum - Action Items: <ul style="list-style-type: none"> Brainstorm Project 	02/11
Scrum - Action Items: <ul style="list-style-type: none"> Determined Idea - Mini Games Begin brainstorming mini games. Use Mario Party as example Think of assets to use/create for obstacles 	02/25
Scrum - Action Items: <ul style="list-style-type: none"> Defined which course capabilities to fulfill (multiplayer, flocking, procedural content, etc.) Begin listing game features by priority 	03/04
Scrum - Action Items: <ul style="list-style-type: none"> Created base project and developed some basic movement/camera features Began creating actual tasking for mini games. Each team member in charge of a mini game 	03/11
Scrum - Action Items: <ul style="list-style-type: none"> Worked on basic initial items for scenes (mechanics, asset gathering, lobby, etc) 	03/18
Scrum - Action Items: <ul style="list-style-type: none"> Made more progress on scene mini games. Mini games are now able to be played somewhat 	03/25
Scrum - Action Items: <ul style="list-style-type: none"> Continued progress on mini games. Bug fixes, game improvements, workflows coming together 	04/08
Scrum - Action Items: <ul style="list-style-type: none"> More bug fixes, additional levels within mini games created, will work on way for each mini game to be considered "finished", resulting in going back to the lobby 	04/22
Milestone: Game is ready for play testing. Can load game and play through various mini games from beginning to end	04/26
Play Testing	04/27
Scrum - Action Items:	04/29

Work Environment

Remote Collaboration

Due to remote courses as a precaution for COVID, all work and meeting sessions will be done online either through zoom or discord. All work will be facilitated using SDSU's canvas tools as well as GitLab on Rijeka for development and issue tracking.

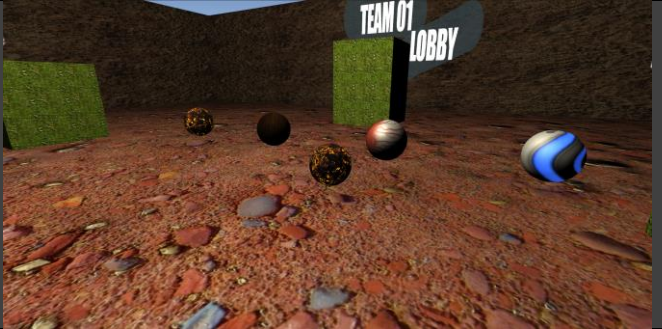
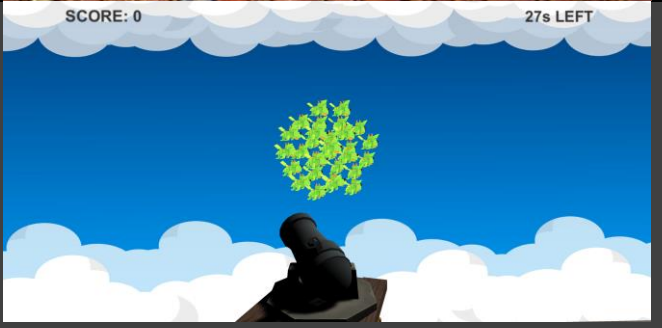
File Formats & Naming Convention

Asset Type	SubType	Naming Convention	File Format	Annotations
Animations		AnimationClipName	In the Model FBX	Move
Textures		imagename_purpose	PNG to FBX	
Scripts	Controllers UI	GameController ScoreKeeper	C# C#	
Materials	Characters Meshes	CharacterMaterial mesh_material	*.mat	Materials come from many different sources so naming conventions will vary
UI		UIElementState	PNG	

Levels

Ball In will consist of a lobby and four levels. Each level will represent a mini game where each player will try to finish in first place.

The overall game will support 1 to 4 players.

Level & Description	Image
<p><i>Lobby</i></p> <p>This is the location where all players will initially appear and will have the ability to join the various mini games by moving near the respective entrance. Entrances are in various areas within the lobby and are shown as doors.</p>	
<p><i>Cannon Boids</i></p> <p>Description: Cannon Boids is a game where each player will control a cannon. The cannon is meant to fire cannon balls aimed towards the moving targets. The targets themselves exhibit flocking behavior, making them hard to hit.</p> <p>Goal: The player with the most targets hit with the least number of cannons fired wins.</p> <p>The game ends when either no targets are left, or time has run out.</p>	
<i>Alexa's Game</i>	
<i>Cameron's Game</i>	
<p><i>Peter's Game</i></p> <p>Description: _____ is a game where each player needs to get through each mini putt level in the least amount of hits. Each level is complete when the ball enters the hole.</p> <p>Goal: Whoever reaches the end with the least amount of hits wins.</p> <p>The game ends when all players have reached the end.</p>	