|  |
| --- |
| Technical Design Document |
| Coffee Mania |
| By William Tjang and Pasqual Fletcher |

|  |
| --- |
|  |

Contents

[1.0 Revision History Version 2](#_Toc11750920)

[2.0 Development Environment 2](#_Toc11750921)

[2.1 Game Engine 2](#_Toc11750922)

[2.2 IDE 2](#_Toc11750923)

[2.3 Source Control procedures 2](#_Toc11750924)

[2.4 Third Party Libraries 2](#_Toc11750925)

[2.5 Other Software 2](#_Toc11750926)

[3.0 Game Overview 2](#_Toc11750927)

[3.1 Technical Goals 2](#_Toc11750928)

[3.2 Game Objects and Logic 2](#_Toc11750929)

[3.3 Game Flow 2](#_Toc11750930)

[4.0 Mechanics 2](#_Toc11750931)

[5.0 Graphics 3](#_Toc11750932)

[6.0 Artificial Intelligence 3](#_Toc11750933)

[7.0 Physics 3](#_Toc11750934)

[8.0 Items Item 3](#_Toc11750935)

[9.0 Game Flow 3](#_Toc11750936)

[9.1 ‘Mission’ / ‘Level’ structure 3](#_Toc11750937)

[9.2 Objectives 3](#_Toc11750938)

[10.0 Levels 3](#_Toc11750939)

[11.0 Interface 4](#_Toc11750940)

[11.1 Menu 4](#_Toc11750941)

[11.2 Camera 4](#_Toc11750942)

[11.3 Controls 4](#_Toc11750943)

[12.0 Asset List 4](#_Toc11750944)

[13.0 Technical Risks 4](#_Toc11750945)

# Revision History Version

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Description** |
| 1.0 | 14 Oct 2019 | Initial Document |
|  |  |  |

# Development Environment

## Game Engine

Unity v2018.3.8f1

## IDE

Windows Visual Studios 2017

## Source Control procedures

GitKraken, Unity Collab

## Third Party Libraries

## Other Software

# Game Overview

## Technical Goals

<3d graphics, 60fps, Challenging AI etc.>

## Game Objects and Logic

<A list of logical elements in the game, i.e. door, button, pistol, ammo, light, bullet, wall, character etc. and description of their behavior and purpose>

## Game Flow

<description of what the player can do (actions) from the start menu to playing the game, through to hitting quit. Include how to win, how to lose, how the player is moved, and what programmer things might need to be considered>

# Mechanics

<A list of the core game mechanics. I.e., what the player can do and how they achieve this, and what these triggers in the game. For example, shooting enemies is a core mechanic in an FPS>

Moving

* Side to side controls
* Strafe: The player switches to an adjacent lane either left or right using the A or D keys
* Turns: The player turns 90 degrees to the left or right using the arrow keys <- or ->
* Jump: The player jumps a certain altitude using the SPACE key

# Graphics

3D environment, stylized art

# Artificial Intelligence

<Describe how AI works, i.e. state machine, fuzzy logic, GOAP. Describe the various behaviours and how they change behaviour, how do the ‘creatures’ in the game evaluate the world>  
<include diagrams/flowcharts showing decision making processes>

# Physics

<if needed>  
<What engine are you using, what features from it (spring? Colliders?) how will physics be handled for objects? (box or sphere collider for objects, capsule for player) need to record specific locations for any reason? Potential slowdowns and how to mitigate.>

# Items Item

<List of items you can pick up that can affect the player, and what they will affect, like ‘picking up the hammer (refer collisions above) adds 5 to the players attack attribute’. Include details on how items influence gameplay or AI logic.>

|  |  |
| --- | --- |
| **Item** | **Description** |
| Coffee Bean | Adds 1 currency to the player’s overall account |
| Sugar | Speeds up the player character and grants invincibility for 5 seconds |
| Ice Cubes | Slows down the level speed for 10 seconds (may be subject to future changes) |
| Net | Allows the player to catch all nearby coffee beans in an AoE |
| Barista Knife | Allows player to ignore the next obstacle they crash into and destroy it |
| Milk | Grants a score multiplier buff |
| Tea | Inverts the player’s control for 10-15 seconds |

# Game Flow

## ‘Mission’ / ‘Level’ structure

<Are all levels stored in memory? what data is saved across levels, are levels loaded synchronously to prevent pauses?>

## Objectives

<What does the player try to accomplish on each level/mission? How is the players progress evaluated?>

# Levels

<If any of the Levels require specific behaviours, describe those here>

# Interface

## Menu

<What are the menu options and what do they do?>

## Camera

<Describe the camera, how it moves, perspective/orthographic, can it switch? How? Does it need to render-to-texture? does it prevent itself going through walls, use flowcharts to document behaviour>

Camera View: 3rd person perspective, positioned slightly higher than player, angled slightly downwards

## Controls

<Keyboard, tablet touch/swipe/tilt, joystick, mouse etc. record double taps, multi touch, use mouse smoothing/ scale mouse for aiming etc.>

# Asset List

<List all files needed, along with known attributes >

# Technical Risks

<if you want your game to be a 1000 player pvp battle royale with 4k 120fps graphics, you need to say if this is doable and how you intend to do it>