ANJIES ARCADE TDD

# FOCUS AREAS

## LEVEL DESIGN

ARCADE ROOM:

The models and lighting will represent how the rooms feel as well as the music in each section of the arcade room.

Each arcade machine will have floating UI set in world space canvas in form of spatial UI that I learned from research.

There will be bright neon lighting using the material emission feature and game noises in the background. These will be attached to objects that each sound relates to.

## UI

MENUS:

The main menus will use a world canvas and panels to navigate through the options such as the settings and controls screen. From research having UI set in the world makes it more immersive and interesting. Each button will have its own function that leads to different panels and menus.

SCENE MANAGEMENT:

The scene management will be controlled in one script that holds all the functions for switching between the scenes including the transitions for them. It will use Unity’s built in Scene Manager library.

HUD:

For the main player HUD there will be an empty object attached to the player which will manage all the UI as it will be controlled in world space to give the UI a floating effect instead of stuck to the screen. The players interact indicator will also appear next to the object in world space, instead of the screen in the 2D canvas space.

# Main character

MOVEMENT:

The characters movement will use the horizontal axis to allow the character to move forward and backward. The player will have the main camera attached to them and can rotate the camera left and right to change the direction in which they are moving and to also change the camera direction.

Animation states:

A drawing of a speed line

Description automatically generated with medium confidence

INTERACT:

This feature will primarily be used as a means to access the main functionalities of the games, it will use colliders to detect when the player is realistic range to interact with an object. Once in range a UI popup will appear to indicate this to the player and allow them to press a button to start the interaction.

# MINIGAMES

For all 3 minigames the mechanics will be very similar and so can be condensed to improve code efficiency and not repeating same logic over and over.

Collectibles:

The collectible functionality will be the same for each minigame. The items will use colliders to determine if it has been touched by the main player, doing so will increase a score integer that is stored on the player. This will display the players current score on the UI.

Scoring:

The scores will be saved using the built-in PlayerPrefs system within Unity.

## Maze

MOVEMENT:

The movement for this game will use Vector3d and use the main horizontal and vertical axis for movement, the characters velocity will remain in a constant state so that the player doesn’t speed up as they move further.

## FLY AWAY

The player for this game will have 3 health points which will be stored in the main player class as a variable. When the play loses health points to enemies, it will go down and the player will lose the game and have to restart.

MOVEMENT:

The is a 2D game so the movement will use the Vector2D axis to move the player around the screen.

ENEMIES:

Enemies will start by slowly following the player around the screen, it will use the MoveTowards() function and remain at a set speed. As the game goes on the enemy speed will slowly increase which will make it harder for the player to stay alive.

When the player gets hit by the enemy the collider will trigger and the player will lose a health point.

## ENDLESS RUNNER

MOVEMENT:

The movement for the character will start at a set speed, a timer will be used to determine how fast they go, the longer the player stays alive the faster the player moves which increases the difficulty of the game.

COLLISION:

When the player hits a collider, in this context will be an obstacle on the track the game will end.

# AUDIO AND VISUAL DETAILS

GAME SOUNDS AND MUSIC:

Music for the game will be attached to an empty object in the scene that will handle all the background audio. Each room in the main arcade room will all have its own collider, depending on which room the player is stood in a different soundtrack will play to represent the room they are in

Each minigame will have different sound effects will play to represent the action taken such as collectibles being hit and collected and enemies hitting the player.

# Game Engine

## Unity

The game will be made in Unity 2022.3.4f1 as it provides long term support and engine documentation.

# Hardware/Software requirements

The game will be PC only.

* CPU: Intel Core i3 3210 or equivalent
* RAM: 4 GB RAM
* STORAGE: 2GB for base files
* GPU: Intel HD Graphics 4000 or AMD Radeon R5 series | NVIDIA GeForce 400 Series or AMD Radeon HD 7000 series
* OS: 64-bit Windows 7 or later
* Screen Resolution: [1920](https://www.minitool.com/lib/1024-768.html) x 1080