

Kune Kune

Technical Design Document

Developer: Alicia Hayashi



Table of Contents

Kune Kune	1
Change Log	3
Development Environment.....	3
Software requirements	3
Game Overview	4
Technical Naming Conventions	5
Game Flow and structure.....	6
Controls	8
Behaviour/AI	9
Pseudo Code and Flowcharts.....	10

Change Log

V.0 – 17/05/23 – Document created; software requirements documented.

V.0.1 – 24/05/23 – Updated the naming convention list with examples.

V.0.2 – 7/06/23 – Created flowcharts and wrote pseudo code, cleaned up formatting and added evaluation for scripting language choice.

Development Environment

Software requirements

Software	Version	License
Unity	2021.3.13f1	Free
Visual Studio 2022	2022 version	Free
Autodesk Maya 2023	2023 version	Educational
Source Tree	2023 version	Free
GitHub	2023 version	Free

Game Overview

Game Name	Kune Kune
Platform	PC
Genre	Horror, Maze Running Game
Target Audience	15–25-year-old male PC gamers. Players who enjoy the thrill of outrunning a monster. Love Japanese urban legends and games with an eerie vibe. Have played similar games before e.g., Dark Deception
Perspective	3D First Person (no hands or feet seen)

KuneKune is a **maze escape thriller game** in which Players are tasked to outrun a Japanese urban legend the 'KuneKune' while simultaneously navigating a hedge maze to escape. To escape the maze, Players must find three mini figures that represent each KuneKune and throw it down the well to stop their hunt.

Technical Naming Conventions

- All **variables** within code will use the **camelCase** naming convention

E.g., `int [] playerHealth`

- All **functions** within code will use the **PascalCase** naming convention

E.g., `int SumTwoNumbers(int a, int b)`

- All **game objects** within Unity will use the **PascalCase** naming convention

E.g., **MazeWall**, **SpawnSpotLight**

- All file names will use the naming convention

Name_Typeofdocument

E.g., **KuneKune_LDD**,

Game Flow and structure

Game Objectives

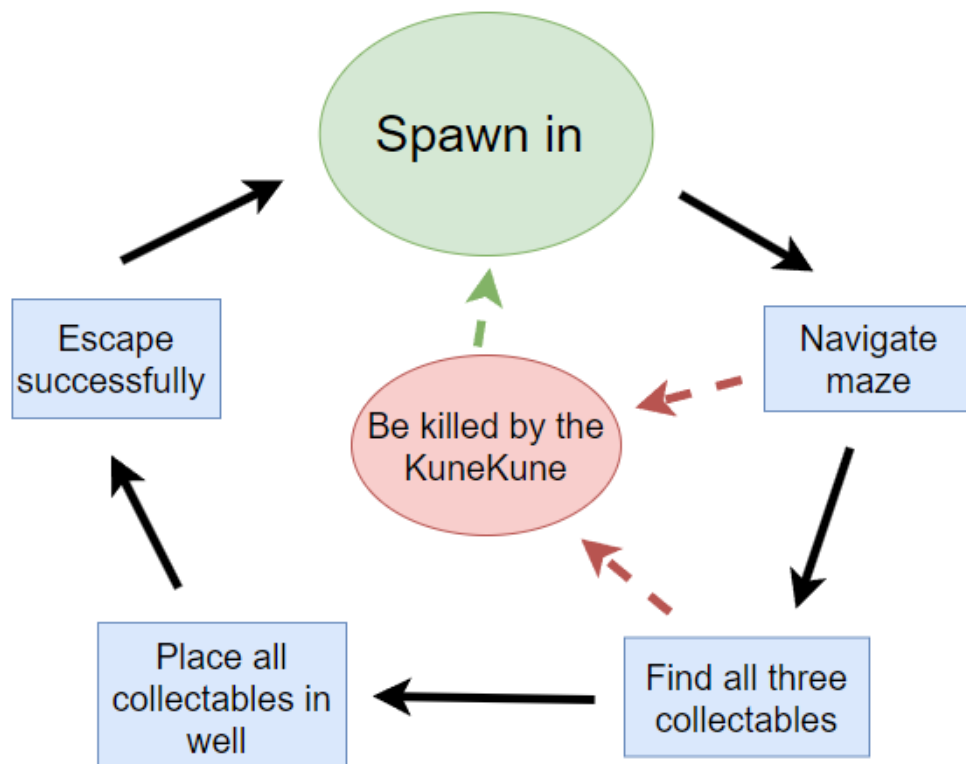
The objective of the default game mode is for the Player to find the three figures within the maze and drop them down the well to kill the KuneKune.

Level Structure

Game Loops

A single game of KuneKune may look like this.

1. Spawn in
2. Navigate the maze
3. Be found and killed by the KuneKune **OR** navigate to the collectables, place them in well and escape.



Feature List

Mechanics List

- Running
- Changing direction
- Jumping
- Picking up/ Putting down a figurine

Potential Scripts needed based on game genre and mechanics

- *First person controller* (Basic movement, sprinting and camera manoeuvrability)
- *Pick-up script*
- *Basic AI enemies* (Follow player, wander state)
- *Various Trigger scripts* (Trigger win, trigger UI, trigger audios/animations)

E.g. Trigger Win script will be placed on a trigger box so that Players must place all three figures in one spot for them to load the winner's scene. Thus, this script cannot be a component of the FPS controller and needs to be a separate script to another trigger script like 'Trigger Death Animation'.

C#

This project will be scripting in C#.

This is due to C# being the only language that is natively supported by Unity, the engine that will be in use for this project.

C# is also an object-oriented scripting language, which becomes useful when scripting for game development.

Controls

- **W, A, S, D** (Movement forward, left, backward and right)
- **Space** (Jump)
- **Mouse movement** (direction player is facing)
- **E Key** (interact e.g. pickup/drop collectables)
- **Shift** (Sprint)

Behaviour/AI

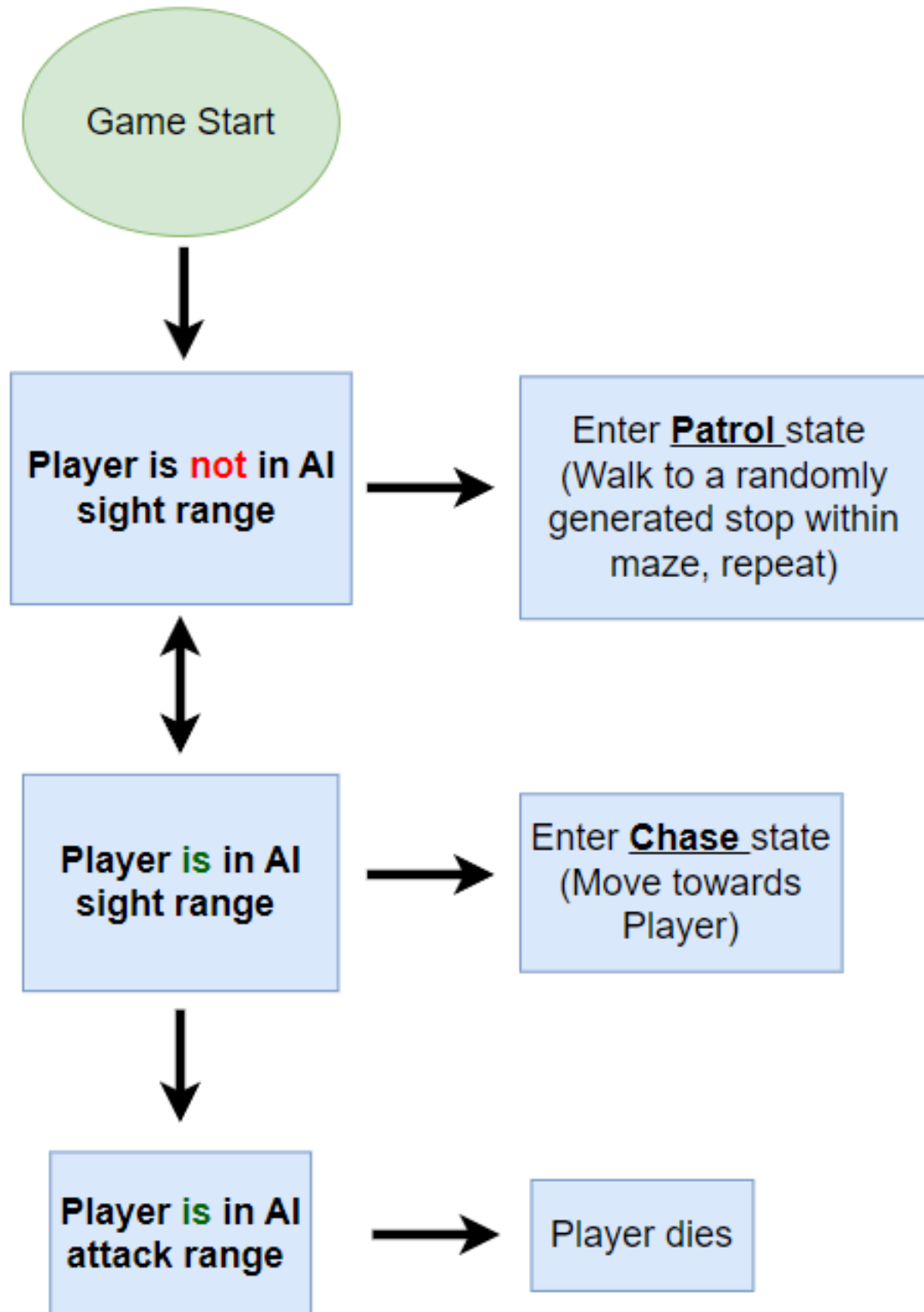
The AI within KuneKune will remain extremely basic but still replicate the feeling of an intelligent monster chasing the Player down.

The different gameplay states that the KuneKune AI monster will flow between include:

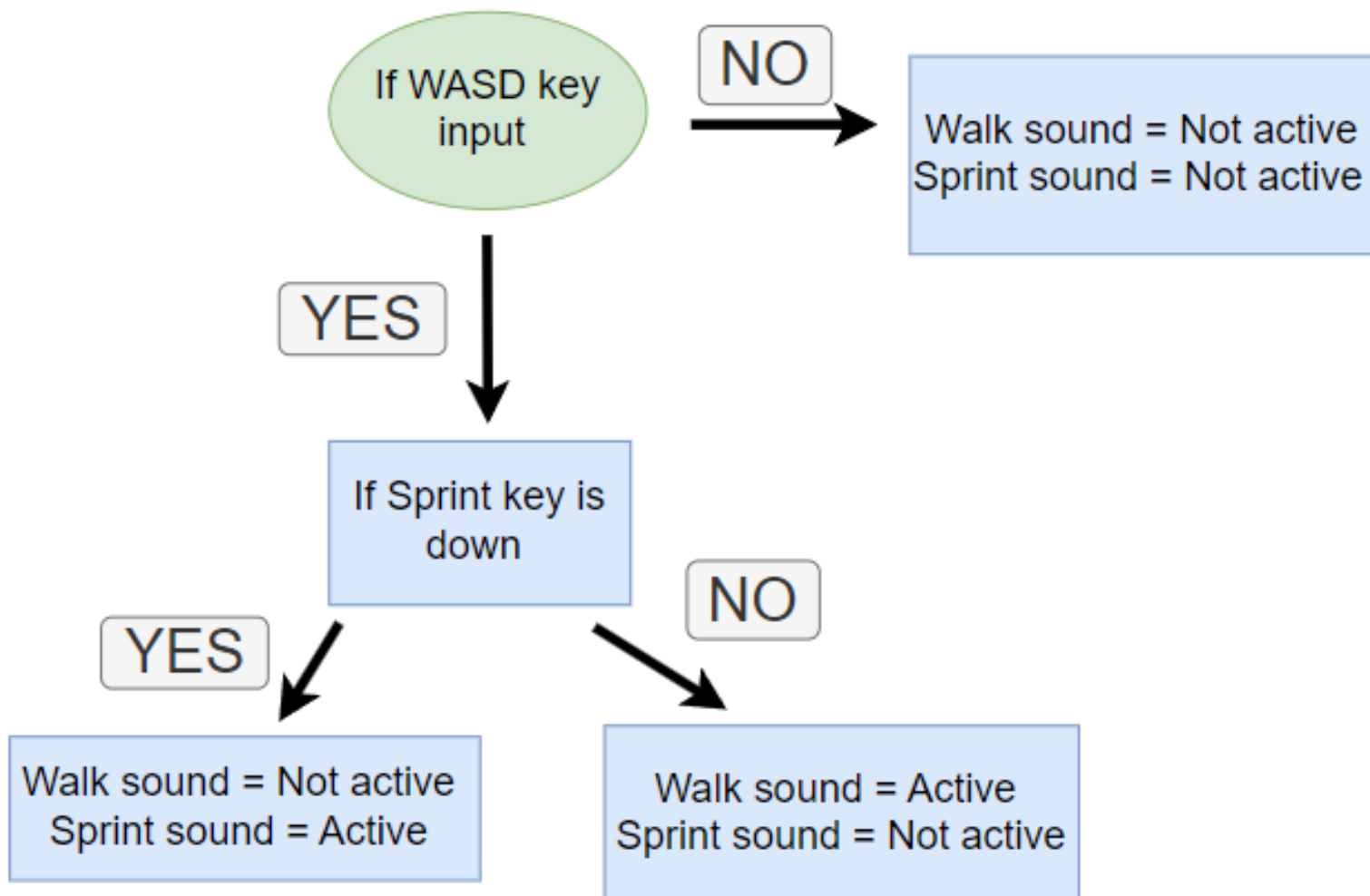
- **Patrolling** (The player is currently not within the range of the AI, it will continue to roam around the maze in search of the Player)
- **Chasing** (The Player has entered the AI's range and the AI now begins to chase down the Player, finding the quickest route from their current location and the Player's location)
- **Killing** (AI is close enough to attack and kill player (AI can one-shot the Player to death))

Pseudo Code and Flowcharts

Enemy States (Flowchart)



Running Audios (Flowchart)



Win Game Trigger (Pseudo Code)

Set all figures to false

On collision

- If (tag name is figure 1)
 Set figure 1 to true
- If (tag name is figure 2)
 Set figure 2 to true)
- If (tag name is figure 3)
 Set figure 3 to true)

If (All three figures are true)

 Load winner scene

Win Game Trigger (Flowchart)

