Version 1.1 2024/25

University of Staffordshire

Prototyping for Technical Designers

Technical Design Document

Sarah Abbas

B00007I@student.staffs.ac.uk

Contents

Project Introduction

This project is a prototype of an arcade-style racing game as stated within the 'Racing Project Brief'. This project aims to fulfil the project brief given within the 8-week time limit.

The project called 'Star Kartz' is an arcade styled racing game set in outer space. It would allow players to race in space going through the stars. The player can complete three laps on the current track with the help of booster pads.

Project Goals

The goal of the project is to provide players a feeling of satisfaction in mastering the controls of the kart and complete the three laps of the track. The physics and feel of controlling the kart is important to achieve this; the easy to learn and hard to master arcade style.

Challenges and Risks

The main challenge centred around the physics and controls of the player's kart. Following a tutorial will be necessary as a kart game hasn't been development before now. This is a huge risk as the development would take longer than usual.

Another challenge when working on the project was the time trial aspect. I was struggling greatly with making the countdown timer show only the seconds with no decimal numbers. This was solved by using a different node instead of 'Time Seconds to String'. This gave me the option to change what the UI text showed in game. Furthermore, I was able to get the player to not move by changing the player's speed modifier instead of changing the player state, this was done as the player is a pawn and not a under a player character actor blueprint. However, I couldn't get the countdown timer and regular timer to correctly initialise in game. To not have the countdown timer eat into the player's timer, I had the player timer start in the negatives so that the player can move as soon as the timer goes into the positive.

Hardware Requirements

Listed below are the recommended requirements for the tools:

- CPU: Intel® Core™ i7-11700
- GPU: NVIDIA GeForce RTX 3080
- Memory: 32GB
- Storage requirements: 719.9 MB

Please note that the above data comes from the PC that this game was developed on. There has been no testing done on hardware with different specs.

Platforms

Target Platform

This project will be used on any desktop (Windows, Mac etc.) that is able to run video games. This game is not designed or formatted to be played on any other platforms.

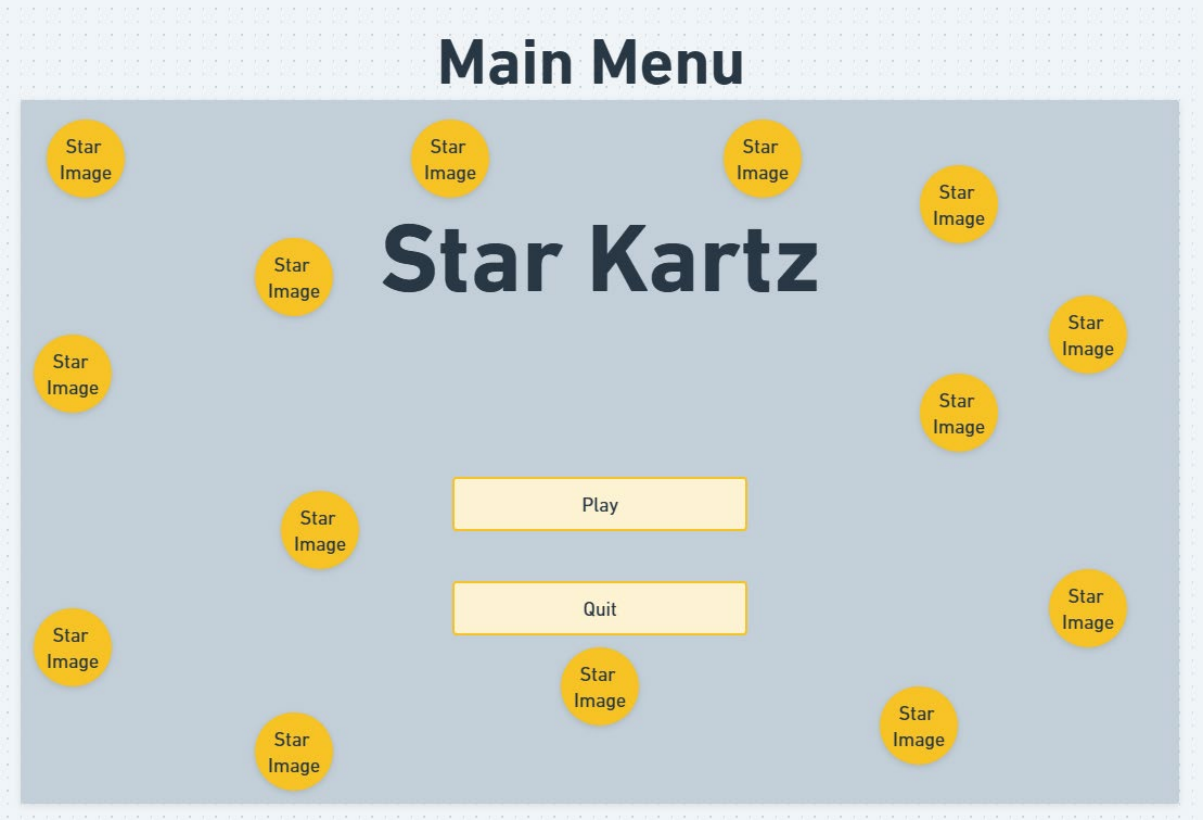
Engine Specific Specifications and Limitations

There would be possible limitation regarding the disk space limitations. The graphics and art style are purposely low poly to allow lower spec PC to play through the game. This project isn't available on console or other platforms of play.

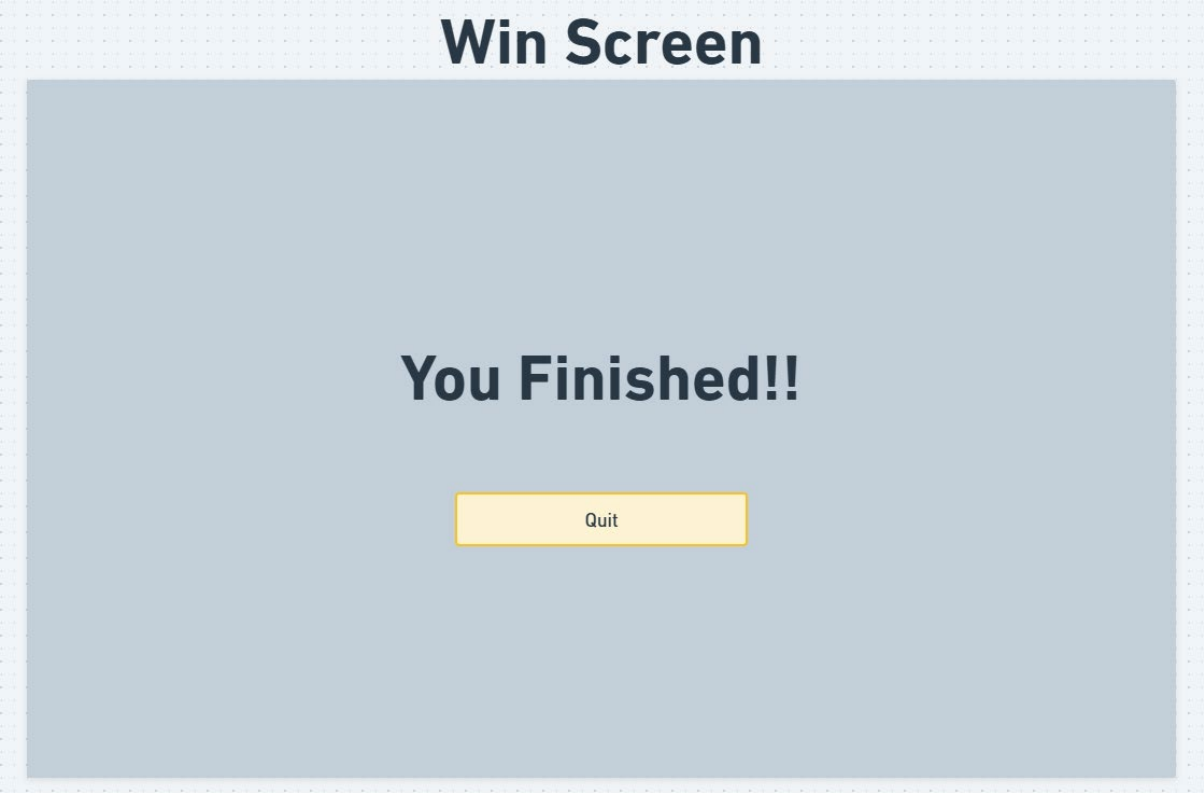
Systems and Diagrams

UI Wireframes

Main Menu

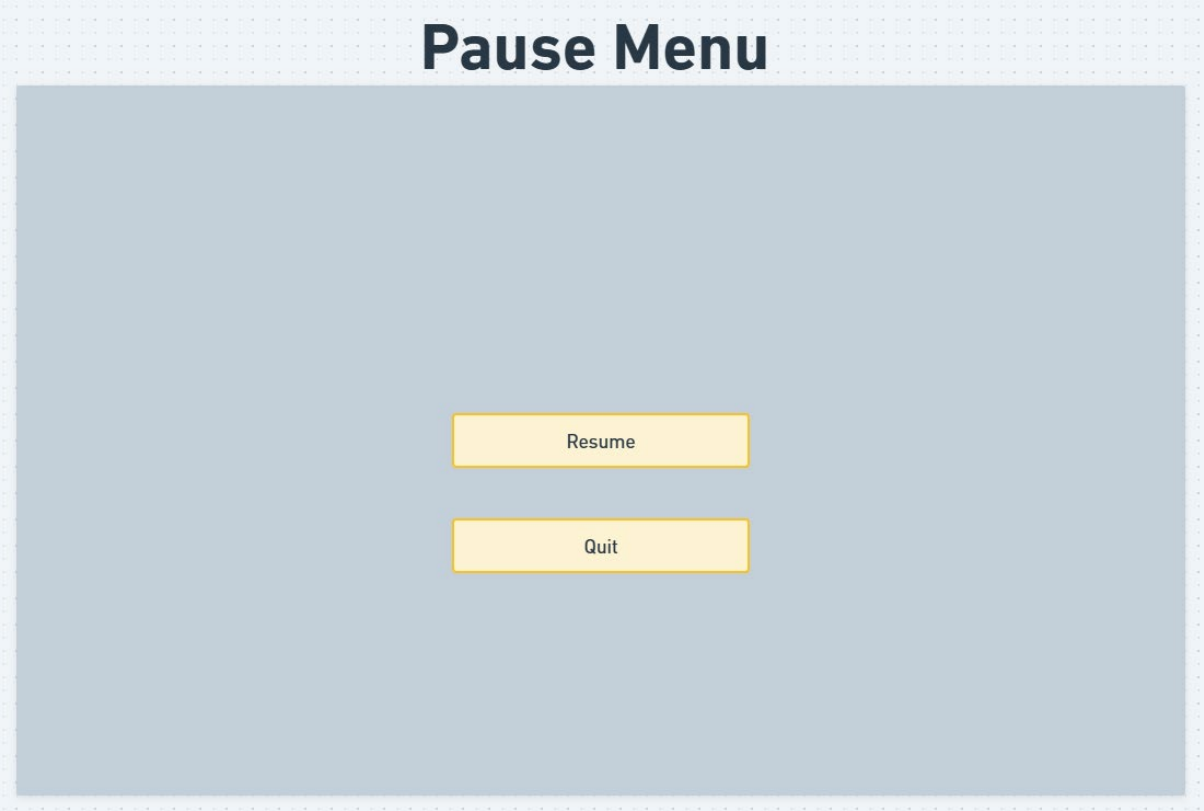


Win Screen



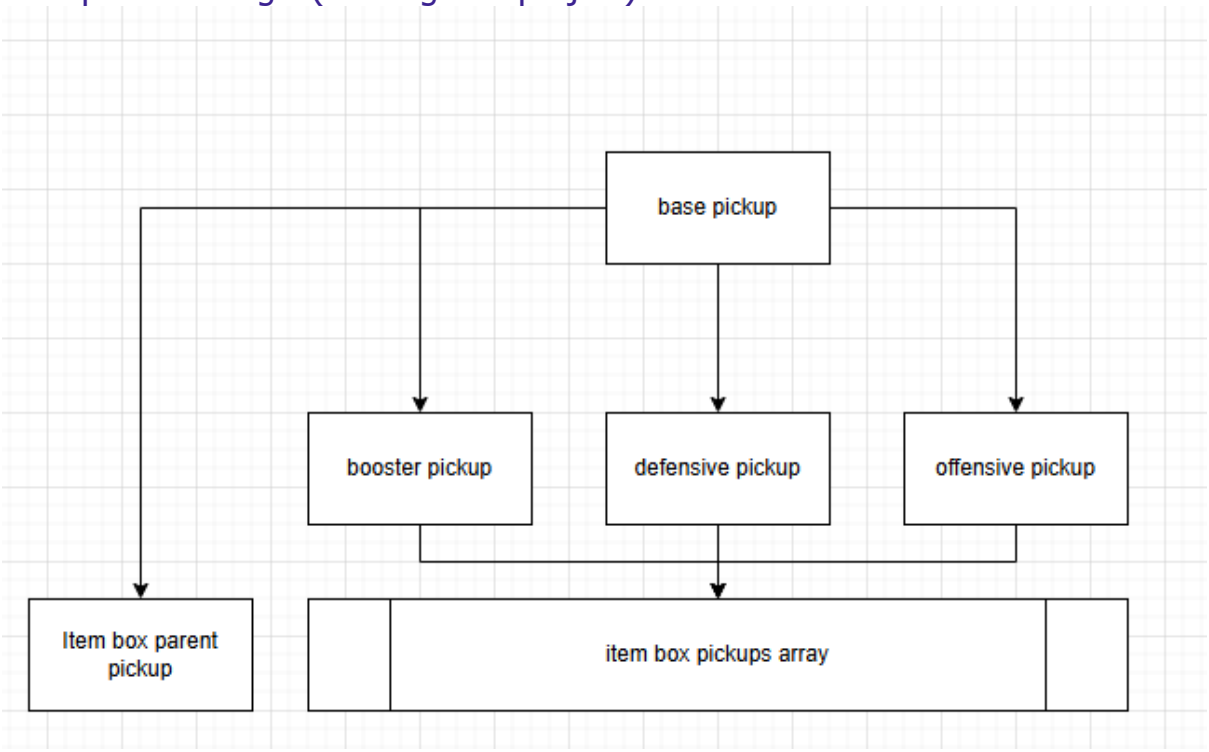
Player HUD

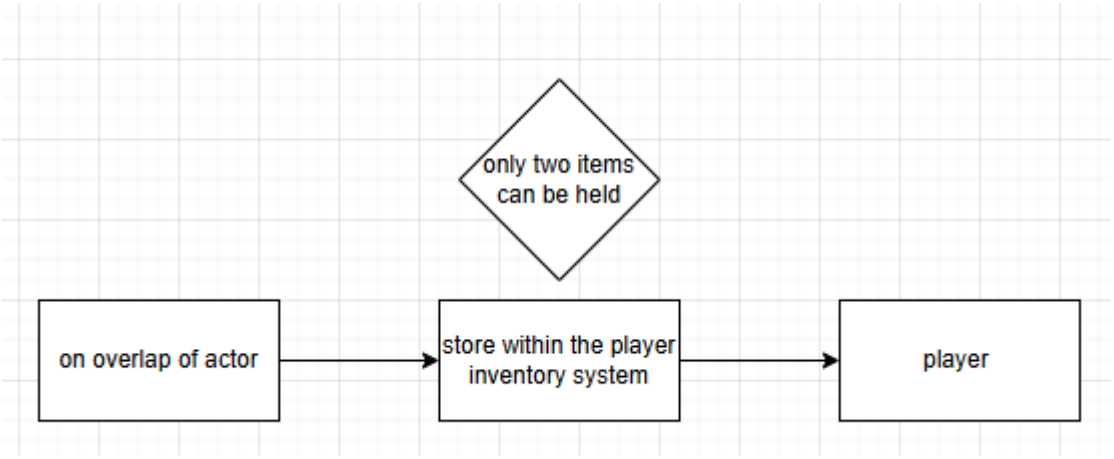
Pause Menu



Blueprint Logic Flowcharts

Pickup Parent Logic (no longer in project)





Optimisation and Profiling

Profiling Systems

Testing sessions will occur during the 4th week of development in which testers will be given the game to play and then afterwards be given a Microsoft form to fill out to collect feedback regarding their play session thoughts. This will allow further improvement to the game as the feedback will be used as a foundation for what features need to be added/improved upon.

This has been outlined further in the digital academy forum post regarding the game's development.

Coding Standards

Programming Standards

There will be a strict naming convention for the assets created. The asset name will include the recommended prefix (taken from this guide) and then pascal case will be used to name the asset. For example, a material would be named, **'M_BaseMaterial'**.

Style Guide

All blueprint functions and complex codes will be commented to allow further developments in the future. Colouring the comments will be done to make readability and differentiation for further developments in future. Adhering to the strict naming convention will be necessary to make the program more understandable for future use.

Production Overview

Moscow

Must	Should	Could	Won't
Drivable kart for player to control	Item boxes to change up gameplay and make it more fun	Multiple different tracks to race on	Multiplayer
A single track to race on	A boost panel to change up the speed of the kart	Ability to hold more than one item at a time	
Gameplay loop – racing around the track against a timer/lap completion	A timer to go against alongside a lap counter	AI karts to race against players	

Green = done || Red = Not Done