## horizontal line



Parkour Racer

Game Design Document

## World Jam Entertainment

Brent Baker, Danny Coyle, Alex Edgar, TJ Hofferberth, Nicolas Re

# Table of Contents

## Overview

## Progression Milestones

## Game Mechanisms

## Artificial Intelligence

## Networking

## Game Elements

## Game States

## Game Progression

## Bibliography

# 

# Overview

3D World Jam is a singleplayer and multiplayer racing game featuring parkour movement elements. Players will use parkour moves and special abilities to traverse levels with paths of varying difficulty, trying to get to the finish line faster than their opponent(s). Conserving and maintaining your momentum is key to keeping pace and ensuring your success.

## Goals

1. Fast-paced and competitive racing experience
2. Rewarding for both new and experienced players
3. Seamless local and online multiplayer play

## Specifications

1. This game uses the Unity3D Engine Version 5.4.0f3
2. Programmed using C# and Unity’s built-in libraries
3. Multiple levels of play that feature varying difficulties

# 

# 

# Progression Milestones

## Timebox 1

3D Pong

Menu System

Loading Screens

## Timebox 2

Player Controller

Beginning Level Design & Brainstorming

Improved Menu System

## Timebox 3

Improved Player Controller

Sounds

## Timebox 4

Multiplayer

AI

Race System

## Timebox 5

Fully Traversable Level

Improved Multiplayer

Improved AI

## Timebox 6

Improved Multiplayer

Improved Lighting

Improved AI

Finish Levels

Implement Speed-Warp

# Game Mechanisms

## Control Scheme

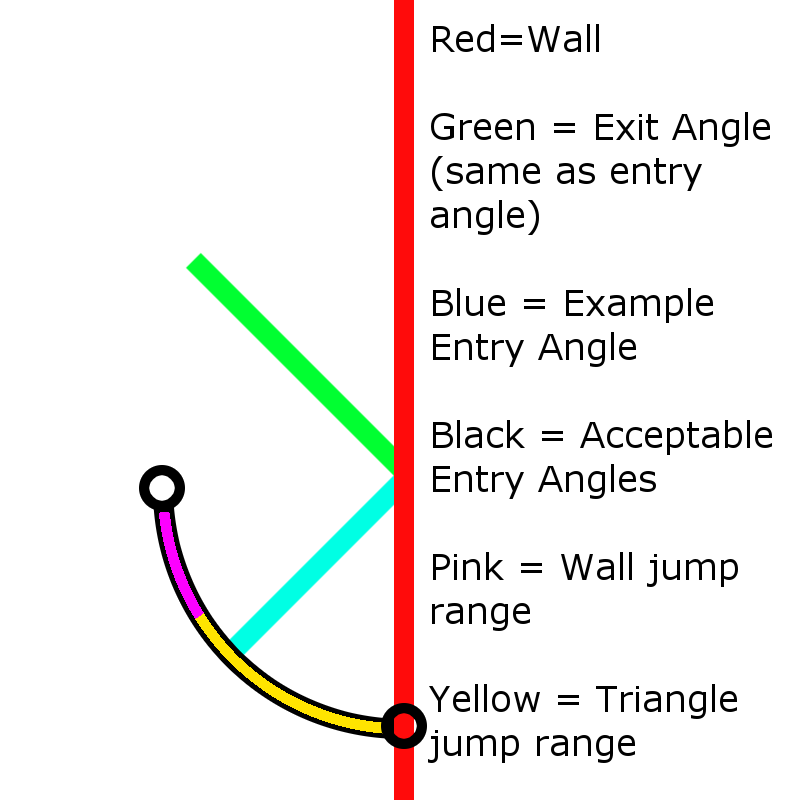
|  |  |  |  |
| --- | --- | --- | --- |
| **Keyboard** | **Xbox Gamepad** | **PS Gamepad** | **Action** |
| WASD | Left Analog Stick | Left Analog Stick | Player Control |
| Spacebar | A | X | Jump |
| Ctrl | B | Circle | Slide/Stomp (in air) |
| Shift | X | Square | Dash |
| Q | Left/Right Bumper | Left/Right Bumper | Speed-Warp |

## Control Details

The player will travel along the ground by using the player controls listed above. To travel over walls or pits the character can jump and use their dash to gain a short burst of momentum to make it over the obstacle. The direction of the burst is influenced by the players controls and direction. To go under a wall or other obstacle the character can perform a slide while running on the ground. If the player wants to stop forward momentum while in the air he can stomp and go straight down.

Wall jumps occur when a player hits a wall while in the air and the direction they are facing is between 150 and 180 degrees away from the normal of the wall. If these conditions are meet then the player can jump again before landing. If the player wanted to go up to a higher location when they find themselves between 2 close walls they could wall jump to get on top of the 2 walls. Wall jumps will cause the player to reflect off the wall but gain a large amount of momentum in the upward direction.

Triangle jumps occur when a player hits a will while in the air and their forward direction is between 91 and 150 degrees away from the normal of the wall. If these conditions are meet then the player can jump again before landing. Triangle jumps will cause the player to reflect off the wall but gain a large amount of momentum in the direction of the reflection.



A top down diagram explaining triangle jumps and wall jumps for when the player is jumping into a wall.

## 

## Race/Lap System

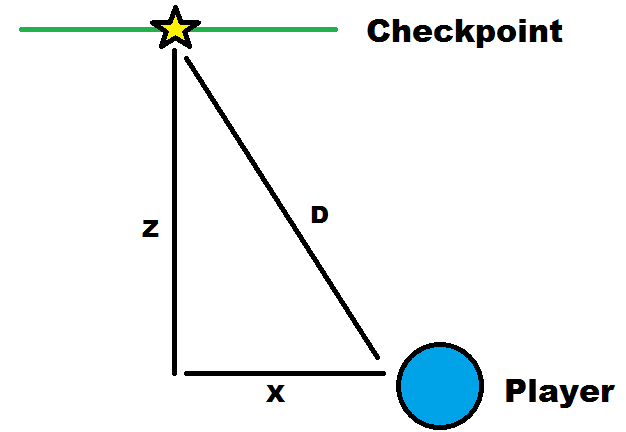
The race position system is used in multiplayer since the only enemy racers in singleplayer are a single AI opponent or a replay ghost of the player’s best time. The algorithm involves 3 main steps to determine where the players are placed amongst each other. The algorithm compares 3 attributes of the player’s position against the attributes of their opponents.

**Position Algorithm**

These steps are repeated for each opponent in the race. The player is initially assumed to be in 1st

1. Determine which lap the current player is on, if it is...
   1. Greater than their opponent → End
   2. Less than their opponent → Increase their standing by 1
   3. Same as their opponent → Go to step 2
2. Determine which checkpoint the player just passed, if it is
   1. Greater than their opponent → End
   2. Less than their opponent → Increase their standing by 1
   3. Same as their opponent → Go to step 3
3. Determine the distance to the next checkpoint, if it is...
   1. Greater than their opponent, increase their standing by 1
   2. Less than their opponent, end

The distance calculation for measuring to the next checkpoint is done using a point to point calculation, ignoring the Y-axis. A simple Pythagorean Theorem calculation is used to determine the distance based just on the player and checkpoint positions.



## Level Design Goals & Ideals

* Be mindful of the speed and distances of the player’s movements
* Consider combinations of movements/abilities
* Minimize stops and slowdowns
* When possible, implement “fail-safe” routes for beginners
* Make paths and their advantages as clear and noticeable as possible
* Allow for multiple means of traversal over the same section
* Encourage traversal methods through route variety
* Discourage/punish backtracking
* Offer situational choices based on timing or race position
* Encourage players to interact and cross paths often through the level design

## Speed-Warp

To help aid players in keeping races competitive and close players have a catch up mechanic known as “speed-warp”. Located in the bottom left of a player’s screen is a bar known as their speed-warp charge. Once a player’s charge has reached 100%, player’s can activate their speed-warp for an increase to their maximum speed for a short amount of time. To keep player’s closer together and thus keep races more competitive player’s charge will increase faster based on their current place in the race. For example, 1% every second for 1st place, 2% every second for 2nd place, 3% every second for 3rd place, and 4% every second for 4th place.

# Artificial Intelligence

## NPCs

Players will have the option to race against computer controlled opponents which will have all of the speed and abilities that the player possesses. This will give the player the competition of a normal race without the pressure of racing a real opponent and will allow the player to improve their skills for multiplayer.

## Real-time Replays

The game will automatically save the player’s fastest run for each map which the player can race against in real-time. A ghost will follow the player’s saved run which the player can race against in an attempt to create a new record by coming in first. This allows players to see specifically where on each map that they can most improve their time on.

# Networking

## Lobby Networking

The multiplayer lobby network is setup using the networkmanager which is a default asset in unity. The multiplayer UI is setup to call functions from the networkmanager to allow for players to connect or host games. When the players join a lobby only the host can start the vote timer, the vote timer is 5 seconds. Once the vote timer has started the players vote on the maps and the one with the most votes wins. Once everyone has finished the map the server waits 5 seconds and then transitions everyone back to the lobby.

## Player Networking

The players synched through the networkmanager have their position and rotation automatically synced by attaching the network transform to it. The player's color is synced across the network by having the color be based off of the players netID (a unique number to identify the object on the server). The players animations are synced by having the other factors synced across the network. These factors are weather the player is sliding, dashing or grounded and their current speed.

# Game Elements

## Player Character

The player or players will each control a single character. The character is a blocky humanoid with animations for running, jumping, sliding, dashing, and idling. The player character emits light for increased visibility of opponents in multiplayer.

## Levels

Each level will be designed with a specific theme and level of difficulty in mind. While each level is traversable, some are more forgiving to falls from the ideal path than others.

## Obstacles

Some of the levels have obstacles scattered throughout. Our various obstacles include moving blocks/platforms, fans, and electrical shock traps.

## Race System

The race system keeps track of player positions throughout the course, displaying their current place amongst the racers. The system records the player’s time to complete the level and displays their best times in the level select screen.

# Game States

States.png

# Game Progression

There is a tutorial course in place as the first map that allows players to learn and become comfortable with the game mechanics and controls before venturing off into the other three levels, which are listed in order of difficulty.

# 

# 

# Bibliography

Engine - Unity3d.com

Art Work

* Atey Majeed Ghailan
* Josan Gonzalez
* Tiago de Silva
* S. Jamal.K
* Von Schlippe
* Tarek
* Sora Valtieri

Sound

* Little Robot Sound Factory
* Nectar Sonic Lab

The Ohio State University, CSE 5912