DC9 Technical Document

Pelle Bruinsma - CMV1G
389066
Programming for real time (DC9)
1st year CMD - Game Design
12th of June 2020
stupidplusplus.com

Table of contents

A short introduction

I dislike phaser, for a few reasons. The main reason being that it's poorly documented. The second reason being that personally I feel there's better game engines out there (especially web based) than phaser. And finally that we are swinging in between programming languages and engines instead of allowing students time to build up a level of expertise with one of them. So far we have seen development in gamemaker, p5.js and now phaser. I would've preferred taking on multiple projects with just one of these in mind.

Obviously though it's hard to please everyone and especially someone that's addicted to the Unity engine. So I still tried my best to make this game a fun experience for me to make and you to play. Now let's get started.

Game Concept description

Welcome to ***Super 1v1 Wizard Soccer (featuring dog)!*** Or in short: ***SWS.***

SWS is a modern variation of the game Pong from 1972. In case you don't know pong, in Pong players face off against each other in what could best be described as a simplified and digital version of table tennis or better known by ping pong.

SWS attempts to reform pong to have faster and more randomized gameplay that makes the game much more unpredictable and have a more party-ish feel to it.

I made these changes specifically to make this concept feel more refreshing and fit a quick play format better. Some examples of these changes are the dog, the randomly spawned static objects (palm trees) and the small random movement of the ball.

When designing this game I took inspiration from games like Undertale (visuals wise) and Lethal League.

Especially lethal league I found to be an interesting source of inspiration and this can definitely be seen in the game in most notably how the ball seems to get faster every time it bounces off of something.

Game elements

I will now list all game elements derived from the game concept. This list will contain all the elements that need implementing in code and will be accompanied by their properties needed for the game and the ways they interact with each other, the player and/or other game elements.

- **The player(s)**

(Sidenote: SWS is intended to be played by two players so there will be two player instances found in the game.)

Properties:

- Movement speed
- Score

Behavior:

The player element will only move when controlled by the user or when pushed away by other game elements.

Interactions:

This element interacts with the following:

- **The ball:** The player collides with the ball.
- **Other Players:** The player collides with other players.
- **The dog:** The player collides with the dog.
- **Static objects:** The player collides with static objects.

- **The ball**

Properties:

- Random movement speed
- Bounciness
- Velocity limit

Behavior:

The ball element will move randomly, when pushed away by other game elements or when bouncing off other game elements.

Interactions:

This element interacts with the following:

- **The Player(s):** The ball collides with the player(s).
- **The dog:** The ball collides with the dog.
- **Static objects:** The ball collides with static objects.
- **Invisible Goal Sprites:** The ball disappears and (when the game has not ended) respawns when hitting an invisible goal sprite.

- **The dog**

Properties:

- Random movement speed

Behavior:

The dog element will move randomly or when pushed away by other game elements.

Interactions:

This element interacts with the following:

- **The Player(s):** The dog collides with the player(s).
- **The ball:** The dog collides with the dog.
- **Static objects:** The ball collides with static objects.

- **Static objects**

(Sidenote: the pseudo-randomly generated environment consists entirely of this group and you will therefore see numerous instances of these objects)

Properties:

- None, these are static objects and therefore do not have any properties.

Behavior:

None, these are static objects and therefore do not have any behavior.

Interactions:

- None, these are static objects and therefore do not have any interactions.

- **Invisible Goal Sprites**

(Sidenote: SWS is intended to be played by two players so there will be two invisible goal sprite instances found in the game.)

Properties:

- None, these are static objects and therefore do not have any properties.

Behavior:

None, these are static objects and therefore do not have any behavior.

Interactions:

- **The ball:** The ball disappears and (when the game has not ended) respawns when hitting an invisible goal sprite.
- **The scoreboard:** When the ball is respawned, one point is added to the invisible goal sprite's other side of the scoreboard.

- **The scoreboard**

Properties:

- The scores

Behavior:

The scoreboard keeps track of and displays the score.

Interactions:

- **Invisible Goal Sprites:** When the ball is respawned, one point is added to the invisible goal sprite's other side of the scoreboard.

Description of process

Honestly when I started this project I had no clue what I wanted to do, so I browsed the internet in search of a cool game idea. I did not find one. So instead as sort of a hello world project I just decided to replicate pong using a spritesheet from kenney assets, who I'm sure you know. I used specifically the micro roguelike asset (https://www.kenney.nl/assets/micro-roguelike), but additionally also made a ton of assets myself and all the audio in the game is completely self made.

Honestly I just implemented everything as I went along and made this report afterwards. Honestly programming this wasn't as much of a struggle but I can list the steps I went through for this game:

- Hello world!
- Showing sprites
- Single player input
- Physics groups
- Created the ball
- Added drag to the ball
- Added second player and switched input method
- Started on terrain generation
- Made ball move slightly randomly
- Added bounciness to ball
- Limited ball's velocity
- Removed drag from ball
- Created dog
- Added dog Ai
- Created goal zones
- Created scoreboard
- Replaced dog Ai with random movement
- Made game replayable
- Added game logo
- Added sound effects
- Added music
- Added home screen controls and website link
- DONE!!!!!!!!

Alrighty well I hope you enjoyed playing SWS and enjoyed reading through my arbitrary code comments. Seeya!

Thank you for reading.