Fireball

Technical Documentation

Written by:

Finn Perry

Team:

* Design:
  + Glittika Germandee (Nancy)
  + Douglas Tilden (Richard)
* Programming:
  + Finn Perry
* Art:
  + Tiarna Kilmister
  + Wing Him Choi

Table Of Contents

[**Game Overview**](#_rd4u6olxsw3z) **4**

[Basic Overview:](#_k6p3v29cqxgk) 4

[Genre/Theme:](#_offgwfgrhexi) 4

[Overview Of Play:](#_i6491cdztshm) 4

[Overview Of Key Mechanics:](#_rsh5f3300yh6) 4

[Overview Of Camera And Controls:](#_63vgnmong017) 4

[**Development Environment**](#_iaua3vlu12tc) **5**

[Engine:](#_txlseyk9x46c) 5

[Target Platform:](#_pvwek2tcbkp8) 5

[Source Control:](#_4eocr5c8gy05) 5

[Asset Implementation:](#_aopifog9fmsp) 5

[**Software Requirements**](#_gi9swwmknafs) **6**

[Unity3d:](#_qkf02qcsl397) 6

[Visual Studio:](#_ykm7kdn5r8et) 6

[Photoshop:](#_47fvurvje84k) 6

[Maya:](#_1ivzgv8iuum4) 6

[Zbrush:](#_wa6gjbz4z8c0) 6

[Substance Painter:](#_i66qu5g7udkv) 6

[Github Desktop:](#_m4h1ixy9ftn9) 6

[**Mechanics**](#_66ib47fzl3kl) **7**

[Movement:](#_749qmval9in3) 7

[Dragons:](#_tzs7xt85ueyq) 7

[Critters:](#_p429mtbg859n) 7

[Game Timer:](#_dwts9067lvb0) 7

[Points:](#_41z3rv87mqpb) 7

[**Coding Standard**](#_wqav759pgai9) **8**

[**Player Update Flowchart**](#_sz2cad2dlcsj) **9**

[**Critter Update Flowchart**](#_5tzkr9a2gwrm) **10**

# Game Overview

## Basic Overview:

An arena party game for 4 players where players control baby dragons to run around and shoot fireballs at each other and at small creatures (critters).

## Genre/Theme:

The theme is cute, the dragons are babies that are just learning to shoot fire. The critters look cute and fluffy too.

## Overview Of Play:

Players control their dragons with xbox controllers (or similar joysticks). The dragons can shoot fireballs at each other and at the critters to get points.

## Overview Of Key Mechanics:

Players are scored with a points system, killing a dragon rewards 3 points and killing a critter rewards 1 point. When a dragon dies it becomes stunned for a few seconds. During the stun the dragon cannot move or attack. Once the stun ends the dragon has it’s full health restored so the player can continue playing.

When a critter dies it has a chance to drop a random powerup. Players can pick up different power-ups to receive temporary buffs. Some power-ups provide a stat boost for a few seconds such as increasing movement speed. Some powerups give the dragon a single use special attack that replaces their standard fireball attack.

## Overview Of Camera And Controls:

The camera stays in a fixed location looking down on the arena from a slight angle.

The players control their dragons movement using the left joystick on the xbox controller, the dragon attacks when the ‘A’ key is pressed on the xbox controller.

# Development Environment

## Engine:

The engine that will be used is Unity3d. Unity handles feature such as controller axis inputs and rendering, allowing more time for developing the gameplay.

Unity also allows building to many platforms which will be useful if the game is ported to consoles.

## Target Platform:

The target platform is pc because developing for pc is easier to test and the team is most comfortable with developing for pc. PC is also easier to build to as consoles require more optimisation and some systems may not be compatible with certain consoles.

## Source Control:

Github will be used for source control as the Github desktop client is intuitive to use so team members who have not used source control before can start using it quicker.

## Asset Implementation:

New art and audio assets are placed into their respective folders and uploaded to source control. Once the asset is complete and it has been reviewed by other team members it can be imported into Unity.

All assets are uploaded to the main source control branch.

# Software Requirements

## Unity3d:

The game engine that will be used is Unity because most team members are familiar with the engine and members who aren’t have experience in other similar software.

## Visual Studio:

Visual Studio will be used for code editing and debugging. Visual Studio will be useful because it supports breakpoints for debugging while testing the game in Unity.

## Photoshop:

Photoshop is used by artists to create concept art, textures, UI, etc.

## Maya:

Maya is used by artists to create meshes, rig models and animate models.

## Zbrush:

Zbrush is used by artists for modeling organic shapes and fine detail.

## Substance Painter:

Substance Painter is used by artists to create textures.

## Github Desktop:

Github is used for source control for the whole team.

# Mechanics

## Movement:

### Dragons:

Dragons can move forwards/backwards/left/right in the arena.

The arena is always flat ground to the dragon controller is always locked to the same y position to prevent entity collisions from pushing them up in the air.

### Critters:

Critters can be in an idle state or a wander state. A critter in the idle state will stay still for a period of time. Once the time is up the idle state ends. A critter in the wander state will attempt to move to a random position in the arena.

If a critter in the wander state reaches its destination or it takes too long to reach its destination then the wander state will end.

When a critters current state ends it randomly picks a new state. If the new state is idle then it randomises an amount of seconds to idle for. If the new state is wander then the critter will find a new random position in the arena to go to.

## Game Timer:

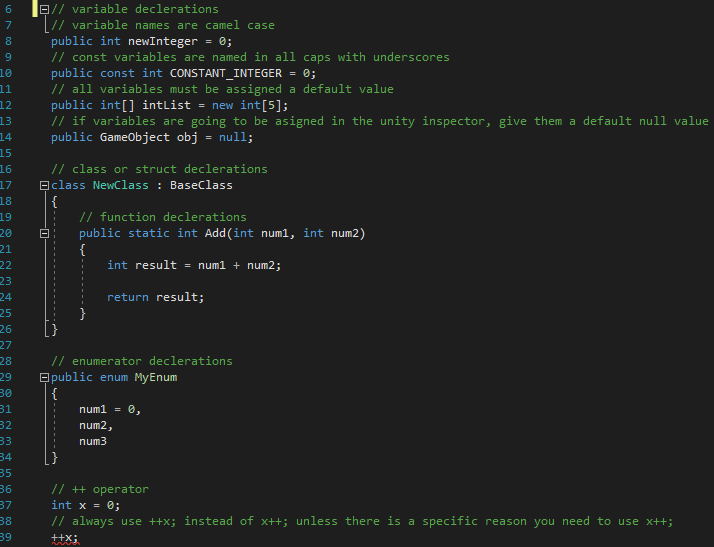
The game runs for a set amount of time, all players continue playing for the entire time (except if they get stunned for a few seconds). Once the timer reaches zero the game ends and a score screen is displayed showing how many points each player earned.

## Points:

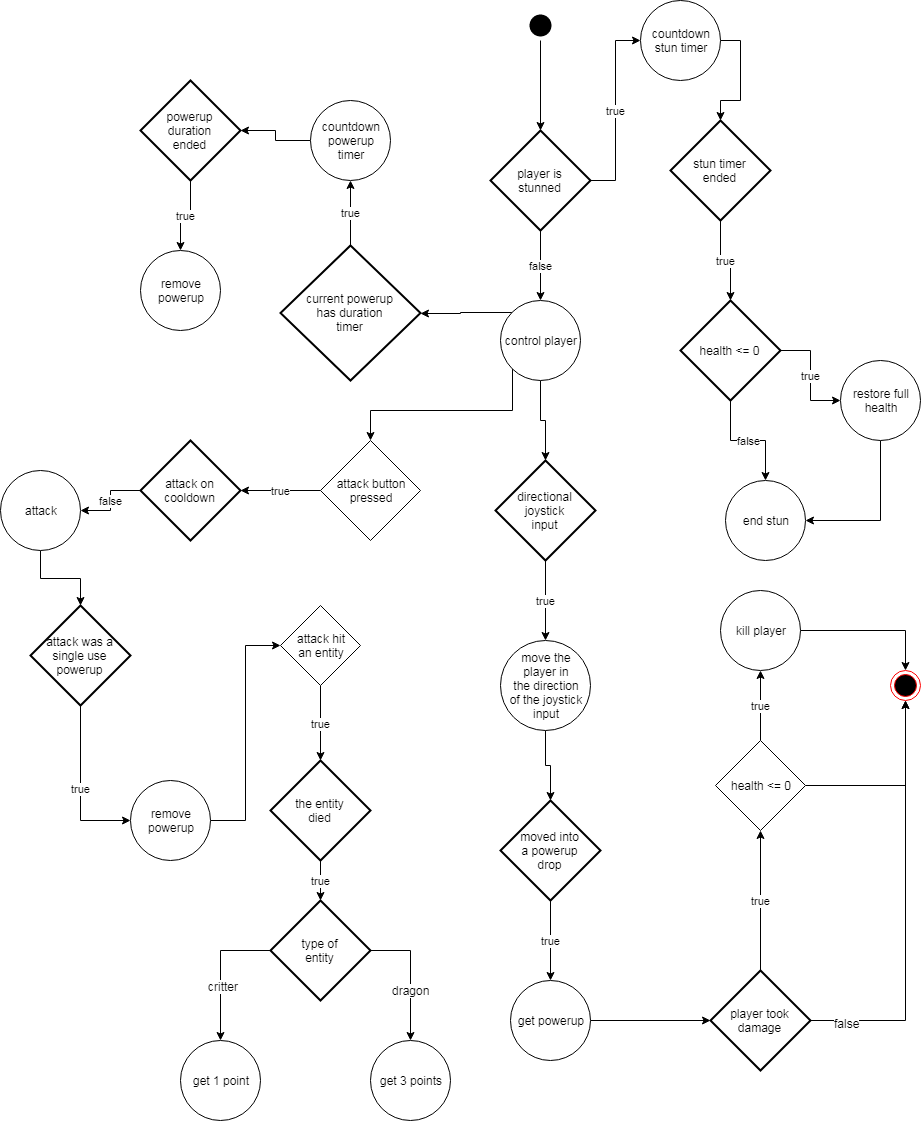
When a player kills another entity they will receive points. Killing a dragon rewards 3 points, killing a critter rewards 1 point.

There is no score limit, all players continue to earn points until the game timer runs out.

# Coding Standard



# Player Update Flowchart



# Critter Update Flowchart

