Technical Design Document

Dash n’ Smash

Adam Frewen – 0813664

Table of Contents

Logic and Algorithms

- Description of logic and technical algorithms required to implement game features

Painters algorithm

- Assists with sorting 2D sprites depth order.

Tiling algorithm

- The arena layout will be created based on input file and stored in an array. This array will them be draw in the playable area. Platforms with collision will need to be identified.

Jump algorithm

- Jump algorithm to calculate jump height and how it is affected by gravity.

Collision algorithm

- Circle vs circle algorithm to detect collision between player characters, player attacks, Power Orbs, and arena platforms. A quad tree will be utilised.

AI movement algorithm

- Algorithm to handle sporadic Power Orb movements. Movement will guide the Power Orbs on a random path through the map. Power Orb must to more than just float around a single area. Pathing nodes may be required.

AI spawning algorithm

- Orbs. Spawning locations will be predefined in relevant sections of the map. The algorithm will decide where to spawn AI while maintaining a random, but even spread.

Data Structures

Quad tree

- To handle collisions.

Abstract data type

- To store different entities.

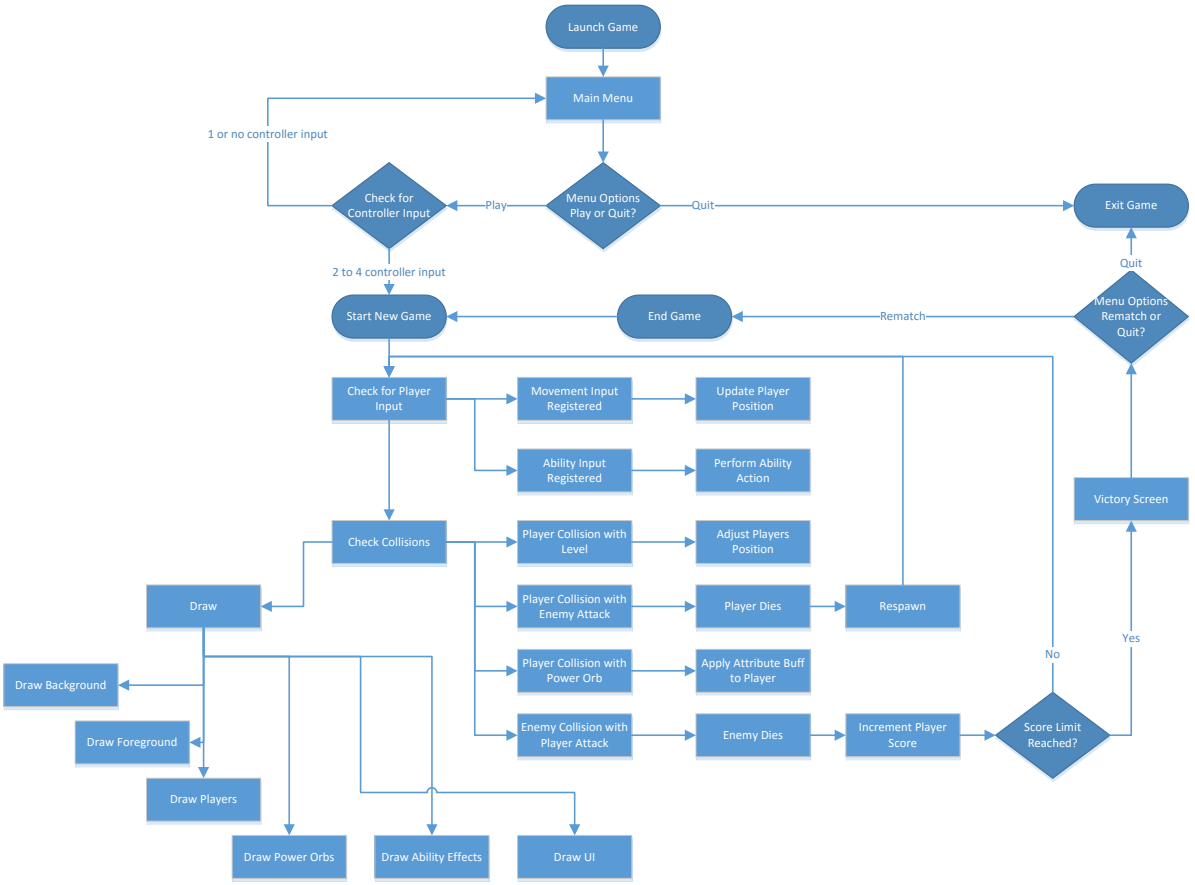
Array

- Storing the arena tiles.

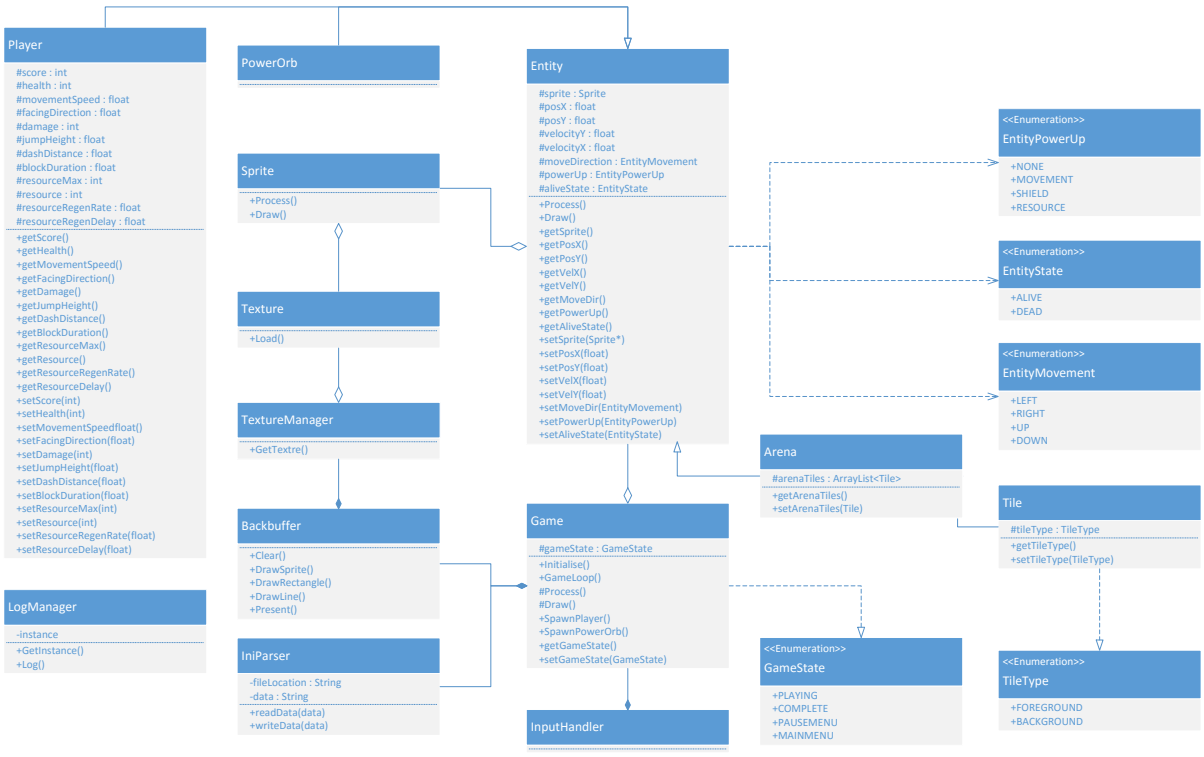
Queue

- Player revive order.
- Power Orbs preloaded and stored, ready to be released during the match.

Flowchart



UML Activity



File formats

- .png
  - PNG format will be used for all sprite sheets and 2D assets.
- .mp3
  - MP3 format will be used for all game sound effects and music.

Debug Features

Frames per second display	FPS display to help track FPS drops during.
Reduce game speed	Manual reduce the tick speed so gameplay can be observed in slow motion.
Game pause	Live pause (without UI clutter) to allow aspects of the game to be inspected while stationary.
Collision display	Visual display of collision. Collision boxes should change colour when collision is detected.
Power orb spawn timer	Spawn timer showing the duration remaining until next Power Orb spawns.

Naming Schemes

Camel casing will be utilised for all assets.

- Classes: ClassName
- Variables: variableName
- Enumeration: ENUM
- PNG: SpriteImage
- Audio: SoundFile

Testing Plan

1. Does player score update correctly in all cases?
   - Double kill. Both enemies killed at same moment with same ability.
   - Multi kill. Enemies killed in rapid succession.
2. Victory condition successful?
   - Game won when player reaches score limit.
3. Is collision performing as expected against all types?
   - Level, Player, Power Orb, Damage
4. Are any memory leaks present?
5. Flowchart scenarios able to be execute successfully?
6. Is game performance optimized?
7. Do Power Orbs provide correct improvements?
8. Are all assets loading correctly?
9. Do all player abilities function as intended?
   - Punch
   - Block
   - Dash
   - Jump
10. Does multiplayer function correctly?
    - 2 to 4 simultaneous players.
    - 2 to 4 simultaneous controllers.