**Technical Design Document**

**Basic Overview:**

|  |  |
| --- | --- |
| Genre: Arcade, share screen Battler Arena | Theme: Western, Mexican Stand off |
| Camera: Top-Down Isometric |  |

**Controller Input Keybinds:**

|  |  |
| --- | --- |
| **Mechanic** | **Keybind** |
| Movement & Directionality | Up, Down, Left, Right, Left Joystick |
| Shoot | R2, X, ■ |
| Reload | R1, O, ▲ |
| Molotov | L1 |
| Dash/Roll | L2 |

**Core Mechanics & Variables Data Sheet:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Mechanic** | **Description** | **Component Variables/Metrics** | | | | | | | | | | | | | |
| **Player Controller:** | | | | | | | | | | | | | | | |
| Combat System/ Economy | Health & Ammo statistics | **Health:** | | | 6 | | | | | **Ammo:** | | | 6 | | |
| Movement& Directionality | Forward, Back, Strafe Left, Strafe Right, 360 degree Turn | **Move Speed:** | | | 100% | | | | | **Turn Rate:** | | | Fast | | |
| Dash/Roll | Player dashes infront of the faced direction | **Displacement Distance:** | | | | | | 5-8yrds | | | | | | | |
| Default Shoot | Player fires a projectile infront of the faced direction | **Fire Rate:** | | | 0.5s | | | | | **Damage:** | | -1 | | | |
| Reload | Player reloads x6 ammo | **Delay/Channel:** | | | | | | 0.8-1sec | | | | | | | |
| **Collectables:** (One collectable weapon equip at a time) | | **Spawn time:** | | | | | | 12secs | | | | | | | |
| Revolver | Player fires two consecutive projectiles infront of the faced direction | **Fire Rate:** | 0.15s | | | **Active Time:** | | | 5s | | **Equip Time:** | | | | 5s |
| 0.3s | | |
| Cavalry Rifle | Player fires three simultaneous projectiles infront of the faced direction | **Fire Rate:** | 0.8-1s | | | **Active Time:** | | | 5s | | **Equip Time:** | | | | 5s |
| Molotov | Player throws a projectile area of effect snare | **Move Speed Decrease:** | | | | | | - 40% | | | | | | | |
| Bandage | Heal & movement speed increase | **Move Speed:** | | 40% | | | | | | **Heal Quantity:** | | | | +1 | |
| Parachute | Collectables are randomly spawned via parachute & descend with indications to their landing locations | **Float Speed:** | | -50% | | | | | | **Spawn Time:** | | | | 12s | |
| **Map Mechanics:** | | | | | | | | | | | | | | | |
| Ricochet | Projectiles rebound off terrain collisions | - | | | | | | | | | | | | | |
| **Play Mode Mechanic Iterations:** | | | | | | | | | | | | | | | |
| Sandstorm (1v1) | sandstorm descends decreasing the map area until it reaches the maximum decrease area | **Spawn Timer:** | | | | | 45secs -1min | | | | | | | | |
| Train (1v1) | the scheduled train pass through town obstructing projectiles & destroying projectiles upon carriage collision | **Spawn Timer:** | | | | | 30secs | | | | | | | | |
| Projectile Depletion Iteration | **1v1)** Projectiles depletes upon collision with map centre/rail track | - | | | | | | | | | | | | | |
| **FreeForAll)** Projectiles Depletes after 2secs |
| Projectile Interference Iteration | **1)** Projectiles deplete upon intersecting with other projectiles | - | | | | | | | | | | | | | |
| **2)** Projectiles Ignore interception with other projectiles |

**Development Details:**

|  |  |
| --- | --- |
| Platform: Console, PC | Engine: Unity |
| Source Control: GitHub | External Assets & Plugins: |
| File Share: DropBox | Scrum Software: Trello |

1. **Revision History**

|  |  |
| --- | --- |
| Version | Description |
| 1.0 | Initial Document |

1. **Development Environment**
   1. Game Engine

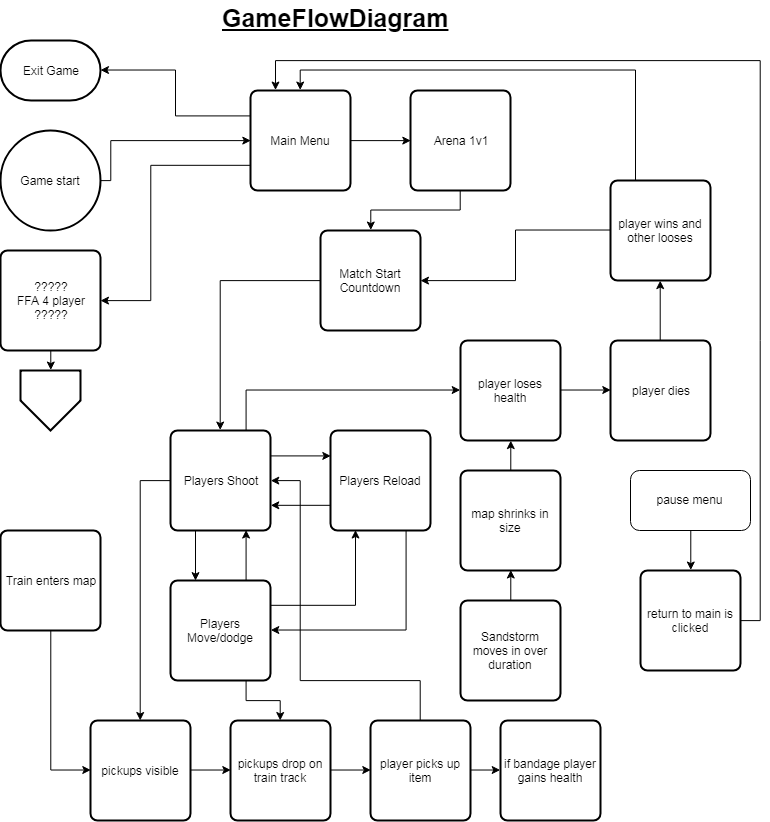
Unity 2017.3.0f3

* 1. **IDE**

Visual Studio

* + 1. Coding Guidelines:
* K&R Bracket Placement
* Simple and concise comments
* Block comment where needed
* Indent under functions and statements
* Curly brace all statements where possible
* Public data types should be placed above protected types and privates below them
* Group data types together (int’s with int’s, vector3 with vector3)
  + 1. Naming Conventions:
* Meaningful data names (what it is)
* Function names should be verbs or a name in relation to what the method does
* Name designed around readability
* Use of “camelCase”
  + 1. Source Control procedures:
* The use GitHub
* Multiple branches for Art, Programming, Design and Testing as to keep master for complete Builds
* Always keep a working Build
  + 1. Memory limits per system
  1. **Third Party Libraries**
  2. **Other Software**
  3. **TRC registration requirements**

1. **Game Overview**
   1. **Technical Goals**

* Responsive and fluent character control
* Optimisation
* Party arcade style share screen multiplayer
  1. **Game Objects and Logic**
* Revolver (default weapon), single fire, 6 round pistol
* Dual Revolver (Pickup), twin fast fire upgrade of the standard revolver, 6 rounds each
* Double barrel shotgun (Pickup), two shot spread fire weapon
* Ammo, each weapon type has its own ammo set .44 and buckshot, .44 comes in 6 round clips and buckshot in 2 shells
* Player characters, move, shoot and receive input, instances hold their own sets of data at runtime.
* Bandage (pickup) restores health that has been lost
* Molotov (pickup) throwable area denial tool, deals damage when a player stands in its fire.
* Train, source of pickups, players must shoot pickups off the back of the train, also a quick way to get yourself killed if you get to close
* Sandstorm; used to shrink map size and increase confrontation to end matches.
  1. **Game Flow**

1. **Mechanics**
   1. **??**
   2. **??**
   3. **??**
   4. **??**
2. **Graphics**
3. **Physics**
4. **Items**
5. **Game Flow**
   1. **Level Structure**
   2. **Objectives**
6. **Levels**
7. **Interface**
   1. **Menu**
   2. **Camera**
   3. **Controls**
8. **External File Formats**
9. **Audio**
10. **Asset List**
11. **Scripts**
12. **Technical Risk**

**Technical Game System Breakdown (TSB):**

Core Gameplay Outline:

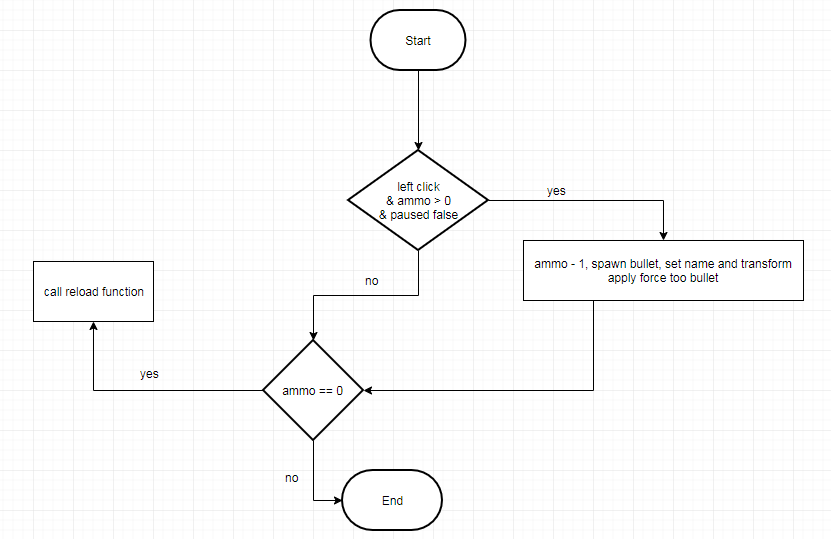
Basic controller movement, to allow for 2 to 4 players, main streamed flow of input for smooth and responsive gameplay.

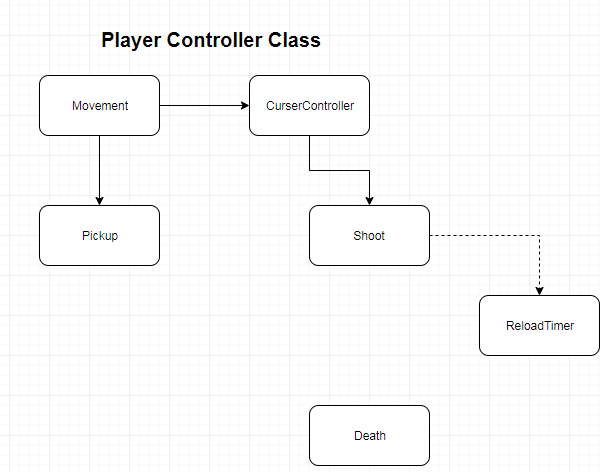
Simple aim and shoot mechanic allows for easy implementation and smooth, responsive gameplay to be created quickly

Simple UI gives feedback to the players on ammo and health

Collectable pickup system so players have variety and rick/reward gameplay

Rebounding bullets create a unpredictable gameplay environment and add increased pace and movement in gameplay

**Shooting**:



Systems outline:

The movement system employs unity’s input manager to give fluent access to console style controllers, where unity’s input axis array has the ability to be added to and allow the multiple controller inputs we need to have the game function smoothly and respond nicely.

We are employing a basic aim and shoot system built in to the character controller, players aim by rotating there character to face and use the right bumper to shoot. This also uses a reloading mechanic to add strategy to ammo preservation