**Technical Design Document**

**Basic Overview:**

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

**Gameplay Pitch**: Top-down isometric arcade western shooter with skill shot style mechanics

**Gameplay Overview:**Set in the Mid 1800’s in the Mid- Southern States in a recluse, desolate place of a town, where the ultimate Mexican standoff between mercenaries, bandits & county rangers like personas is about to begin. Players are given default revolvers & seek to acquire combat collectables which are shot off the back of a train, in hopes of wielding mechanically superior combatant weaponry to overwhelm & further eliminate their foe(s)

**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

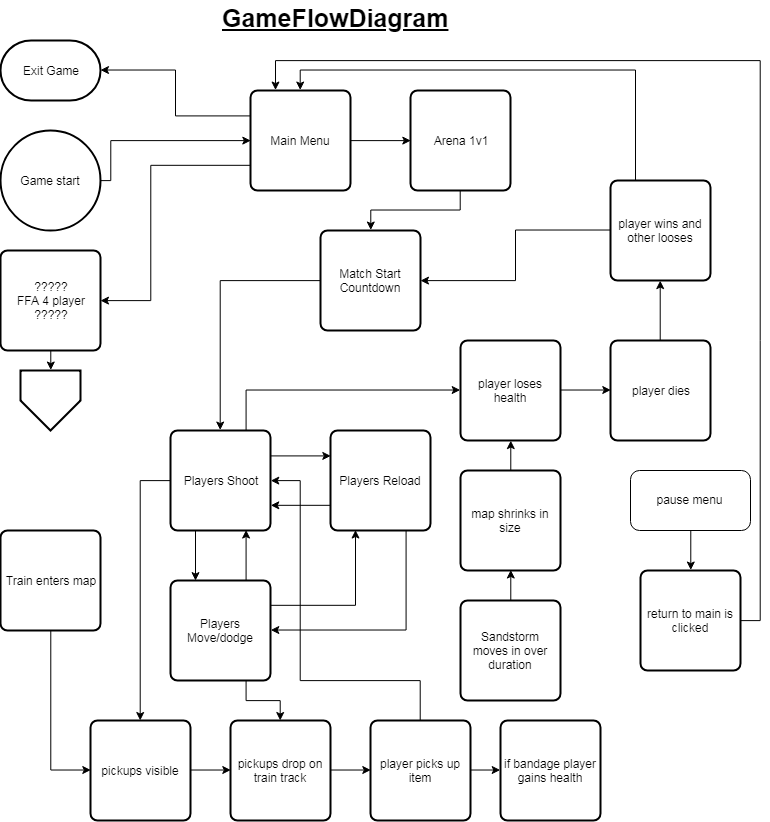
Unity 2017.3.0f3

* 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). e.g. moveSpeed
* Function names should be verbs or a name in relation to what the method does. e.g. Movement or SpawnBullet
* 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**
* Autodesk Maya
* Substance Painter
* ZBrush
* Adobe Photoshop
  1. **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. **Movement**

* The movement of the game is controlled by console controllers, left joystick is used for movement and right used for aiming the curser. Because of the share screen multiplayer, PlayerController.cs needs to made generically with references to the object the instance of the code is attached to and the instances of the related objects needed by the player object to function correctly. This affects how the movement will look in code, with 2 or players we need to create multiple input axis in unity’s input manager and reference them in code specifically if the input is for P1 or P2.
  1. **Shooting**
* The shooting in the game is handled by the player controller with its own method, a check is made to see if the player has pressed the fire button, whether they have ammo and if they a currently reloading. Once this check returns true a bullet object is spawned, force is applied to it forwards of its position and the ammo value decreases. If a player is hit by a bullet they lose health. If the player runs out of ammo they automatically reload or drop the gun if is a pickup weapon.
  1. **Sandstorm**
* Over time the map shrinks in size to force an ending, this is done with a timer and colliders, the timer will tell the colliders to move in after a set amount of time and if the colliders hit a player the player will lose health rapidly.
  1. **Train / Pickups**
* Pickups are used in the game to give the players slight advantageous bonuses, these items will be run through the map on the back of a train which will come through at as set interval and players will need to shoot them off. The pickups will drop in place on the train track the players can run in t collect them once the train leaves, both players can rush in for the dropped loot.

1. **Graphics**

* Refer to Art Bible for images and graphical reference

1. **Physics**

* Ricochet, unity’s in built physics engine, with the possible addition of some new code for fixing potential bugs and issues

1. **Items**

|  |  |  |  |
| --- | --- | --- | --- |
| Item | parameter | parameter | Description |
| Revolver | 1 damage | 1 shots | Default weapon |
| Dual Revolver | 1 damage | 2 shots | Pickup |
| Shotgun | 1 damage | 3 shots | Pickup |
| Bandage | 1 heal | -------- | Pickup |
| Molotov | 5 area size | 1 damage / time | Pickup |

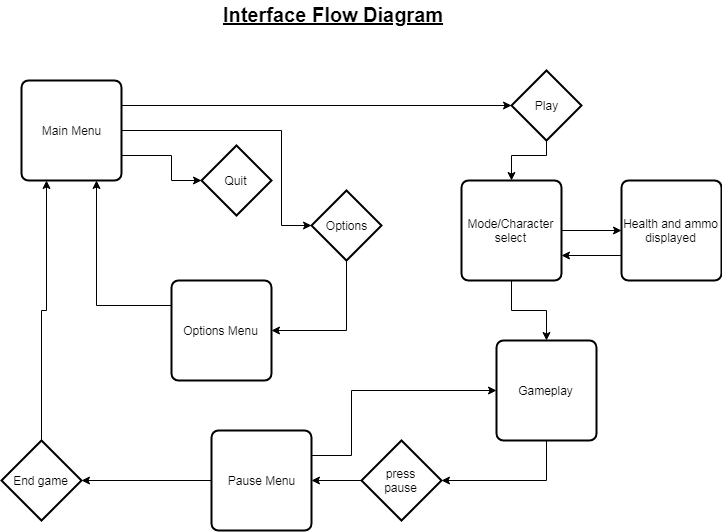
1. **Game Flow**
   1. **Level Structure**

* StateManager.cs will exist over all scenes
  1. **Objectives**
* Shoot to defeat your opponent
* Win and archive bragging rights the ultimate accomplishment

1. **Levels**

* One map iteration for the time being and the possible addition of more later, mostly TBD

1. **Interface**

****

* 1. **Menu**
* 2d interface for main and pause menus
* Mouse will be used to travel in the menus, will do research on being able to do this with a console controller.
  1. **Camera**
* In 1v1 map the camera is locked stationary
  1. **Controls**
* Console controller will be the input device we use
* Left joystick for movement
* Right joystick for aiming
* Right bumper button for shooting
* Left bumper button for throw Molotov

1. **External File Formats**

* TBD

1. **Audio**

* Shooting audio (revolver and shotgun)
* Movement
* Ricochet
* Reload
* Train movement
* Train smoke/steam
* Collectables falling
* Sandstorm tornado
* Wind ambience
* Background audio

1. **Asset List**
2. **Scripts**

* List of currently needed and accounted for .cs scripts:

1. PlayerController.cs
2. Bullet.cs
3. Train.cs
4. Pickup.cs
5. GameManager.cs
6. StateManager.cs
7. Sandstorm.cs
8. **Technical Risk**

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

Core Gameplay Outline:

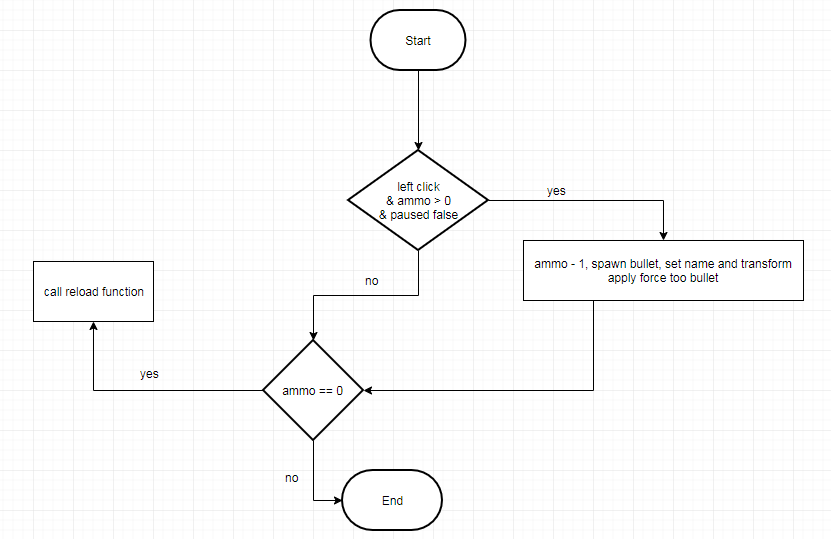
Basic controller movement, to allow for 2 to 4 players, main streamed flow of input for smooth and responsive gameplay.

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**: