**TEAM 4 (INFINITE ENGINE) AGREEMENTS**

**JORDAN JAMES & ESK EVE**

**ACTORS**

Player character

Projectile

Item (Special Energy Pack, Health Pack, Damage Boost Pack)

Portal

Stage

**COMPONENTS**

Transform Component

Render Component

Player Input Component

Collision Component

AI Component

Player Character Component

Pickup Component

Portal Component

Stage Component

Projectile Component

Audio Component

**MAIN LOOP**

* Take input
* Run AI
* Update sprite animations
* Update timers (game timer, timers for spawning items and portals, special move cooldown)
* Update world (position of player/enemy, items, projectiles)
* Render visuals
* Play audio

**CODING CONVENTIONS**

* Classes declared in upper camelcase (e.g. TransformComponent)
* Member functions declared in lower camelcase (e.g. TransformComponent.getPositionVector)
* Member functions declared in lower camelcase (e.g. TransformComponent.mPositionVector)
* Braces on a new line, i.e.:

if (myBool)

{

myBool = false;

}

* Prefixes:
  + m for member variables
  + mp for member pointer variables
  + g for global variables
  + No prefixes for functions
* Comments: double slashes followed by a single space (e.g. // Some comment)

**DIRECTORY STRUCTURE**

* Source
  + Base Classes
    - Component
    - Component System
  + Physics
    - Transform Component
    - Transform Component System
    - Collision Component
    - Collision Component System
  + Graphics & Audio
    - Render Component
    - Render Component System
    - Audio Component
    - Audio Component System
  + Player Characters
    - Player Character Component
    - Player Character Component System
    - Player Control Component
    - Player Control Component System
    - AI Component
    - AI Component System
  + Items
    - Pickup Component
    - Pickup Component System
  + Features
    - Stage Component
    - Stage Component System
    - Portal Component
    - Portal Component System

**ENGINE QUESTIONS**

1. ***What limitations do you think your game engine would have? Try to explain what these limitations are****.*

I think our engine is limited by its lack of tools for online multiplayer games. This includes any forms of communication or gameplay. There are no components to handle server hosting, stability or searching. Furthermore, there are no components to support communication between players even if the engine did support online play in its current state. A component for voice chat or even a chat box would be necessary.

1. ***How can you adapt your game engine in order to accept a different game genre?***

Our engine is built for a third person top down shooter so it has player collision, movement and a few projectile attack components. With a few changes, it could be adapted to a arcade-style fighting game.

First, the player would need components for melee combat. Using Street Fighter as an example, this would include low, mid, high and air attacks. Depending on the game design another component for combination moves.

Secondly, the physics component interface would need more components to handle this new moveset. Player reactions may be less physics based and instead depend on the exact point of contact and type of attack.

Finally, the stage could use make use of the physics interface class for in-depth stage interaction. It would then be possible to create a moving stage with realistic physics for players to battle in. Alternatively, the stage could be static and have a component for breakable terrain or objects increasing overall immersion.