# GAM 465 Proof of Concept

**Tank Player Character**

by Cody Owens

**Summary**

**Overview**

This asset is a player character. It will be a tank view from a top down perspective. The character will be able to move, aim it turret and shoot in 4 directions up, down, left, and right. It will receive and action point every 3 seconds with a maximum amount of 2 action points. It will be able to move or shoot once per action point, turret rotation is a free

action.

**Genre**

This is a top down shooter asset that also leaks into an action puzzle game. With limited resources tied to movement and shooting, the player will need to consider every action and response to the enemy.

**Platform**

Target platform will be windows. The asset will be able to be easily transferred to other platforms in the future. Windows is the development platform so it would be efficient to start with Windows as the target platform.

**Feature Overview**

* 4 Directional Movement and turret position – UP, DOWN, LEFT, RIGHT
* Turret with One Projectile per shot
* Resource based movement system – 2 action points, recover 1 every 3 seconds, move or shoot once per action point
* Turret aiming is free

**Target Rating and Target Market**

* Expected ESRB rating is E10+ because of cartoon violence
* Demographic will be pretty large, the asset visuals will be target towards a younger audience, but the mechanics can be appreciated by all demographics
* Asset can be used as player character or blueprint for AI

**Competitive Analysis**

* Game #1: X-COM – Similar action point system but it is turn based instead of real-time.
* Game #2: Binding of Isaac – Similar shooting and movement system – they use 6 directions not 4
* Game #3: Final Fantasy Tactics – Similar grid based movement system

**Specialization**

* Gameplay Programming
  + Creating this game object in unity with C# scripts allows me to practice prototyping and programming
  + I will be coding all of the functions for player control and interaction with the gameplay

**Framework**

**Plan for Creation**

1. *Create Sprite assets – Tank, Turret, and Bullet (Aseprite)*
2. *Create game object with Tank and turret sprites (Unity)*
3. *Determine movement system (Visual Studio C#)*
4. *Determine aiming system (Visual Studio C#)*
5. *Create shooting functionality (Visual Studio C# and Unity)*
6. *Add action points system (Visual Studio C#)*

**Concept Points**

Character will move around on a grid-based system. It will be able to move one square at a time, in 4 directions: up down left and right. Each move costs one action point. The character starts with 2 action points and will recover 1 action point per 3 seconds. 1 action point will be spent when a player moves or shoots.

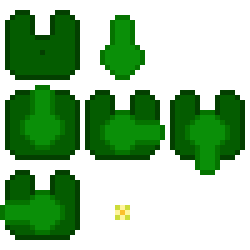
**Other Details**

* Potential base for an AI class for use in game
* I want to use this in a multiplayer game, the real time tactics could be very exciting in a player versus player environment

**Deliverable Format**

* Unity Project
  + Player Controlled Tank Game Object
  + Stationary Targets Game Object Prefab
  + C# Scripts

**Concept Art**



**Development Process**

My development process will begin with creating sprite assets in the pixel art creation program Aseprite. I will then import these sprites into Unity. I will create a game object within Unity that consists of the tank base sprite. Then I will attach another game object to that containing the turret sprite. Finally, a game object will be attached to the turret object and positioned near the barrel. This is where the bullets will fire from. After this a grid will be created inside unity and the entire tank object placed inside it. This will allow the tank to move along the grid in a uniform fashion. Now the tank game object should be set up and ready for coding.

The code will contain an update function that will determine actions based on input from the player. A few base functions will also be included for things like moving in each direction, firing a shot, generating action points, spending action points, and dealing damage to targets. All code will be done in Visual Studio.

**Defense**

I believe my development approach is going to be effective because the workflow is helpful to visualize the final product. Creating the sprite art, and then attaching that to a game object before coding starts is going to help with programming and debugging along the way. I will be able to directly see how each input affects the character on screen.

I am using the listed development tools (Unity, Aseprite, Visual Studio C#) because they are the tools that I am currently most comfortable with. I think this is the most effective approach for quality production of deliverables.