Tanks! Design Doc

Course: COMP 2659, Winter 2016  
Instructor: Paul Pospisil  
  
Author(s): Edison Mendoza, Paul Staples  
Last Modified: Sunday January 17th, 2016

# 1. General Game Overview

Tanks! is a third-person, two dimensional battle-arena game, based around two players attempting to destroy each other by shooting their cannons. Players can move horizontally, and aim their turret anywhere in a 60 degree arc. Upon shooting their cannon, a tank shell will fly out, following an arc along the angle the player shot at. The objective of the game is to land a direct hit on the opposing tank, causing their health to decrease. Once a player’s health reaches zero, the round will end, giving a point to the player left standing.The first player to reach two points (basically best 2 out of 3 rounds), will be declared the match winner. A message will be displayed stating as such, and the game will run back to the menu.



# 2. Gameplay Details for Core 1-Player Version

## Objectives and Rules

When the game starts, after selecting 1-Player, both tanks will be placed at opposite sides of the play area (Arena). Without delay, both players will be ceded control and the match will begin.

The tanks may move horizontally via the ‘A’ and ‘D’ keys, in an attempt to better position themselves and/or dodge the opposing tank shells. Players can adjust their turret via the ‘W’ and ‘S’ keys to move the turret left and right, respectively. A player will score a hit if their tank shell connects with the opposing player’s tank. A win state is reached by being the first player to score 3 points, which are scored by reducing the other player’s health to zero via hits and destroying them.

## 

## Objects

|  |  |  |  |
| --- | --- | --- | --- |
| Object or Object Type Name | Properties | Behaviours | Graphical Image |
| Tank (x 2)  Tank Shell  Score Counter | * Position (Integer pair) * Aim Angle (Integer) * Health (Integer) * Size(Sprite dimensions, integers) * Movement Speed (integer) * Velocity (Integer) * Angle of Travel( Integer) * Location (pair of Integers) * Round count (Pair of Integers) | * Horizontal movement * Aiming * Fire * Changing Velocity * Impact and Fire noises * Will keep track of each players score in the form of 1 - 0, with the left being Player 1 and the right being Player 2 |  |
| Ground  Health Bar (x 2) | * Dimensions(Pair of integers) * Multiple small Gear objects. (Integer) * Dimensions (Pair of integers) | * Defines where the player can move their tank. * eight total gear objects, four for each, getting removed upon a direct hit. |  |
| Player Indicator | * Location (Pair of Integers) | * Denotes which player is which under their health bars |  |
|  |  |  |  |

[Note: for some games, it may not be appropriate or realistic to include images for all object types. However, this is recommended if possible. Simple bitmap-style images (e.g. 8×8, 4×4, 16×16, etc.) are best for many game types. There are free tools to help you construct such images.]

[Note: for some games a particular object may have >1 associated image, depending on its current state (e.g. Mario walking left vs. Mario walking right).]

## Physics

Physics in the game will be limited to the tank shell being fired by either player. The cannon shell will fly at a speed determined by the angle it is fired at, constantly applying gravity (normally -9.81 m/s2 but for ease of use will be rounding to an even 10 m/s2. Once the velocity of the shell reaches 0 from the application of gravity, it will start becoming negative, causing the shell to fall. The physics formulas are as follows: x = VXt for Horizontal Distance, VX = VXo for horizontal velocity, y = VYOt - ½ gt2 for vertical distance and Vy = VYO - gt for vertical velocity, where Vx is velocity along x-axis, Vxo is initial velocity along x-axis, Vy is velocity along Y-axis, Yo is initial velocity along y-axis, g is gravity and t is time taken.

## Asynchronous (Input) Events

|  |  |  |
| --- | --- | --- |
| Event Name | Triggering Input Event | Description |
| Tank Movement  Turret Aiming  Tank Fire | ‘A’ or ‘D’ keys are pressed  ‘W’ or ‘S’ keys are pressed  ‘Space’ is pressed | Tank horizontal position will change, adjusting the X and Y coordinates by the tank’s Speed.  The turret angle will change, adjusting the angle by increments left or right, by ‘W’ and ‘S’ respectively.  The tank will fire a shell at the given angle |
|  |  |  |
|  |  |  |
|  |  |  |

## Synchronous (Timed) Events

|  |  |  |
| --- | --- | --- |
| Event Name | Trigger Timing | Description |
| Tank shell arc  AI shot | After every Fire event  After a regular period of time | A physics calculation will take place every so often, changing the shell’s velocity  The non-human player will move their turret to a random angle and fire, every time the system clock ticks over, with the angle being within a certain range decided by the player’s position. |
|  |  |  |
|  |  |  |
|  |  |  |

## Condition-Based (Cascaded) Events

[Note: some events may trigger other events, conditionally. This is called event “cascading”. The triggering event(s) may themselves be asynchronous, synchronous, or other condition-based events.]

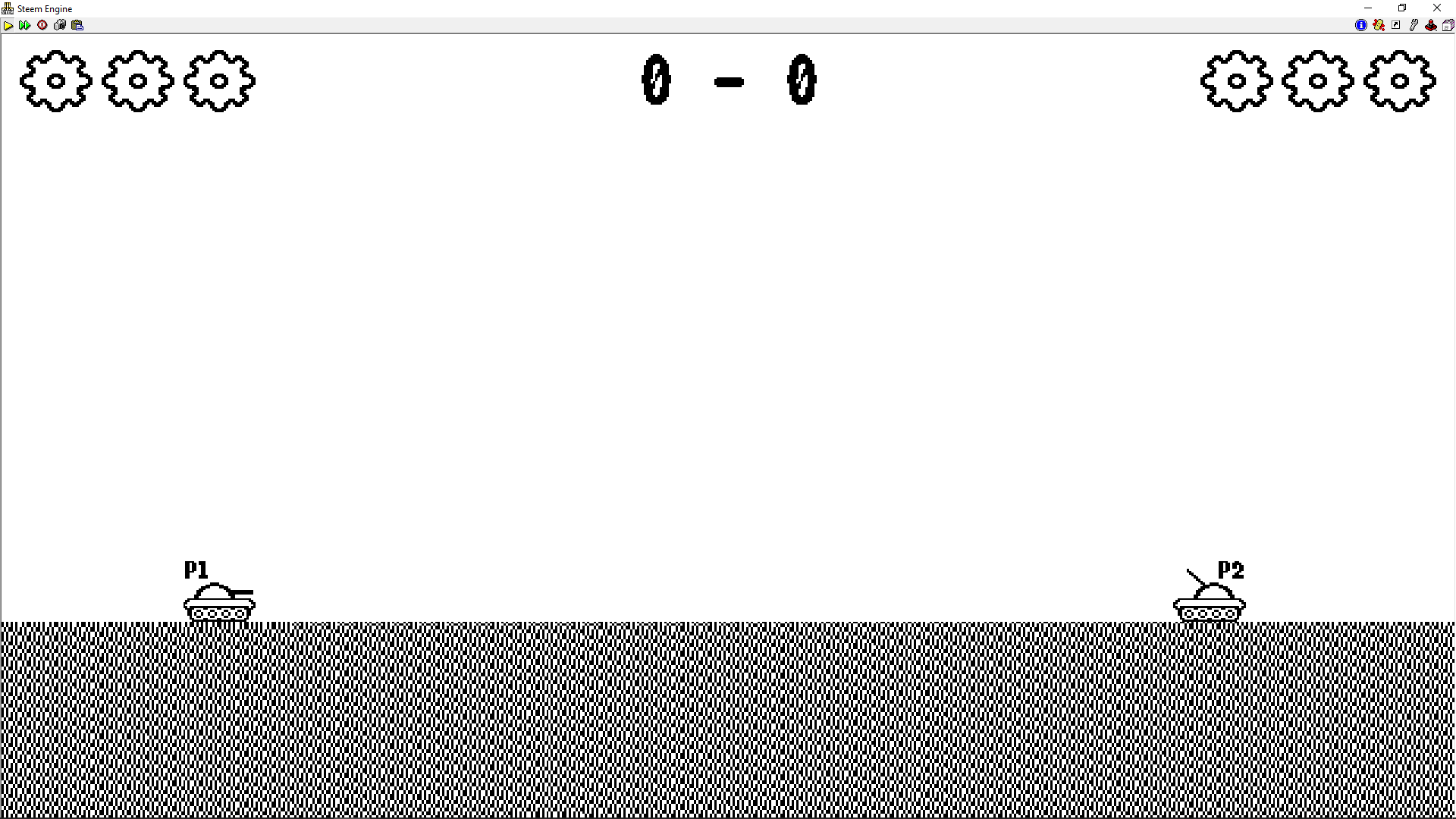
|  |  |  |
| --- | --- | --- |
| Event Name | Triggering Condition | Description |
| Tank damage | A tank is hit by a flying shell. | The damaged tank will be destroyed, playing a sound effect. |
|  |  |  |
|  |  |  |
|  |  |  |

## Hypothetical Gaming Session

A typical session will start with both players (in this case, one human and one AI) will start off at opposite sides of the map, and begin moving, or firing, or both. As shots are fired and positions adjusted, the human player will get a feel for the arc of the shot and the physics, and eventually hit the AI player, who will be firing indiscriminately somewhat in the direction of the Player. After scoring 2 points for a player (enemy tank is destroyed), the game will end with a Player X (1 or 2) Wins! message, and terminate.

# 3. Gameplay Details for Core 2-Player Version

A typical two-player session will play identically to a 1 player session, with the opponent being replaced by a, likely more functional, human opponent. Otherwise, the rules don’t change at all, controls remain the same, and the win condition remains the same.



# 4. Sound Effects

|  |  |  |
| --- | --- | --- |
| Sound Effect Name | Brief Description | Event which Triggers Playback |
| Cannon Fire | A loud pop, using as low of a note as possible | Player hits the Fire button |
| Shell Hit | Best explosion approximation | A tank is hit by a shell. |
|  |  |  |
|  |  |  |

[Optional: brief description of background music. Music will be required by assignment 3.]

# 5. Additional Features (Time Permitting)

If possible:

* randomly generated terrain.
* level select with 4 predetermined levels.
* changing air velocity with direction and intensity affects the trajectory of the shell.

