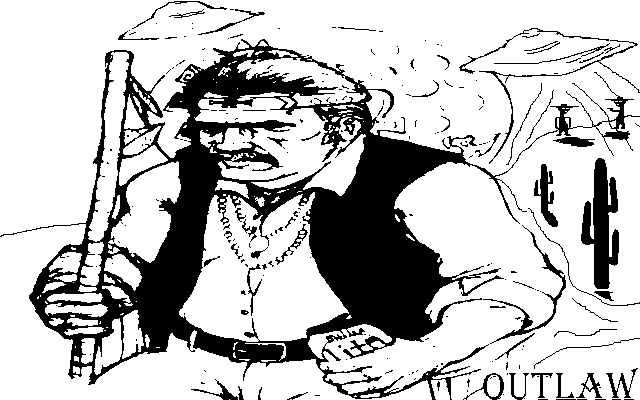
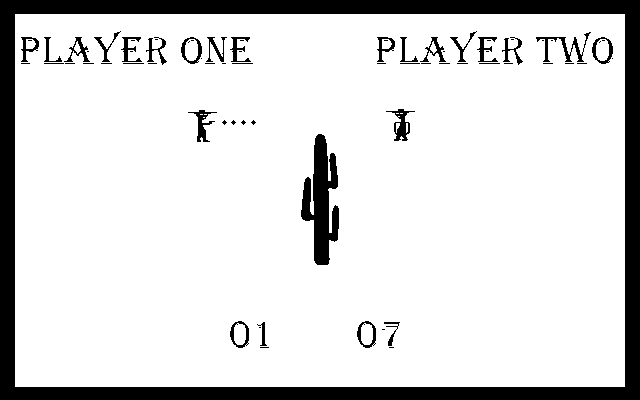
Detailed Game Specification:



Course: COMP 2659, Winter 2020  
Instructor: Dr. Tyson Kendon   
  
Author(s): Michael S, Walker  
Last Modified: 1/27/2020

# 1. General Game Overview

In the outlaw game, you control the movement of a gunslinger. You score points when your gunslinger shoots your opponent's gunslinger.



# 2. Game Play Details for Core 1-Player Version

## Objectives and Rules

In one-player games (Player Vs AI), you control the movement of a gunslinger. You score points when your gunslinger shoots your opponent's gunslinger. The first player to score 10 points wins the game.

## Objects

|  |  |  |  |
| --- | --- | --- | --- |
| Object or Object Type Name | Properties | Behaviours | Graphical Image |
| Bullet  Cactus  Score | * 1 pixel in size * N Pixels in size (to be determined later) * 8x8 pixel font used | * Projectile from gun * A Center line (can’t pass) bullets can’t penetrate * Updated on player hit | (a dot)    0,1,2,3,4,5,6,7,8,9 |
| Player 1 | * player\_1.normal\_stance * player\_1.walk\_stance * player\_1.shoot\_straight * player\_1.shoot\_up * player\_1.shoot\_down * player\_1.dead | * P1 starting sprite * (default sprite) * P1 walking sprite * P1 shoot straight sprite * P1 shoot up sprite * P1 shoot down sprite * P1 dead |  |
| Player 2 | * player\_2.normal\_stance * player\_2. walk\_stance * player\_2.shoot\_straight * player\_2.shoot\_up * player\_2.shoot\_down * player\_2.dead | * P2 starting sprite * (default sprite) * P2 walking sprite * P2 shoot straight sprite * P2 shoot up sprite * P2 shoot down sprite. * P2 dead |  |

## Physics

## Asynchronous (Input) Events

Error checking and logic black boxed in subroutines.

|  |  |  |
| --- | --- | --- |
| Event Name | Triggering Input Event | Description |
| P1/P2 Walk Up | P1 (W key is pressed)  P2 (UP key is pressed) | MovePlayerY(&player, “up”);  /\* player1->y\_pos -= 32; \*/ |
| P1/P2 Walk Down | P1 (S key is pressed)  P2 (DOWN key is pressed) | MovePlayerY(&player, “down”);  /\* player1->y\_pos += 32; \*/ |
| P1/P2 Walk Right | P1 (D key is pressed)  P2 (RIGHT key is pressed) | MovePlayerX(&player, “right”);  /\* player1->x\_pos += 32; \*/ |
| P1/P2 Walk Left  Player Shoot Straight  Player Shoot Up  Player Shoot Down | P1 (A key is pressed)  P2 (LEFT key is pressed)  Left Shift is held and then released with no directional key held.  (make / break)  Left Shift is held and then released with directional key (P1/P2 Walk Up) held.  (make / break)  Left Shift is held and then released with directional key (P1/P2 Walk Down)  held.  (make / break) | MovePlayerX(&player, “left”);  /\* player1->x\_pos -= 32 \*/  ShootBullet(&player, “direction”);  ShootBullet(&player, “direction”);  ShootBullet(&player, “direction”); |

## Synchronous (Timed) Events

|  |  |  |
| --- | --- | --- |
| Event Name | Trigger Timing | Description |
| Bullet Moves | Every 1/10th of a second. | Bullet moves forward *1* pixel. |
| Player Moves | Every 1/10th of a second. | Player position updated 32 pixels (1 unit). |
|  |  |  |
|  |  |  |

## Condition-Based (Cascaded) Events

|  |  |  |
| --- | --- | --- |
| Event Name | Triggering Condition | Description |
| Bullet collides with  Y(MAX) or Y(MIN) | Bullet is in motion and collides with bounding Y coordinate  (y = 0 or y = 400) | Vector changes direction |
| Bullet collides with X(MAX) or X(MIN) | Bullet is in motion and collides with bounding X coordinate  (x = 0 or y = 640) | Bullet object deleted |
| Bullet collides with player  Score Updated | Bullet is in motion and collides with player  Bullet collides with player | Bullet object deleted  Enemy player model == player.dead  Point added to player score  Score is updated by 1 point if score == 10 game ends |
|  |  |  |

## Hypothetical Gaming Session

Player1 and player2 start on opposite side of screen, players try to shoot each other the first player to score 10 points wins.

# 4. Sound Effects

|  |  |  |
| --- | --- | --- |
| Sound Effect Name | Brief Description | Event which Triggers Playback |
| gun\_shot  player\_hit  y\_wall\_hit  x\_wall\_hit | Sounds like BANG!  Sounds like OOF!  Sounds like ZING!  Sounds like POOF! | When player fires weapon  Bullet collides with player  Bullet collides with  Y(MAX) or Y(MIN)  Bullet collides with X(MAX) or X(MIN) |
| player\_move | Sounds like TAP TAP! | Player Moves |
| game\_start  Music | Sounds like DING!  C, F, G in a loop | New game initialized  Game Music |

# 5. Additional Features (Time Permitting)

Menu with mouse enabled, and networked multiplayer version working.