Multimedia Programming Project

Sechs Krieg

Declan Toohey

844969X

Problem Definition

# Scenario

Sechs Krieg will be a 2D twin stick shooter retro style game. Twin stick shooter meaning that it will be controlled with WASD and the arrow keys, in a similar way to the games “Binding of Isaac” and “Geometry Wars”. These are games in which the player moves around using one joystick, while the other joystick is used to shoot at enemies. To progress through these games waves of enemies would have to be taken out.

Sechs Krieg will have the player spawned in the middle of the screen and instantly thrown into the game with enemies coming from all directions. The amount of enemies and difficulty of the enemies will increase with time. There will be a current score and remaining lives counters on the screen. WASD will be used to move the player, while the arrow keys will be used to shoot in the corresponding direction.

The intended audience for Sechs Krieg is children to adults. This is due to its intuitive controls and low violence level. The game could be easily played by anyone for a short period of time for casual players or for longer periods of time for more advanced players. This is beneficial in a world of mobile games that are opened for 5 minutes at a time while waiting between activities. This game could easily be transferred to mobile with the use of onscreen joysticks.

The most obvious benefit to playing Sechs Krieg is that it develops hand eye coordination. To be able to control the ship, while shooting in the correct direction and avoiding incoming enemies. Playing would also benefit the player by having to use strategy. Sechs Krieg will be able to be played in two ways, run and gun, and strategical. Anyone could jump in and just start playing, while overtime, patterns will be noticed to develop strategy.

# Outcomes

1. **Moving Player and Shooting**

The player moves up/down/left/right using WASD. The player will initially get five lives. This is to allow for learning how to play the game. The player will gain more lives as they get different power ups.

The player shoots up/down/left/right using the arrow keys. If the bullet hits an enemy, the enemy is removed; an animation and a sound is played. The score increases by an amount determined by which enemy it was.

1. **Spawning and Moving Enemies**

Enemies will spawn off screen and move towards the player. There will be different types of enemies, ones that will be destroyed in one shot or multiple shots. Enemies will try to get to the players current location as when the enemy touches the player, the player loses a life, the enemy is removed and a damage sound and graphic will play*.* The enemies add a purpose to the game, otherwise, the game would just be a ship flying around shooting at nothing. The enemies force the player to think strategically and develops hand eye coordination.

1. **Picking Enemy and selecting spawn location**

To maintain some randomness to the game, there will be multiple properties about the enemy. The enemy will be picked from a set of ten; however, not all will be available at the beginning of the game. As the score increases, more enemies will be available to spawn. Enemies will have different properties such as movement speed, health and point value.

The Enemies will be randomly spawned around the edges of the stage. This keeps the player focused on all edges and adds some unpredictability to the game to make it more entertaining.

1. **Power Ups and Power Downs**

Power ups will be available to increase game play and allow for more strategical plays. They will spawn randomly on the play field and only be available for a short amount of time. This is to add to the chaos and give the player small objectives when playing, and to get to the power up as fast as possible. Some power ups include, increasing shooting speed, increasing movement speed and damage shields.

However, to increase difficulty, power downs will spawn just as randomly. This makes players have to either avoid these or adapt to the negative effects. This also adds to the chaos of the game. Some possible power downs include, decreasing shooting speed, decreasing movement speed and making the ship larger.

Analysis and Design

# User Interface

nameSet frame: sets the name of the player to be used if the player beats the current high score. The next button sets the name and changes the frame to the play frame.

nameSetTxt : Input Text

Value will set playerName to be used later for high scores

start frame: starting frame of the game. Each button changes the frame. The play button initiates the game, changing the current frame to nameSet. The settings button changes the current frame to settings. The exit button quits the games.

ship : MovieClip

Will rotate for aesthetics

nextBtn : Button

When clicked to start the game

What’s your Name?

Next

playBtn : Button

When clicked to change stage to nameSet

exitBtn : Button

When clicked to exit the game

settingsBtn : Button

When clicked to change stage to settings

Settings

Exit

Play

Sechs Krieg

replayBtn : Button

button to replay game

Replay

ship, enemy1, enemy2, enemy3 : MovieClips

Will rotate for aesthetics

hsNameTxt : DynamicText

Shows the name of the highest score

hsScoreTxt : DynamicText

Shows the score of the highest score

finalScoreTxt : DynamicText

Shows the final score of the game

Game Over!

Score:

High Score:

end frame: end frame is the last frame of the game, it changes once lives hits 0. It displays the final score of the current game as well as the highest score and the person who achieved it.

play frame: main frame for the game, player will move with WASD and shoot with the arrow keys. Enemies and powers will spawn randomly off stage and come into the stage. Enemies will target and follow the player, while the powers will act passive and just pass through the stage. When an enemy is killed the score will increase, when an enemy hits the player, the player loses a life.

player : MovieClip

Controlled by arrow keys and WASD

livesTxt : DynamicText

Shows the current lives of the player

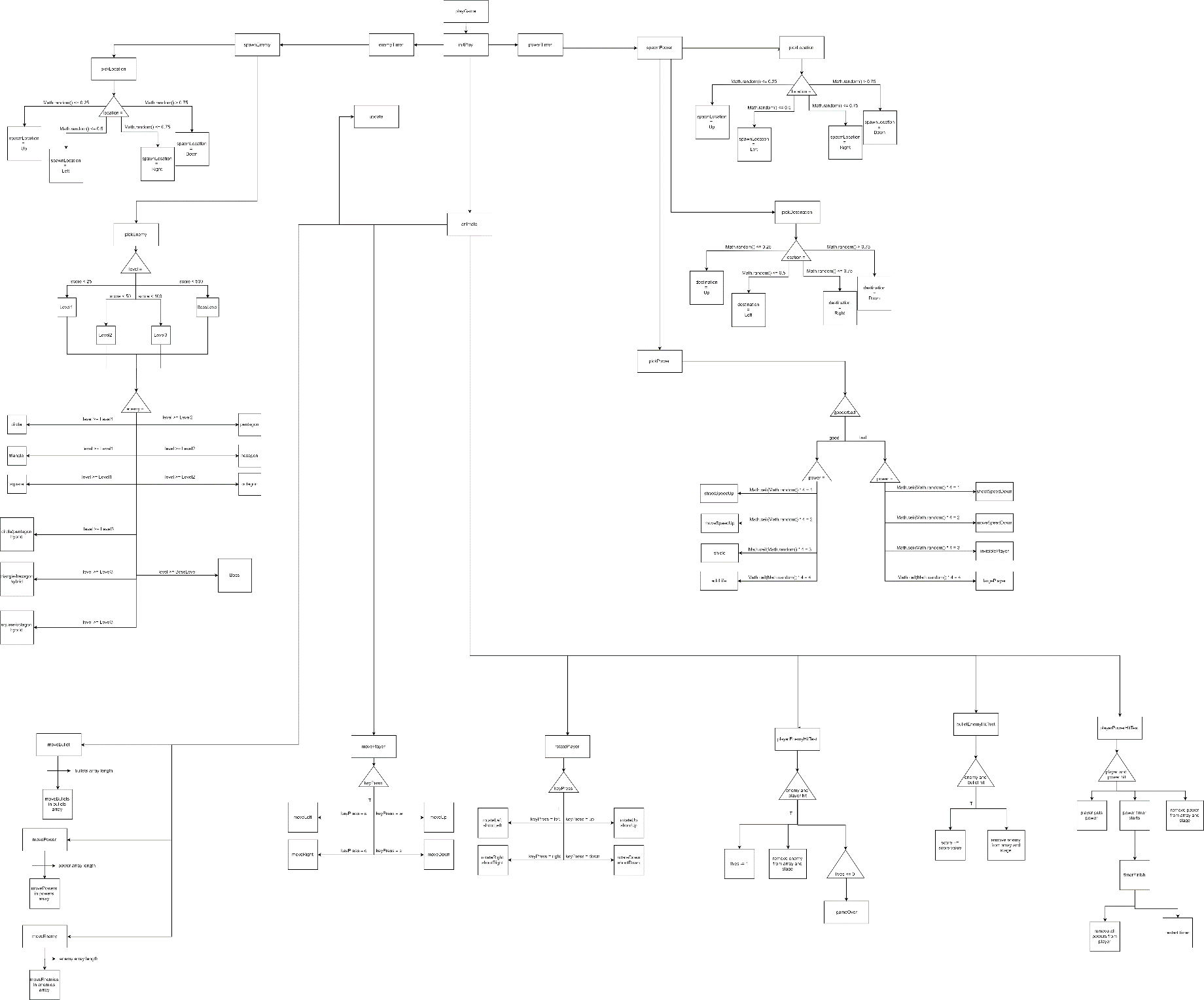
scoreTxt : DynamicText

Shows the current score of the game

Lives:

Score:

# Game Structure



# Outcomes

**Outcome 1: Moving Player and Shooting**

|  |  |  |
| --- | --- | --- |
| Input | Processing | Output |
| WASD keys pressed down | IF w key pressed  wPressed = true  ELSE IF a key pressed  aPressed = true  ELSE IF s key pressed  sPressed = true  ELSE IF d key pressed  dPressed = true  END IF | Player can move according to key press:  W = Up  A = Left  S = Down  D = Right |
| Arrow keys pressed down | IF left arrow key pressed  leftPressed = true  ELSE IF right arrow key pressed  rightPressed = true  ELSE IF up arrow key pressed  upPressed = true  ELSE IF down arrow key pressed  downPressed = true  END IF | Player rotates and shoots bullets in the corresponding direction of the button pressed |
| WASD keys released | IF w key released  wPressed = false  ELSE IF a key released  aPressed = false  ELSE IF s key released  sPressed = false  ELSE IF d key released  dPressed = false  END IF | Player stops moving |
| Arrow keys released | IF left arrow key released  leftPressed = false  ELSE IF right arrow key released  right Pressed = false  ELSE IF up arrow key released  upPressed = false  ELSE IF down arrow key released  downPressed = false  END IF | Player stops shooting |
| Enter-Frame | IF wPressed  player yPosition += ySpeed  ELSE IF aPressed  player xPosition += xSpeed  ELSE IF sPressed  player yPosition += ySpeed  ELSE IF dPressed  player xPosition += xSpeed  END IF | Player moves distance ySpeed or xSpeed |
| Enter-Frame | IF leftPressed  spawn bullet moving left  player rotates left  ELSE IF rightPressed  spawn bullet moving right  player rotates right  ELSE IF upPressed  spawn bullet moving up  player rotates up  ELSE IF downPressed  spawn bullet moving down  player rotates down  END IF | Player shoots bullet in direction according to WASD key pressed |
| Enter-Frame | create bullet  add bullet to stage  add bullet to bullets array  bullet xPosition = player xPosition  bullet yPosition = player yPosition  bullet rotation = player rotation | Bullet spawned at player location and rotation, and is put into bullets array |
| Enter-Frame | FOR each bullet in bullets array  bullet xPosition += bulletSpeed  bullet yPosition += bulletSpeed  IF bullet xPosition < 0 or > stageWidth  or bullet xPosition < 0 or > stageHeight  remove bullet from stage  remove bullet from bullets array  END IF  END FOR | Move bullets from player position at bulletSpeed and if it is outside of the stage boundaries, remove it from the stage and bullets array |
| Enter-Frame | FOR each enemy in enemies array  FOR each bullet in bullets array  IF bullets collide with enemies  current score += enemy point value  remove bullet from stage  remove bullet from bullets array  remove enemy from stage  remove enemy from enemies array  END IF  END FOR  END FOR | When the bullets collide with an enemy, the enemy and bullet are removed from the stage and their respected arrays. The score will increase as well. |

**Outcome 2: Spawning and Moving Enemies**

|  |  |  |
| --- | --- | --- |
| Input | Processing | Output |
| Enemy spawn timer ticks | create enemy  add enemy to stage  add enemy to enemies array | Spawn enemies |
| Enter-Frame | FOR each enemy in enemies array  move enemy towards player with  enemy’s speed property  END FOR | Enemies will move towards the player |
| Enter-Frame | FOR each enemy in enemies array  IF player collides with enemy  reduce current lives by 1  remove enemy from stage  remove enemy from enemies array  IF lives = 0 or lives < 0  game over, change scene to end  screen  END IF  END IF  END FOR | When enemies hit the player, the player loses a life and the enemy is removed |

**Outcome 3: Picking Enemy and selecting spawn location**

|  |  |  |
| --- | --- | --- |
| Input | Processing | Output |
| Enemy spawn timer ticks | Choose random enemy  IF enemy is circle  enemy graphic is circle  enemy health = 1  enemy points = 1  enemy speed = enemy base speed  ELSE IF enemy is triangle  enemy graphic is triangle  enemy health = 1  enemy points = 2  enemy speed = enemy base speed  ELSE IF enemy is square  enemy graphic is square  enemy health = 1  enemy points = 5  enemy speed = enemy base speed  ELSE IF enemy is pentagon  enemy graphic is pentagon  enemy health = 2  enemy points = 5  enemy speed = enemy base speed \* 1.25  ELSE IF enemy is hexagon  enemy graphic is hexagon  enemy health = 2  enemy points = 10  enemy speed = enemy base speed \* 1.25  ELSE IF enemy is octagon  enemy graphic is octagon  enemy health = 2  enemy points = 15  enemy speed = enemy base speed \* 1.25  ELSE IF enemy is circle/pentagon hybrid  enemy graphic is circle/pentagon hybrid  enemy health = 3  enemy points = value between 10 and 20  enemy speed = enemy base speed  ELSE IF enemy is triangle/hexagon hybrid  enemy graphic is triangle/hexagon hybrid  enemy health = 3  enemy points = value between 10 and 25  enemy speed = enemy base speed  ELSE IF enemy is square/octagon hybrid  enemy graphic is square/octagon hybrid  enemy health = 3  enemy points = value between 10 and 30  enemy speed = enemy base speed  ELSE IF enemy is boss  enemy graphic is boss  enemy health = 10  enemy points = 50  enemy speed = enemy base speed \* 0.5  END IF | Enemies spawn with different graphics, health, point values and speed. |
| Enemy spawn timer ticks | Randomly pick location  IF location is up  enemy yPosition = 0  enemy xPosition = number between 0 and stageWidth  ELSE IF location is down  enemy yPosition = stageHeight  enemy xPosition = number between 0 and stageWidth  ELSE IF location is left  enemy yPosition = number between 0 and stageHeight  enemy xPosition = 0  ELSE IF location is right  enemy yPosition = number between 0 and stageHeight  enemy xPosition = stageWidth  END IF | Enemies spawn randomly around the stage. |

**Outcome 4: Power Ups and Power Downs**

|  |  |  |
| --- | --- | --- |
| Input | Processing | Output |
| Powers timer ticks | create power  add power to stage  add power to powers array  choose power  IF power is move speed up  power graphic is move speed up graphic  power’s power = move speed up  ELSE IF power is move speed down  power graphic is move speed down graphic  power’s power = move speed down  ELSE IF power is shoot speed up  power graphic is shoot speed up graphic  power’s power = shoot speed up  ELSE IF power is shoot speed down  power graphic is shoot speed down graphic  power’s power = shoot speed down  ELSE IF power is shield  power graphic is shield graphic  power’s power = shield  ELSE IF power is extra life  power graphic is extra life graphic  power’s power = extra life  ELSE IF power is invisible player  power graphic is invisible player graphic  power’s power = invisible player  ELSE IF power is large player  power graphic is large player graphic  power’s power = larger player  END IF  Randomly pick spawn location  IF location is up  power yPosition = 0  power xPosition = number between 0 and stageWidth  ELSE IF location is down  power yPosition = stageHeight  power xPosition = number between 0 and stageWidth  ELSE IF location is left  power yPosition = number between 0 and stageHeight  power xPosition = 0  ELSE IF location is right  power yPosition = number between 0 and stageHeight  power xPosition = stageWidth  END IF  Randomly pick destination location  IF destination location is up  destination location yPosition = 0  destination location xPosition = number between 0 and stageWidth  ELSE IF destination location is down  destination location yPosition = stageHeight  destination location xPosition = number between 0 and stageWidth  ELSE IF destination location is left  destination location yPosition = number between 0 and stageHeight  destination location xPosition = 0  ELSE IF destination location is right  destination location yPosition = number between 0 and stageHeight  destination location xPosition = stageWidth  END IF | Spawn power ups and/or downs |
| Enter-Frame | FOR each power in powers array  move power from current location to destination location  IF power xPosition < 0 or > stageWidth  or power xPosition < 0 or > stageHeight  remove power from stage  remove power from powers array  END IF  END FOR | Move powers from one side of screen to the other |
| Enter-Frame | FOR each power in powers array  IF power collide with player  IF power is move speed up  xSpeed \* 1.5  ySpeed \* 1.5  ELSE IF power is move speed down  xSpeed \* 0.5  ySpeed \* 0.5  ELSE IF power is shoot speed up  bulletSpeed \* 1.5  ELSE IF power is shoot speed down  bulletSpeed \* 0.5  ELSE IF power is shield  When enemies hit, do not lose life, stay active until power timer ends  ELSE IF power is extra life  current lives + 1  ELSE IF power is invisible player  turn player invisible  ELSE IF power is large player  make player larger  END IF  remove power from stage  remove power from powers array  start power end timer  END IF  END FOR | Player has powers and remove powers timer starts |
| Remove powers timer ticks | Turn off shield  Reset shooting speed to default value  Reset xSpeed and ySpeed  Make the player visible  Return player scale | Player loses powers and timer is stopped |

Validation

**Outcome 1: Moving Player and Shooting**

|  |  |
| --- | --- |
| **Test** | **Expected Result** |
| Start game and press WASD keys | Ship will move in the direction of the key pressed |
| WASD keys released | Ship will stop moving and remain stationary |
| Arrow keys pressed and held | Ship will rotate and shoot in direction of arrow keys, shooting will remain as long as key is pressed |
| Arrow keys released | Ship stops shooting |
| Shoot an enemy | Bullet and enemy will be removed from stage and their respected arrays, and score will increase by enemy point value |

Testing shows that the player moves when WASD is pressed, additionally, when WASD is released, the player stops moving. When the arrow keys are pressed and held the ship keeps shooting, and when released, the shooting stops. Testing also showed that when an enemy was shot, both the bullet and the enemy were removed.

**Outcome 2: Spawning and Moving Enemies**

|  |  |
| --- | --- |
| **Test** | **Expected Result** |
| Move player | The enemies should follow the player around the stage |
| Player is hit by an enemy | Remove a life and remove the enemy from the stage and array |
| Player is hit by an enemy with 1 life remaining | The game should conclude and move onto the end frame |

Testing demonstrates that when the player is moving, the enemies do follow the player. Additionally, when the player is hit by an enemy, the enemy is removed and a life is taken away. Finally, testing showed that when the player was hit with only one life remaining, the game changed frame to the end frame.

**Outcome 3: Picking Enemy and selecting spawn location**

|  |  |
| --- | --- |
| **Test** | **Expected Result** |
| Move player around stage | A range of enemies will spawn from the top, bottom, left or right of the stage. |
| Spawn enemy with a low score | Lower level enemies should spawn, initially only three types |
| Spawn enemy with score of 50 or below | 6 different types of enemies will spawn |
| Spawn enemy with score of 100 or below | 9 different type of enemies will spawn |
| Spawn enemy with score of 500 or below | 10 different types of enemies will spawn, one of which includes a boss |
| Shoot enemy with a health value great than 1 | The bullet should hit, but the enemy continues to try reach the player |

Testing verified that enemies are spawned randomly around the edges of the stage and that the different enemy types are spawned according to current score. When an enemy, which has more than one health point, is shot, the bullet is removed and one health point is removed, the successive bullets do the same until health is zero.

**Outcome 4: Power Ups and Power Downs**

|  |  |
| --- | --- |
| **Test** | **Expected Result** |
| Player collides with power | The player should have an altered playing state, e.g. they have a faster shooting speed or slower movement speed |
| Spawn and move power | Power should randomly spawn and move to a random position across from its current location |
| Enemy collides with power | Nothing |
| Bullet collides with power | Nothing |

Testing further illustrated that when the player collides with powers, the player gains a power, negative or positive and is effected by it for a short amount of time. The powers are spawned at random positions around the edges of the stage and are then moved to another random position around the edge of the stage. When an enemy or bullet collides with the power, nothing happens as expected.

Evaluation

Sechs Krieg has met the requirements set in the problem definition. The final product is a 2d twin stick shooter using the WASD and arrow keys. It has also maintained a simplistic arcade feel to the game’s design.

There is very little time to get used to the controls and the player is put right into the action. Lives and score counters are on screen at all times to let the player know how well they are going.

Additionally, all four of the outcomes have been achieved. The player is able to move and shoot, while the enemies are spawning and moving. Enemies randomly select the type of enemy being spawned as well as the spawn location is randomly selected. There are power ups and power downs, four each. These spawn randomly and are passive to the player.

Further modifications that could be made to Sechs Krieg include a story. There is hint at a story in the help section as it was easier to explain what different aspects of the game were used for through story. Sechs Krieg could also benefit from more enemies and powers, different weapons and different ships. Enemies that shoot at the player could be added as well as new powers that could be accessed by enemies. A range of different kinds of weapons could be added, a heavy laser cannon, does more damage than the default blaster but is slower at firing. The different ship could be that it has more starting lives but is slower, or is faster but has fewer lives and has a lesser damage when shooting.

These additional features could increase the amount of strategy needed to play the game as well as further develop hand eye coordination.