***Joint Project 1***

***Part 1***

Game Description: In my game there are two types of enemies one which is either green or yellow that chases down the player and another which is either blue or red which very slowly moves towards the player shooting at the player. You kill these enemies in certain ways depending on what enemy type it is and what colour the enemy is, if the enemy shoots at you the player must jump on the enemy. If the enemy chases you down you must shoot them. You must also kill the player in the opposite zone depending on the colour for instance if the enemy is red he must be killed in the blue zone, if they spawn in the top right of the map they must be killed in the lower left of the map.

As the game progresses enemies will become faster and more of them will spawn. If you shoot an enemy in the wrong zone you take the hit after five hits the player is killed the player also takes damage if the enemy / enemy bullets collide with the player.

Score increases by one per enemy killed in the correct zone. If you die there will be a play again option. Score and health will be displayed in either corners of the screen. There will be an upgrade that increases the players health by one unless it’s full as well as a sprint upgrade and maybe a double tap upgrade. These upgrades may be random or bought through a shop system. There may be a timer to dictate when an enemy spawns.

Game Genre: Action shooter with rogue like elements similar to The Binding of Isaac which is a randomly generated action RPG shooter with rogue like elements.

Goals: The goal of my game is to survive as long as possible to try get the highest score you can to compare and compete with friends

Player: The game is played by moving with w,a,s,d, shooting with the arrow keys and jumping with spacebar.

Enemies: the chaser enemy will move towards the player with a slight delay to the player speed while the shooter enemies move slowly towards the player while shooting towards them if an enemy hits the player the player takes one damage if a player hits an enemy correctly the enemy dies.

Progression: The game progresses by the enemies becoming faster and more of them spawning.

***Part 2***

**Classes**

Game

Player playerOne = new Player();

Shooter\_Enemy shooterOne = new Shooter\_Enemy();

Chaser\_Enemy chaserOne = new Chaser\_Enemy();

GraphicsDeviceManager graphics;

SpriteBatch spriteBatch;

// makes all of the objects used for my game within the classes

Initialize() // used to give each of the variables initial values

LoadContent() // used to load in everything were going to use for the game like images and sounds

UnloadContent() // used to unload the loaded content if needed

Update() // called 60 times a second until the program closes to update everything in the game from position to health and score etc

Draw() // called 60 times a second until the program closes but is only used to draw nothing else

Collision detection() // used to check if anything has collided like a bullet with an enemy or an enemy with the player

Gameover() // used to check when the game is over and display an appropriate message to the player if it is

Reset() // used to reinitialise and reset the game if the player loses and wishes to start again

Player

bool alive = true; // used to know whether to draw the sprite or not

Vector2 position = new Vector2(); // sprite position

Texture2D playerTexture;

Texture2D playerTextureUp;

Texture2D playerTextureDown;

Texture2D playerTextureLeft;

Texture2D playerTextureRight;

Texture2D playerKilled; // different textures depending on which way the sprite is facing

int health = 5; // used to see how many more times the player can be hit

int speed = 20; // player speed

int direction = 0;

int score = 0;

Shoot() // method to shoot the bullet when the player wants it shot

Move() // method to move the sprite about the screen

Score() //when to increment and decrement score

Die() // used to know when to end the game

Health() // used to know when to take health away and when a health upgrade is gained it increments health

BoundryChecking() // used to check if the player has collided with something

Chaser Enemy

bool alive = true; // used to know whether to draw the sprite or not

Vector2 chaserPosition = new Vector2(); // sprite position

Texture2D chaserTexture;

Texture2D chaserTextureUp;

Texture2D chaserTextureDown;

Texture2D chaserTextureLeft;

Texture2D chaserTextureRight;

Texture2D chaserKilled; // different textures depending on which way the sprite is facing

int speed = 15; // enemy speed

Move() // method to move the sprite about the screen

Die() // used to know when to respawn the enemy

BoundryChecking() // used to check if the enemy has collided with something

Shooter Enemy

bool alive = true; // used to know whether to draw the sprite or not

Vector2 shooterPosition = new Vector2(); // sprite position

Texture2D shooterTexture;

Texture2D shooterTextureUp;

Texture2D shooterTextureDown;

Texture2D shooterTextureLeft;

Texture2D shooterTextureRight;

Texture2D shooterKilled; // different textures depending on which way the sprite is facing

int speed = 5; // enemy speed

Move() // method to move the sprite about the screen

Die() // used to know when to respawn the enemy

Shoot() // method to shoot a bullet at the player

BoundryChecking() // used to check if the enemy has collided with something

Bullet

bool alive = true; // used to know whether to draw the sprite or not

Vector2 shooterPosition = new Vector2(); // sprite position

Texture2D shooterTexture;

Texture2D bulletTextureUp;

Texture2D bulletTextureDown;

Texture2D bulletTextureLeft;

Texture2D bulletTextureRight;

// different textures depending on which way the sprite is facing

int speed = 5; // bullet speed

BoundryChecking() // used to check if the bullet has collided with something

***Part 3***

**Research**

A part of my project that I don’t know how to do exactly is the boundary checking for the different enemies in different parts of the screen because for my game a certain type of enemy can only be killed a certain way in a certain area on the map.

The solution was to simply zone off different parts of the screen as certain areas and check what colour the enemy was and if both were okay then you would kill them if not you would take damage

**Pseudo-Code**

Vector2 yellowPosition;

Vector greenPosition;

Vector bluePosition;

Vector redPosition;

Rectangle enemyYellow;

Rectangle enemyBlue;

Rectangle enemyGreen;

Rectangle enemyRed;

hit = false;

int health = 3;

If (bullet.Intersects(enemyYellow) AND ((yellowPosition.x < 250 AND yellowPosition.y < 200) OR (Jump.Intersects(enemyBlue) AND (bluePosition.x < 250 AND bluePosition.y < 200))) //makes sure that it will only count as a hit if it’s the blue or yellow enemy

{

hit = true;

}

Else if (bullet.Intersects(enemyRed) OR bullet.Intersects(enemyBlue) OR bullet.Intersects(enemyGreen) OR jump.Intersects(enemyRed) jump.Intersects(enemyYellow) OR jump.Intersects(enemyGreen))

{

hit =false;

health--;

}

Else

{

hit = false;

}

// this makes it so that if the enemy is within a certain area of the map and is of a certain type then when the enemy is attacked he will die. If the player hits an enemy he’s not supposed to it will cause him damage. otherwise it will just be a miss