[Five big quacking zephyrs jolt my wax bed](http://www.1001fonts.com/nakki-ldr-font.html)

Project by: Sebastian Kruzel

15/12/2015

**Sentinel**

**Game Description:**

Sentinel is a variety of a tower defence game. The player takes control of Sentinel the defender of the Power crystals. An alien race called Krug is trying to mine the Power Crystals and take them away for their own use. Since the crystals give power to the whole planet, everything on it would die if the crystals are gone. It is your job as the Sentinel to defend the crystals from the horde of the Krug.

The game begins by placing the player at the bottom of the screen, a Krug soldier and a Krug gatherer spawn at the top of the screen. The player has a laser cannon that they can fire to kill the Krug. Each enemy awards 50 points (some enemies drop power ups). If a Krug soldier hits the Sentinel, 10% of HP is lost. (This regenerates slowly over time).

**Game Genre:**

This game is a cross between a tower defence game such as Kingdom Rush and a shoot ‘em up like Hydorah. It focuses on strategic defending of the crystals (such as defending a base in a tower defence game) while killing enemies by firing at them (such as in shoot ‘em ups).

**Goals:**

The goal of the game is to defend the crystals from being taken by the Krugs. Once the player kills all the Krugs they win. If all the crystals are taken or if all HP is lost the player loses.

**Player:**

The player moves using WSAD keys, they fire using the spacebar.

**Enemies:**

*Krug Soldier* – The soldiers move at half the speed of the player initially, they follow the player in order to deal damage. The Krug Soldier deals 10% DMG to the player. One laser hit kills a Krug at the beginning

*Krug Gatherer* – The gatherer moves at half speed of the player at the beginning of the game, they follow a path straight down to the crystal and then begin gathering, when they are done they head back up to the top of the screen. The gatherer deals no damage and one laser hit kills them. If the player manages to kill the gatherer before he gets away, the crystal is then returned otherwise the crystal is permanently removed.

**Progression:**

As the game goes on the difficulty increases keeping the player engaged. The Krug Soldiers speed increases and the gathering speed of the gatherers increases, the health of both enemies is also increased. The power ups that drop from the enemies cause the Sentinel to become more powerful as the game progresses which also keeps the player interested.

**Class Diagrams:**

Game1.cs

{

//instance variables

GraphicsDeviceManager graphics; //handles the graphics device

SpriteBatch spriteBatch; //Enables groups of sprites to be drawn

Sentinel sentinel; //The player object

KrugSoldier soldier; //The enemy soldier object

KrugGatherer gathererOne; //the gatherer object

//methods

Initialize(); //initialises all variables

LoadContent(); //loads the game sprites

RestartGame(); //restart the game once its lost

}

Sentinel.cs

{

//instance variables

Texture2D spriteTexture; //the default image of the sprite

Texture2D textureUp; // the up image of the sprite

Texture2D textureDown; // the down image of the sprite

Texture2D textureLeft; // the left image of the sprite

Texture2D textureRight; // the right image of the sprite

const int Start\_Position\_X = 300; //start x position

const int Start\_Position\_Y = 500; //start y position

public bool sentinelAlive = true; //keep track of whether the player

is alive

int Move\_Up = -4; //for moving up

int Move\_Down = 4; //for moving down

int Move\_Left = -4; //for moving left

int Move\_Right = 4; //for moving right

int sentinelWidth = 0; //for collision detection

int sentinelHeight = 0; //for collision detection

int health = 100; //set health to be 100

int score = 0; //set score to be 0

//methods

MoveUp(); //movement of the sentinel

MoveDown();

MoveLeft();

MoveRight();

Shoot(); //shoot the laser

BoundaryCheck(); //don’t let the player off the screen

Hurt(); //if player gets hurt take away HP

Die(); //if the player is dead game over

Update(); //updates the game variables

Draw(); //draws the sprites on the screen

}

KrugSoldier.cs

{

//instance variables

Texture2D spriteTexture; //the default image of the sprite

Texture2D textureUp; // the up image of the sprite

Texture2D textureDown; // the down image of the sprite

Texture2D textureLeft; // the left image of the sprite

Texture2D textureRight; // the right image of the sprite

int Start\_Position\_X =0; //start x position

int Start\_Position\_Y = 0; //start y position

public bool krugAlive = true; //keep track of whether the Krug is alive

int soldierWidth = 0; //for collision detection

int soldierHeight = 0; //for collision detection

int health = 10; //health of the soldier

//methods

FollowPlayer(); //follow the player

Bounce(); //when player is hit the enemy should be bounced back

Spawn(); //the soldier has to spawn after death

Hurt(); //take away HP

Die(); //the soldier dies

Update(); //updates the game variables

Draw(); //draws the sprites on the screen

}

KrugGatherer.cs

{

//instance variables

Texture2D spriteTexture; //the default image of the sprite

Texture2D textureUp; // the up image of the sprite

Texture2D textureDown; // the down image of the sprite

Texture2D textureLeft; // the left image of the sprite

Texture2D textureRight; // the right image of the sprite

public bool gathererAlive = true; //keep track of whether the gatherer is alive

int Start\_Position\_X =0; //start x position

int Start\_Position\_Y = 0; //start y position

int gathererWidth = 0; //for collision detection

int gathererHeight = 0; //for collision detection

int health = 10; //health of the soldier

//methods

Spawn(); //spawn the gatherer

MoveDown(); //move toward the crystal

Gather(); //gather the crystal

MoveUp(); //escape with the crystal

Escape(); //disappear with the crystal

Hurt(); //if hit take away HP

Die(); // when HP is 0, die and return crystal

Update(); //updates the game variables

Draw(); //draws the sprites on the screen

}

Bullet.cs

{

//instance variables

Texture2D spriteTexture; //the default image of the sprite

Texture2D textureUp; // the up/down image of the sprite

Texture2D textureLeft; // the left/right image of the sprite

Int positionX = 0; //X position of the bullet

Int position = 0; //Y position of the bullet

Bool fired = false; //bullet initially not firing until space is pressed

//methods

Spawn(); //spawn when player hits space

Despawn(); //Despawn if either goes off screen or collides with enemy

}

**Researching:**

Researching how to make the game restart after the player loses all lives meaning that the game is lost.

I found out that I should create an instance variable of type bool “gameOver” and a method within the game class called “RestartGame” which will reinitialise all the variables and objects. A simple version to this solution would look like this:

//variables

gameOver = false;

if(sentinelAlive == false) //checking if

{

RestartGame();

}

//restart game method

RestartGame()

{

sentinel = new Sentinel();

gatherer = new KrugGatherer();

soldier = new KrugSoldier();

bullet = new Bullet();

}