Part 2 of Joint Project – Classes

**Part II**

Class Player

{

Vector2 position = new vector2 (x, y); // the position of the player sprite

Int score = 0; // this will increase when the player will kill an enemy

Int Ammo = 400; //determines how much ammo you have

Int health = 100; // shows the amount of health you have

Int lives = 3; // you have 3 lives in a game

Int damage = 40; // this is the damage that the bullet causes

Int speed = 3; //speed by which the enemy moves

Int direction //determines which way the player is moving

Const int North = 1; //number 1 represents north

Const int East = 2; //number 2 represents east

Const int South = 3; //number 3 represents south

Const int West = 4; //number 4 represents west

Texture2D bulletR; //image for the bullet facing right

Texture2D bulletL; //image for the bullet facing left

Texture2D bulletU; //image for the bullet facing up

Texture2D bulletD; //image for the bullet facing down

Texture2D startingPlayer // image of the player when game loads

Texture2D imageRight; // image when the player is facing right

Texture2D imageLeft; // image when the player is facing left

Texture2D imageUp; // image when the player is facing up

Texture2D imageDown; // image when the player is facing down

Move() // this method will allow you to move

Fire() // allow you to fire a gun

BoundaryChecking()// makes sure you are on the screen

AddScore() // adds score If you kill an enemy

DecAmmo() // decreases ammo when shot is fired

DecHealth() decreases health of the player when hit by enemy

DecLives() //when health reaches 0 it will take away a life

Draw() //to draw the sprite

Update()// here everything will update. The position, speed, damage, lives, health etc.

LoadContent() to load all the sprite, noise etc.

MoveDown() //method for the player to move down

MoveUp()//method for the player to move Up

MoveLeft()//method for the player to move Left

MoveRight()//method for the player to move Right

FireDown() //method which moves the y coordinate down the screen

FireUp() //method which moves the y coordinate up the screen

FireLeft() //method which moves the x coordinate left of the screen

FireRight() //method which moves the x coordinate right of the screen

}

Class TankyEnemy

{

Vector2 position = new vector2 (x, y); // the position of the enemy sprite

Int Direction = 1; // this will determine in which direction is the enemy moving

Health = 100; // determines how much health

Damage = 30; // determines how much damage it deals

Speed = 1; //how fast the enemy moves

Bool alive = true; //states whether the enemy is alive or not

Const int North = 1; //number 1 represents north

Const int East = 2; //number 2 represents east

Const int South = 3; //number 3 represents south

Const int West = 4; //number 4 represents west

Texture2D enemy // image of the enemy when game loads

Texture2D imageUp; // image when the enemy is facing up

Texture2D imageDown; // image when the enemy is facing down

Texture2D imageRight; // image when the enemy is facing right

Texture2D imageLeft; // image when the enemy is facing left

Follow() // this method will allow the enemy to follow the player

BoundaryChecking() // will not allow the enemy to move off the screen

DecHealth() //decrease health when is shot by player

Respawn() // new enemy will have a new starting position when previous enemy is killed

Dead() //determines that the enemy is dead

Draw() //to draw the sprite

Update()// here everything will update. The position, speed, damage etc.

LoadContent() to load all the sprite, noise etc.

}

Class FastEnemy

{

Vector2 position = new vector2 (x, y); // the position of the enemy sprite

Health = 60; // determines how much health

Direction = 1; // this will determine in which direction is the enemy moving

Speed = 10; //the speed of the enemy

Bool alive = true; while the enemy is alive it will draw him on the map

Const int North = 1; //number 1 represents north

Const int East = 2; //number 2 represents east

Const int South = 3; //number 3 represents south

Const int West = 4; //number 4 represents west

Texture2D enemy // image of the enemy when game loads

Texture2D imageUp; // image when the enemy is facing up

Texture2D imageDown; // image when the enemy is facing down

Texture2D imageRight; // image when the enemy is facing right

Texture2D imageLeft; // image when the enemy is facing left

Movement()// how the enemy moves

Dead() //determines that the enemy is dead

DecHealth // decreases health when is hit

Respawn() // it will respawn when other enemies are killed

BoundaryChecking()// makes sure that the enemy will not move off the screen when moving

Draw() //to draw the sprite

Update()// here every thing will update. The position, speed, damage etc.

LoadContent() to load all the sprite, noise etc.

}

Class TheZombie

{

Speed = 1; //speed of the zombie

Vector2 position = new vector2(x, y); position of the zombie

Direction = 1; //determines in what direction the zombie will move

Bool alive = false; //determines if he’s alive or not (he is 1 shot one kill, no need for health)

Damage = 10; //the damage is over time

Random RndGen = new Random(); //used to give a random position for the grave to spawn at a map.

Texture2D ZombieGrave; // represents the dead zombie

Texture2D ZombieL;//when zombie moves to the left

Texture2D ZombieR; when zombie moves to the rigth

Texture2D ZombieU; when zombie moves to the up the screen

Texture2D ZombieD; when zombie moves to the down

Const int North = 1; //number 1 represents north

Const int East = 2; //number 2 represents east

Const int South = 3; //number 3 represents south

Const int West = 4; //number 4 represents west

Follow() //the movement of the zombie

Dead() // when dead it looks of the player to be in the radius to become alive

Alive() //when alive it will follow the player

BoundaryChecking() //for the zombie not to go out of the screen

DealDamage() //takes away players health over time

Draw() //to draw the sprite

Update()// here every thing will update. The position, speed, damage etc.

LoadContent() to load all the sprite, noise etc.

}

Class HealthPotion

{

Vector2 position = new vector2(x, y); //the position of the potion

healthPoints = 60; //determines how much health it regenerates

Bool show = true; // it will determine

Texture2D potion; //it will hold the image of the potion.

AddHealth() //the method adds health to player

Draw() //to draw the sprite

Update()// it will update whether to show the Health potion or not.

LoadContent() to load the image of the potion.

}

Class SpeedBoost

{

Vector2 position = new vector2(x, y); //the position of the speed boost

SpeedInc = 10 ; //Adds speed to your player

Bool show = true; // it will determine if the speed boost is drawn or not

Texture2D potion; //it will hold the image of the potion.

AddSpeed() //adds the speed to the player

Draw() //to draw the sprite

Update()// it will update whether to show the speed boost or not.

LoadContent() to load the image of the speed boost.

}

Class ExtraLife

{

Vector2 position = new vector2(x, y); //the position of the power up

Life = 1; //it adds 1 life to your character

Bool show = true; // it will determine to show the power up or not

Texture2D Life; //image for the Extra life

Add()//adds the life to the character

Draw() //to draw the image

Update()// it will update whether to show the Extra life or not.

LoadContent() to load the image of the life power up.

}

Class Door

{

Vector2 position = new vector2(x, y) //the position of the door

Texture2D door; //the image of the door

Draw()//draw the image

Update() //where I will change the position of the door each map

LoadContent()//where the image will be uploaded

}

Class DamageIncrease

{

Vector2 position = new vector2(x, y); //the position of the potion

MoreDamage = 40; //show by how much it will increase your damage

Bool show = true; // it will determine to show the image or not

Texture2D potion; //it will hold the image of the potion.

AddDamage()// the damage of the potion will add to your damage

Draw() //to draw the sprite

Update()// it will update whether to show the Damage potion or not.

LoadContent() to load the image of the potion.

}

Class Invincibility

{

Vector2 position = new vector2(x, y); //the position of the power up

Cons tint NoDamage = 0; //determines how much health it regenerates

Bool show = true; // it will determine to show the power up or not

Texture2D potion; //it will hold the image of the power up

ReduceDamage()//this method will assign “NoDamage” variable o enemy damage so it will look like the player is invincible

Draw() //to draw the sprite

Update()// it will update whether to show the power up or not.

LoadContent() to load the image of the power up.

}

Class Map

{

Rectangle position;

Each map will also have its own background song.

Song songMap1;//background song for map one

… and so on. Different song for each map.

Texture2D StartingMap() //draw method will only draw this image

Texture2D map1; //image for map one

Texture2D map2; //image for map two

Texture2D map3; //image for map three

… // I will add maps depending on how many levels I will have (it’s not decided yet)

Initialize()// here I will initialize that the position of the map which will be the size of the screen

LoadContent() // load all the maps and songs

Draw() //draw the background

Update()// this is where I will change maps and songs

}

Class Game

{

Player player; //create an object of type player

TankyEnemy enemy1; //create an object of type TankyEnemy

FastEnemy enemy2; //create an object of type FastEnemy

TheZombie enemy3; //create an object of type TheZombie

HealthPotion hpotion; //create an object of type HealthPotion

SpeedBoost sBoost; //create an object of type SpeedBoost

Map mapImage; //create an object of type Map

Invincibility invincibility; //create an object of type Invincibility

Door door; //create an object of type Door

DamageIncrease dmgInc; //create an object of type DamageIncrease

ExtraLife exLife; //create an object of type ExtraLife

LoadContent() // load the starting images

Draw() //where I will draw the starting images of the game

Update()// where all the updates of all object will be

Initialize() //where I will initialize all the objects

}

**Part III**

In my project I am not sure and I find complex the shooting of a bullet. I don’t know how to make my character to shoot a bullet from the middle of his body and in the same way he is facing.

For example if the character is facing right and I press the fire button, a bullet should start from his body towards the right side.

I will implement the following pseudocode in my player class.

***Pseudocode of my research***

***I hope its ok to just involve a movement to one side as it would be a lot of writing to do all 4 sides***

Class Player

I need to add new variables to the class player which are as follows

//varaibles

origin;// this variable is here to make sure that the origin is always on the same side which the image is facing

bulletPos;//position of the bullet

bulletAlive = false;//wether the bullet will be drawn(I use this in draw method)

MoveLeft

{

if(direction == West)

{

posotion.X --;

}

SartingTexture = imageLeft;

origin.X = mPosition.X - bulletTexture.Width;

origin.Y = mPosition.Y;

bulletPos.X = origin.X;

bulletPos.Y = origin.Y;

}

In the update method I should have as follows

Update

{

If(left arrow key is pressed)

{

Direction = west;

MoveLeft();

}

If(fire button is pressed)

{

Ammo--; //decrease the amount of ammo when a fire is shot

bulletAlive = true;

}

}

Draw

{

If(bullet == true)

{

Draw bulletTexture;

}

}

In the game class I will also do a collision with the bullet and the enemy. It will look like this

Class Game

{

Update

{

If (the texture of bullet collides with enemy)

{

Enemy.DecHealth();

}

}

}

***Reference***

<https://pixabay.com/pl/shuriken-star-rzucanie-gwiazda-ninja-153172/> - shuriken

<http://gamedev.stackexchange.com/questions/32203/interval-timing-in-xna> - managing time