Game code

class Boss\_Movement : ZilchComponent

{

var Time : Real = 0;

function Initialize(init : CogInitializer)

{

Zero.Connect(this.Space, Events.LogicUpdate, this.OnLogicUpdate);

}

function OnLogicUpdate(event : UpdateEvent)

{

this.Time +=1;

if(this.Time == 1)

{

var random = new Random();

var chance = random.DieRoll(100);

if (chance >75)

{

var temp = this.Owner.Transform.Translation;

temp.X +=-1;

this.Owner.Transform.Translation =temp;

}

else if (chance <25)

{

var temp2 = this.Owner.Transform.Translation;

temp2.X +=1;

this.Owner.Transform.Translation =temp2;

}

this.Time = 0;

}

}

}

class Boss\_kill : ZilchComponent

{

var Health : Real = 5.0;

function Initialize(init : CogInitializer)

{

//Zero.Connect(this.Space, Events.LogicUpdate, this.OnLogicUpdate);

Zero.Connect(this.Owner, Events.CollisionStarted, this.OnCollisionStarted);

}

function OnCollisionStarted(event : CollisionEvent)

{

var otherObject = event.OtherObject;

if (this.Owner.Name == "Boss")

{

if (otherObject.Name == "Bullet")

{

this.Health -=1;

if (this.Health ==0.0)

{

this.Owner.Destroy();

}

}

}

}

}

class Button\_Spawner : ZilchComponent

{

function Initialize(init : CogInitializer)

{

Zero.Connect(this.Space, Events.LogicUpdate, this.OnLogicUpdate);

}

function OnLogicUpdate(event : UpdateEvent)

{

if(Zero.Keyboard.KeyIsDown(Keys.A))

{

var temp = this.Owner.Transform.Translation;

temp.X +=-1;

this.Owner.Transform.Translation =temp;

}

if(Zero.Keyboard.KeyIsDown(Keys.D))

{

var temp2 = this.Owner.Transform.Translation;

temp2.X +=1;

this.Owner.Transform.Translation =temp2;

}

if(Zero.Keyboard.KeyIsPressed(Keys.Space))

{

var spawnPosition = this.Owner.Transform.Translation;

var archetypeToSpawn = Archetype.Find("Bullet");

var createdBullet = this.Space.CreateAtPosition(archetypeToSpawn, spawnPosition);

createdBullet.RigidBody.Velocity = Real3(0,100,0);

}

}

}

class Button\_Spawner\_2 : ZilchComponent

{

var Direction : Real3 = Real3(0,0,0);

function Initialize(init : CogInitializer)

{

Zero.Connect(Zero.Keyboard, Events.KeyDown,this.OnKeyDown);

Zero.Connect(this.Space, Events.LogicUpdate, this.OnLogicUpdate);

}

function OnLogicUpdate(event : UpdateEvent)

{

if(Zero.Keyboard.KeyIsDown(Keys.A))

{

var temp = this.Owner.Transform.Translation;

temp.X +=-1;

this.Owner.Transform.Translation =temp;

}

if(Zero.Keyboard.KeyIsDown(Keys.D))

{

var temp2 = this.Owner.Transform.Translation;

temp2.X +=1;

this.Owner.Transform.Translation =temp2;

}

}

function OnKeyDown(event : KeyboardEvent)

{

if(event.Key == Keys.Up)

{

var spawnPosition = this.Owner.Transform.Translation;

var archetypeToSpawn = Archetype.Find("Bullet\_2");

var createdBullet = this.Space.CreateAtPosition(archetypeToSpawn, spawnPosition);

createdBullet.RigidBody.Velocity = Real3(0,75,0);

}

}

}

class Button\_Spawner\_3 : ZilchComponent

{

function Initialize(init : CogInitializer)

{

Zero.Connect(this.Space, Events.LogicUpdate, this.OnLogicUpdate);

}

function OnLogicUpdate(event : UpdateEvent)

{

if(Zero.Keyboard.KeyIsPressed(Keys.Space))

{

var spawnPosition = this.Owner.Transform.Translation;

var archetypeToSpawn = Archetype.Find("Bullet\_3");

var createdBullet = this.Space.CreateAtPosition(archetypeToSpawn, spawnPosition);

}

}

}

class Button\_spawner\_4 : ZilchComponent

{

function Initialize(init : CogInitializer)

{

Zero.Connect(this.Space, Events.LogicUpdate, this.OnLogicUpdate);

}

function OnLogicUpdate(event : UpdateEvent)

{

if(Zero.Keyboard.KeyIsPressed(Keys.Up))

{

var random = new Random();

var chance = random.DieRoll(100);

if (chance >85)

{

var spawnPosition = this.Owner.Transform.Translation;

var archetypeToSpawn = Archetype.Find("Bullet\_3");

var createdBullet = this.Space.CreateAtPosition(archetypeToSpawn, spawnPosition);

}

}

}

}

class Enemy\_movement : ZilchComponent

{

var Time : Real = 0;

function Initialize(init : CogInitializer)

{

Zero.Connect(this.Space, Events.LogicUpdate, this.OnLogicUpdate);

}

function OnLogicUpdate(event : UpdateEvent)

{

this.Time +=1;

if(this.Time == 50)

{

var random = new Random();

var chance = random.DieRoll(100);

if (chance >75)

{

var temp = this.Owner.Transform.Translation;

temp.X +=-5;

this.Owner.Transform.Translation =temp;

}

else if (chance <25)

{

var temp2 = this.Owner.Transform.Translation;

temp2.X +=5;

this.Owner.Transform.Translation =temp2;

}

this.Time = 0;

}

}

}

class Interaction : ZilchComponent

{

function Initialize(init : CogInitializer)

{

//Zero.Connect(this.Space, Events.LogicUpdate, this.OnLogicUpdate);

Zero.Connect(this.Owner, Events.CollisionStarted, this.OnCollisionStarted);

}

function OnCollisionStarted(event : CollisionEvent)

{

var otherObject = event.OtherObject;

if (otherObject.Name == "Enemy")

{

otherObject.Destroy();

if (this.Owner.Name == "Bullet 2")

{

this.Owner.Destroy();

}

}

}

function OnLogicUpdate(event : UpdateEvent)

{

}

}

class Interaction\_2 : ZilchComponent

{

var Level\_One: Level = Level.Find("Level\_One");

function Initialize(init : CogInitializer)

{

//Zero.Connect(this.Space, Events.LogicUpdate, this.OnLogicUpdate);

Zero.Connect(this.Owner, Events.CollisionStarted, this.OnCollisionStarted);

}

function OnCollisionStarted(event : CollisionEvent)

{

var otherObject = event.OtherObject;

if (otherObject.Name == "Player")

{

this.Space.LoadLevel(this.Level\_One);

}

else if (otherObject.Name == "Bullet")

{

otherObject.Destroy();

}

else if (otherObject.Name == "assist")

{

otherObject.Destroy();

this.Owner.Destroy();

}

}

}

class Levels2 : ZilchComponent

{

var Level\_One: Level = Level.Find("Level\_One");

var Level\_5: Level = Level.Find("Level\_5");

var Level\_6: Level = Level.Find("Level\_6");

var Level\_7: Level = Level.Find("Level\_7");

var Level\_8: Level = Level.Find("Level\_8");

var Level\_9: Level = Level.Find("Level\_9");

var Level\_10: Level = Level.Find("Level\_10");

var Level\_Boss: Level = Level.Find("Level\_Boss");

var Level\_End: Level = Level.Find("Level\_End");

var Level\_Final: Level = Level.Find("Level\_Final");

var Level\_Victory: Level = Level.Find("Level\_Victory");

function Initialize(init : CogInitializer)

{

//Zero.Connect(this.Space, Events.LogicUpdate, this.OnLogicUpdate);

Zero.Connect(Zero.Keyboard, Events.KeyUp, this.OnKeyUp);

}

function OnKeyUp(event : KeyboardEvent)

{

if (event.Key == Keys.R)

{

this.Space.ReloadLevel();

}

else if (event.Key == Keys.Right)

{

var currentLevelName = this.Space.CurrentLevel;

if (currentLevelName == this.Level\_5)

{

this.Space.LoadLevel(this.Level\_6);

}

else if (currentLevelName == this.Level\_6)

{

this.Space.LoadLevel(this.Level\_7);

}

else if (currentLevelName == this.Level\_7)

{

this.Space.LoadLevel(this.Level\_8);

}

else if (currentLevelName == this.Level\_8)

{

this.Space.LoadLevel(this.Level\_9);

}

else if (currentLevelName == this.Level\_9)

{

this.Space.LoadLevel(this.Level\_10);

}

else if (currentLevelName == this.Level\_10)

{

this.Space.LoadLevel(this.Level\_Boss);

}

else if (currentLevelName == this.Level\_Boss)

{

this.Space.LoadLevel(this.Level\_End);

}

else if (currentLevelName == this.Level\_End)

{

this.Space.LoadLevel(this.Level\_Final);

}

else if (currentLevelName == this.Level\_Final)

{

this.Space.LoadLevel(this.Level\_Victory);

}

else if (currentLevelName == this.Level\_Victory)

{

this.Space.LoadLevel(this.Level\_One);

}

}

}

}

class Shield : ZilchComponent

{

function Initialize(init : CogInitializer)

{

//Zero.Connect(this.Space, Events.LogicUpdate, this.OnLogicUpdate);

Zero.Connect(Zero.Keyboard, Events.KeyUp, this.OnKeyDown);

Zero.Connect(this.Owner, Events.CollisionStarted, this.OnCollisionStarted);

}

function OnKeyDown(event : KeyboardEvent)

{

if(Zero.Keyboard.KeyIsDown(Keys.L))

{

this.Owner.Sprite.Visible = true;

this.Owner.SphereCollider.SendsEvents = true;

}

else if(Zero.Keyboard.KeyIsUp(Keys.L))

{

this.Owner.Sprite.Visible = false;

this.Owner.SphereCollider.SendsEvents = false;

}

}

function OnCollisionStarted(event : CollisionEvent)

{

var otherObject = event.OtherObject;

if (otherObject.Name == "Bullet\_3")

{

otherObject.Destroy();

}

}

}

class Stop : ZilchComponent

{

function Initialize(init : CogInitializer)

{

//Zero.Connect(this.Space, Events.LogicUpdate, this.OnLogicUpdate);

Zero.Connect(this.Owner, Events.CollisionStarted, this.OnCollisionStarted);

}

function OnCollisionStarted(event : CollisionEvent)

{

var otherObject = event.OtherObject;

if (otherObject.Name == "Bullet\_3")

{

otherObject.Destroy();

}

}

}

class Time\_spawner : ZilchComponent

{

var Time : Real = 0;

function Initialize(init : CogInitializer)

{

Zero.Connect(this.Space, Events.LogicUpdate, this.OnLogicUpdate);

}

function OnLogicUpdate(event : UpdateEvent)

{

this.Time +=1;

if(this.Time == 150)

{

var random = new Random();

var chance = random.DieRoll(100);

if (chance >85)

{

var spawnPosition = this.Owner.Transform.Translation;

var archetypeToSpawn = Archetype.Find("Bullet\_3");

var createdBullet = this.Space.CreateAtPosition(archetypeToSpawn, spawnPosition);

}

this.Time = 0;

}

}

}

class Levels : ZilchComponent

{

var Level\_One: Level = Level.Find("Level\_One");

var Level\_2: Level = Level.Find("Level\_2");

var Level\_3: Level = Level.Find("Level\_3");

var Level\_4: Level = Level.Find("Level\_4");

var Level\_5: Level = Level.Find("Level\_5");

function Initialize(init : CogInitializer)

{

//Zero.Connect(this.Space, Events.LogicUpdate, this.OnLogicUpdate);

Zero.Connect(Zero.Keyboard, Events.KeyUp, this.OnKeyUp);

}

function OnKeyUp(event : KeyboardEvent)

{

if (event.Key == Keys.R)

{

this.Space.ReloadLevel();

}

else if (event.Key == Keys.Right)

{

var currentLevelName = this.Space.CurrentLevel;

if (currentLevelName == this.Level\_One)

{

this.Space.LoadLevel(this.Level\_2);

}

else if (currentLevelName == this.Level\_2)

{

this.Space.LoadLevel(this.Level\_3);

}

else if (currentLevelName == this.Level\_3)

{

this.Space.LoadLevel(this.Level\_4);

}

else if (currentLevelName == this.Level\_4)

{

this.Space.LoadLevel(this.Level\_5);

}

}

}

}