Unity RPG Tutorial #12 – Player Killing and Enemy Movement

1. Now go update your skeletonController script, under assests in Unity.

Slash out //timeBetweenMoveCounter = timeBetweenMove;

//timeToMoveCounter = timeTomove

In void Start() and update to.

timeBetweenMoveCounter = Random.Range (timeBetweenMove \* 0.1f, timeBetweenMove \* 1.25f);

timeToMoveCounter = Random.Range(timeToMove \* 0.75f, timeBetweenMove \* 1.25f);

This is great and can be changed between 0 – 2 f to change the randomness to what you like.

Save the script.

1. Copy timeToMoveCounter = Random.Range(timeToMove \* 0.75f, timeBetweenMove \* 1.25f);

And timeBetweenMoveCounter = Random.Range (timeBetweenMove \* 0.1f, timeBetweenMove \* 1.25f); to the following statements in the same script.

If (moving)

{

timeToMoveCounter -= Time.deltaTime;

myRigidbody.velocity = moveDirection;

if(timeToMoveCounter < 0f)

{

Moving = false;

//timeBetweenMoveCounter = timeBetweenMove;

timeBetweenMoveCounter = Random.Range (timeBetweenMove \* 0.1f, timeBetweenMove \* 1.25f);

}

} else {

timeBetweenMoveCounter -= Time.deltaTime;

myRigidbody.velocity = Vector2.zero;

if(timeBetweenMoveCounter < 0f)

{

moving = true;

//timeToMoveCounter = timeToMove;

timeToMoveCounter = Random.Range(timeToMove \* 0.75f, timeBetweenMove \* 1.25f);

…

}

}

}

save the script.

Now add this function below.

void onCollisionEnter2D(Collision2D other){

if(other.gameObject.name == “Player”)

{

Destroy(other.gameObject);

..

To destroy the game Object, which would in turn need a game over screen to reinitialize the game object. Slash this out.

Put

other.gameObject.SetActive(false); instead, and below the class put

public float waitToReload;

private bool reloading;

and go back.

void onCollisionEnter2D(Collision2D other){

if(other.gameObject.name == “Player”)

{

other.gameObject.SetActive(false);

reloading = true;

}

Above put

If(timeBetweenMoveCounter < 0f)

{

moving = true;

//timeToMoveCounter = timeToMove;

timeToMoveCounter = Random.Range(timeToMove \* 0.75f, timeBetweenMove \* 1.25f);

…

}

}

If(reloading)

{

waitToReload -= Time.deltaTime;

if(waitToReload < 0)

{

Application.LoadLevel(Application.loadedLevel);

}

}

}

Go back up to the variables below class type

Private GameObject thePlayer;

Now back to..

void onCollisionEnter2D(Collision2D other){

if(other.gameObject.name == “Player”)

{

other.gameObject.SetActive(false);

reloading = true;

thePlayer = other.gameObject;

}

Now back to

If(reloading)

{

waitToReload -= Time.deltaTime;

if(waitToReload < 0)

{

Application.LoadLevel(Application.loadedLevel);

thePlayer.SetActive(true);

}

}

}

.

1. Now Highlight all your Enemy Skeletons, and set waitToReload in the inspector to how long it will take to respawn.

A simple way of dying and respawning in the game. This is a great system for traps in the level that are set active.