

Tutorial 2 – 3D Run Away Script

This tutorial will go through creating a script that causes an object to run away from another object.

The first step is to create two objects, one for the object that will be running, and one for the object that it is running from. These should both be 3D objects, with colliders and rigidbodies. Add a tag to the object that is being run from, in my case I used “Player”.

On the runner, add two public floats; one for visibility distance and one for the running speed. Also add a public transform for the object to run from and a rigidbody variable.

```
public float AwareDist = 10f;
public float speed = 5.5f;

public Transform player;

Rigidbody rigidbody;
```

In start, add this line of code to get the object’s rigidbody.

```
rigidbody = GetComponent<Rigidbody>();
```

In update, create three variables, a float called dist, and two Vector3s, one called dir and one called moveDir. Dist contains the distance between the two objects, dir gets the direction from the runner to the object, and moveDir gets the direction from the object to the runner. Dist is used to see if the two objects are close enough to run, dir is used to raycast from the runner and moveDir is the direction that the runner runs if it should be running.

```
// get the distance between the player and the runner
float dist = Vector3.Distance(transform.position, player.position);
// get the direction from the runner to the player
Vector3 dir = Vector3.Normalize(new Vector3((player.position - transform.position).x, 0, (player.position - transform.position).z));
// get the direction from the player to the runner
Vector3 moveDir = Vector3.Normalize(new Vector3((transform.position - player.position).x, 0, (transform.position - player.position).z));
```

Next up, create an if statement that checks if the dist variable is less than or equal to the visibility distance (in my case AwareDist). This ensures that the runner and object are close enough to run.

Inside the if statement, create and check a RaycastHit. If this ray hits anything, store the hit object as a variable and check the tag of the variable. If the tag is the tag of the object (in my case “Player”), set the rigidbody’s velocity to movedir * speed * Time.deltaTime.

```
// create a var to hold raycast data
RaycastHit hit;
// create a raycast from the sheep in dir at a set distance and see if it hits anything - save hit data to hit
if(Physics.Raycast(transform.position, dir * AwareDist, out hit))
{
    // make a var to save the game object that was hit
    GameObject obj = hit.collider.gameObject;

    // if the object was the player...
    if(obj.tag == "Player")
    {
        // move away from the player at speed speed
        rigidbody.velocity = (moveDir * speed * Time.deltaTime);
    }
}
```