CREATE GAMES !

Welcome to this issue of Create Games! It's Amazing, 3 months have passed and we have learned a lot from the first 2 issues. We will go through all new scripts, explaining what it does and how. You can see the game creation itself at: https://youtu.be/tAQyZKoAqI8

In this issue we will create one of those knife throwing games. You can throw a knife at a rotating log and when it hits the log the knife will stick to it. Hit another knife and the knife will fall down. The rotating log rotates clockwise and counterclockwise at random times to make it harder.

Ready let's inspect those scripts.

GameManager Script.

The first three lines are the standard libraries of unity itself. We added one more library that holds all the stuff to switch between scenes. (SceneManagement)

```
///Knive throwing game Script by René Pol RP-Interactive.nl© 01-06-2021//////////

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class GameManager : MonoBehaviour
{
```

A float variable is used so you can see the number of knives that can be thrown.

```
public float NumberOfKnives;
```

In the update function if the R key is pressed and the float variable NumberOfKnives is equal to 0 it will reload the scene it is playing.

```
void Update()
{
if(Input.GetKeyDown(KeyCode.R) && NumberOfKnives == 0)
{
SceneManager.LoadScene("SampleScene");
}
}
}
```

In the inspector's view you can set the number of knives to throw. In my example it's 8.

Log Script.

```
///Knive throwing game Script by René Pol RP-Interactive.nl© 01-06-2021//////////

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Log : MonoBehaviour
{
```

For the rotation speed a float variable is created called Speed and it is set to 1. A Bool variable is created to determine if the rotation should be Clockwise(true) or counter Clockwise (false). Last is another float variable that sets the duration time of the rotating. At start it is set to 3.

```
public float Speed =1f;
public bool Clockwise = true;
public float duration = 3;
```

In the update function the float variable duration will count down until it hits 0. When its 0 and the bool variable is set to true it will give the duration a random time starting with 5 + A random number between 2 and 4. It sets the Bool variable to the opposite statement that it was so true becomes false. Return makes sure it won't go to the next part as the instruction is ready and should be beginning from start when duration hits zero. When the bool variable is false and duration is equal or smaller than 0 it repeats all.

```
void Update()
{
duration -= Time.deltaTime;
if(duration <= 0)
{
if(Clockwise == true)
{
Clockwise = false;
duration = 5+ Random.Range(2,4);
return;
}
if (Clockwise == false)
{
Clockwise = true;
duration = 5+ Random.Range(2,4);
return;
}
}
```

When duration is bigger than 0 and the bool variable is true the log will rotate to the right on the Z-axis with the speed of the speed variable 50-degree rotation continuously. Time.deltaTime remember? makes it happen at the same speed on different systems.
If duration is bigger than 0 and the bool variable is false it rotates the opposite way. (the minus – symbol in front of the degrees number)

```
if(duration > 0 && Clockwise == true)
{
transform.Rotate(new Vector3(0, 0, -50 * Speed * Time.deltaTime));
}
if (duration > 0 && Clockwise == false)
{
transform.Rotate(new Vector3(0, 0, 50 * Speed * Time.deltaTime));
}
}
}
```

You can set the speed and duration in the inspector’s view. This log will need a circle collider2D so it is able to have collision with the knives. You could insert a variable that changes the rotation speed, all is up to your imagination and the challenge you want the game to be.

Next and last for this game project will be the knife script. Pay good attention cause a lot of stuff is happening in this Script. The knife sprite needs a BoxColllider2D and a Rigidbody2D with its gravity set to 0 and its Z constraint box selected. Remember for collision at least one out of the 2 game objects need a rigidbody.

Knife Script.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Knives : MonoBehaviour
{
```

You should know how and what a float,bool and int variable is.

```
public float speed;
public bool thrown = false;
public int move = 0;
public bool IsFalling = false;
```

A gameObject to use is defined and is called Newknive.

```
public GameObject Newknive;
```

3 Sounds to use are defined with the names Knive,Hit,Hitmetal
Instead of making 3 lines it is done with one simply by adding the comma, between the names.

```
public AudioClip Knive,Hit,Hitmetal;
```

A private defining to another script (GameManager) is made and it is called GR.

```
private GameManager GR;
```

A Rigidbody2D is defined with the name rb;

```
public Rigidbody2D rb;
```

In the start function (Begins when the object is in the scene) the Isfalling and thrown bool are both set to 0. The Rigidbody2D we defined is called. The move variable is set to 0 as well as the gravity of the Rigidbody2D. A connection to that other script is made.

```
void Start()
{
IsFalling = false;
rb = GetComponent<Rigidbody2D>();
move = 0;
thrown = false;
GR = FindObjectOfType<GameManager>();
rb.gravityScale = 0;
}
```

If the knife collides with an object that is called (Uses tag) Hit and the move variable is equal to 0 it plays the hit metal sound once and sets the IsFalling bool variable to true. Then the BoxCollider2D it uses gets deactivated (Disabled), gravity is set to 1 and the move variable is set to 1. Now the knife will fall down.

```
void OnCollisionEnter2D(Collision2D other)
{
if (other.gameObject.tag == ("Hit") && move == 0)
{
AudioSource.PlayClipAtPoint(Hitmetal, new Vector3(0, 0, -10));
IsFalling = true;
GetComponent<BoxCollider2D>().enabled = true;
rb.gravityScale = 1;
move = 1;
}
```

If the knife collides with an object that is called (Uses tag) Log the knive name tag will be changed into Hit and it plays the Hit sound once. A tag can only change if the tag exists in the tag list in the inspector's view. Then the BoxCollider2D it uses gets deactivated (Disabled), gravity is set to 1 and the move variable is set to 1. If the move variable is equal to 0 and the numberofknives variable(In gamemanager script) is bigger than 0 it will instantiate(Create) a new knife (The defined gameObject) at the given coordinate with no rotation set to it.

```
if (other.gameObject.tag == ("Log"))
{
gameObject.tag = "Hit";
AudioSource.PlayClipAtPoint(Hit, new Vector3(0, 0, -10));
if (move == 0 && GR.NumberOfKnives > 0)
{
Instantiate(Newknive, new Vector3(0, -4, 0), Quaternion.identity);
}
```

The body type of the Rigidbody is to isKinematic so it wont react when it get's hit by a physics object. The velocity is set to 0 and the object becomes a child of the object it hit. `transform.SetParent(other.transform);` the move variable is set to 1

```
rb.isKinematic = true;
rb.velocity = Vector2.zero;
transform.SetParent(other.transform);
move = 1;
}
}
```

In the Update function we keep track on a lot of thing. If the Isfalling bool variable is true (So it hitted a knife) the object will rotate while falling with a speed of 10.

```
void Update()
{
if(IsFalling == true)
{
transform.Rotate(new Vector3(0, 0, -50 * 10 * Time.deltaTime));
}
```

When the knife position on the y-axis is smaller than -10.5f it will set its velocity to 0 and the gravity back to 0. The thrown variable is set to false.

```
if(transform.position.y < -10.5f)
{
rb.velocity = Vector2.zero;
rb.gravityScale = 0;
thrown = false;
```

if the Numberofknives variable (In gameManager script) is bigger than 0 a new knife (The defined gameObject) is created at the given coordinates with no rotation set to it. The original gameObject is destroyed (Removed)

```
if ( GR.NumberOfKnives > 0)
{
Instantiate(Newknive, new Vector3(0, -4, 0), Quaternion.identity);
}
Destroy(this.gameObject);
}
```

When the spacebar is pressed and the thrown variable is false and the move variable is equal to 0 and the Numberofknives variable is bigger than 0. 1 is subtracted of the Numberofknives the thrown variable is set to true and the knive sound is playing once.

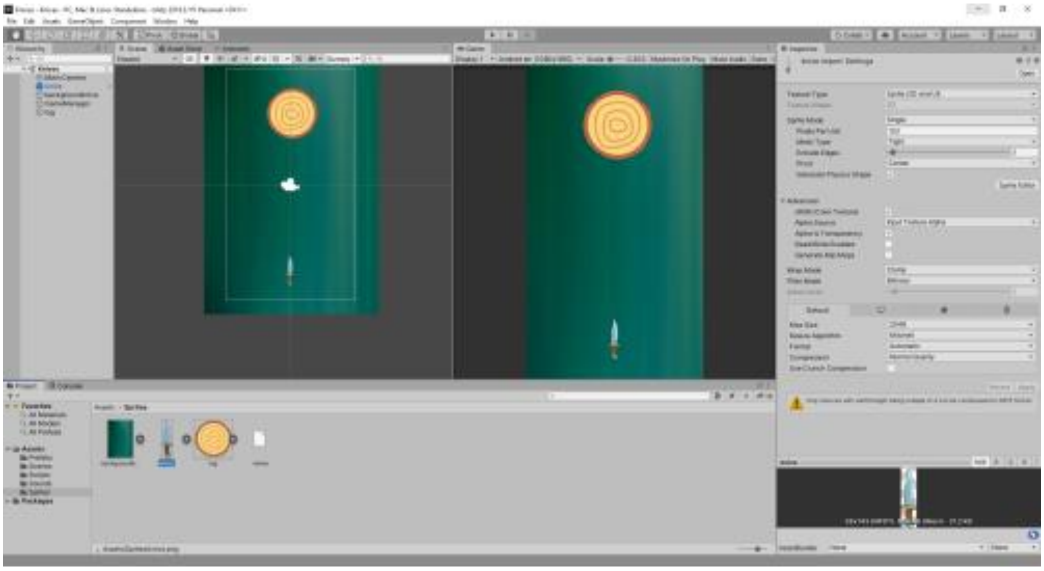
```
if (Input.GetKeyDown(KeyCode.Space) && thrown == false && move ==0 &&
GR.NumberOfKnives >0)
{
GR.NumberOfKnives -= 1;
AudioSource.PlayClipAtPoint(Knive, new Vector3(0, 0, -10));
thrown = true;
}
```

And when thrown is true and move is equal to 0 the game Object will simply move with the given speed up, till it hits something.

```
if(thrown == true && move ==0)
{
transform.Translate(Vector2.up * speed * Time.deltaTime);
}
}
}
```

Create a prefab with the first knife gameObject. You can drag all the sounds and set its variables the way you need them. This gameObject will respawn itself when needed.

You can alter and change this game in so many ways. Change the target or make arrow's shoot the target. Add targets on the rotating object by making them child objects. Throw Horizontal instead of vertical, the only limit you have is your own imagination.



How To ?

The hero uses ropes to go over holes in the ground or to climb into the rope to higer places. In the game Cut the rope little Omnom wants candy that dangles on a rope. The only way to get it is to cut the rope. In how to we will create a rope by using HingJoint2D so the rope will have real physics. How it all works you can see at :

https://www.youtube.com/watch?v=5nO9PFQppkc

To create a rope that uses real physics you must understand that in this example will exist out of 3 different parts

- The hook (Static object where the rope is hanging from)
- Link prefab (Rope parts stored in the prefab folder that can be linked together)
- Candy (Weight that is connected to the end of the rope)

I placed the pin sprite and gave it this script :

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Rope : MonoBehaviour
{
```

We give the pin object a Rigidbody2D and we name it hook. In unity we add this Rigidbody2D and make it Kinematic so it will use physics but not fall down. A piece of rope was created and stored in the prefab folder so this game object is used to create the full length rope. It's called linkPrefab.

```
public Rigidbody2D hook;
public GameObject linkPrefab;
```

the third object used is the Candy but it wio be a separate model with it's own script. So we refer to it by calling it candy. To set the ammount of rope pieces (Links) we create an int variable called links.

```
public Candy candy;
public int links = 7;
```

In the start function I made the GenerateRope function start immidiatly.

```
void Start()
{
GenerateRope();
}
```

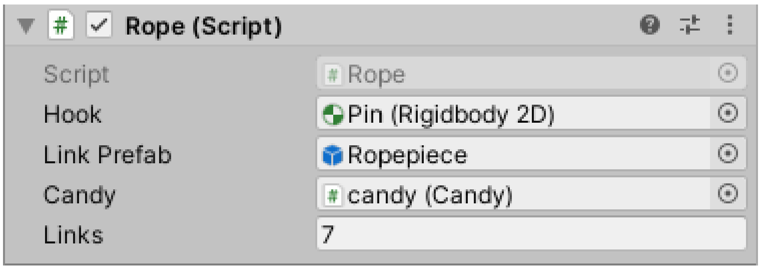
Then I created this function. It will create a rope piece (Link) from the object that is up from it. So the first part has the hook to connect to and the seccond will use the link and so on. It uses the prefab link parts and the number of links is set with the int variable. The rope piece will use it's HingeJoint2D to connect to the other pieces.

```
void GenerateRope()
{
Rigidbody2D previousRB = hook;
for (int i = 0; i < links; i++)
{
GameObject link = Instantiate(linkPrefab, transform);
HingeJoint2D joint = link.GetComponent<HingeJoint2D>();
joint.connectedBody = previousRB;
```

When the ammount of the link variable is reached it will use the Candy as last link to the rope.

```
if (i < links - 1)
{
previousRB = link.GetComponent<Rigidbody2D>();
}
else
{
candy.ConnectRopeEnd(link.GetComponent<Rigidbody2D>());
}
}
}
}
```

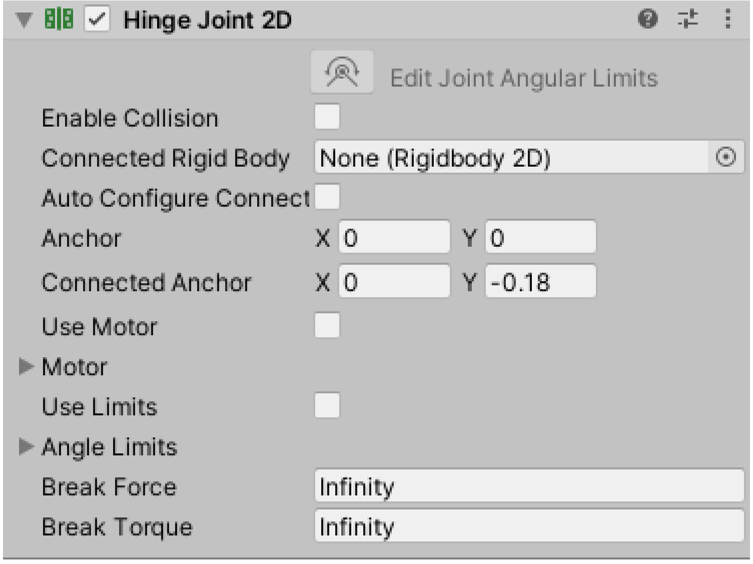
In the inspectors view you are able to drag and drop all parts needed.



Rope (Script)

Script	Rope
Hook	Pin (Rigidbody 2D)
Link Prefab	Ropepiece
Candy	candy (Candy)
Links	7

The link prefab is a gameObject that holds a Rigidbody2D with it's gravity set to 1, Linear drag 0.2 and its angular drag 0.1. You can use any settings that you need to make it look like you want. Just toy around with these settings. The tag name should be Link so it can be cut later. Finally the HingeJoint 2D is needed make sure it uses these settings :



☑ Hinge Joint 2D

Enable Collision	□
Connected Rigid Body	None (Rigidbody 2D)
Auto Configure Connect	□
Anchor	X 0 Y 0
Connected Anchor	X 0 Y -0.18
Use Motor	□
▶ Motor	
Use Limits	□
▶ Angle Limits	
Break Force	Infinity
Break Torque	Infinity

Make sure the link has a box collider 2D, this will be needed for cutting the rope parts.

So the Candy gameobject will be the end of the rope and it will have some weight to it. I created the candy sprite and gave it a circle collider 2D with the is Trigger box selected. The name tag is Candy and the Rigidbody2D it has will have a heavier mass then the link parts so I set it to 5. Finally the candy gets this script :

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Candy : MonoBehaviour
{
```

A float variable is created for where the candy should be connected to the previous link.

```
public float distanceFromChainEnd = 0f;
```

A new public function is created that will be called from the main rope script its uses the rigidbody2D named endRB.

```
public void ConnectRopeEnd(Rigidbody2D endRB)
{
```

The candy is connected to the last HingeJoint2D at the distance that is set in the float variable.the auto configure is set to false as wel as the Vector2 that is set to 0. Now the candy will be the last object on the rope so it will swing.

```
HingeJoint2D joint = gameObject.AddComponent<HingeJoint2D>();
joint.autoConfigureConnectedAnchor = false;
joint.connectedBody = endRB;
joint.anchor = Vector2.zero;
joint.connectedAnchor = new Vector2(0f, -distanceFromChainEnd);
}
}
```

Almost there. I created an empty game object and called it ropecutter. This will contain a script that makes it able to cut rope parts with the click of a mousebutton. I created this script :

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Ropecutter : MonoBehaviour
{
```

An audio file used is defined by the name cutting.

```
public AudioClip cutting;
```

In the Update function every time the mouse button is clicked it will create a raycast2D (line) to the point where the mouse(Cursor) is at that exact moment. When it hits nothing, nothing happens but when it hits an object with the nametag Link it will play the audio file on the given location and it destroys the rope link. Since the rope uses physics it will fall down depending if there is a weight (Candy) or pin attached to it.

```
void Update()
{
if (Input.GetMouseButton(0))
{
RaycastHit2D hit = Physics2D.Raycast(Camera.main.ScreenToWorldPoint(Input.mousePosition), Vector2.zero);
if (hit.collider != null)
{
if (hit.collider.tag == "Link")
{
AudioSource.PlayClipAtPoint(cutting, new Vector3(0, 0, -10));
Destroy(hit.collider.gameObject);
}
}
}
}
}
```

The game character Omnom has a simple box collider 2D and uses 2 sprites. One for waiting with an open mouth and a seccond where its mouth is closed (Eating). A simple script is made so it changes when the candy is eaten. This is the script :

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class OmNom : MonoBehaviour
{
```

I defined 2 sprite to use the Openmouth and Eat one. Also a sound file is defined called eating.

```
public Sprite Openmouth, Eat;
public AudioClip eating;
```

At start the openmouth sprite is used we defined it before.

```
void Start()
{
this.GetComponent<SpriteRenderer>().sprite = Openmouth;
}
```

When triggered by a game object with the name tag Candy the candy will be destroyed and the soundfile will play at given coordinates. The sprite use will be Eat.

```
void OnTriggerEnter2D(Collider2D other)
{
if(other.gameObject.tag == ("Candy"))
{
Destroy(other.gameObject);
AudioSource.PlayClipAtPoint(eating, new Vector3(0, 0, -10));
this.GetComponent<SpriteRenderer>().sprite = Eat;
}
}
```

All the basics for creating Ropes and or chains are here for you to play with.
I'll hope you learned from it and have just as much fun in creating this as I did. I will see you in the next issue.

All of the assets used in this issue can be downloaded here : https://rp-interactive.nl/magazine.html

All scripts used in issue 03 are here for you.

Log Script.

```
///Knive throwing game Script by René Pol RP-Interactive.nl© 01-06-2021//////////

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Log : MonoBehaviour
{
public float Speed =1f;
public bool Clockwise = true;
public float duration = 3;

void Update()
{
duration -= Time.deltaTime;
if(duration <= 0)
{
if(Clockwise == true)
{
Clockwise = false;
duration = 5+ Random.Range(2,4);
return;
}
if (Clockwise == false)
{
Clockwise = true;
duration = 5+ Random.Range(2,4);
return;
}
}
if(duration > 0 && Clockwise == true)
{
transform.Rotate(new Vector3(0, 0, -50 * Speed * Time.deltaTime));
}
if (duration > 0 && Clockwise == false)
{
transform.Rotate(new Vector3(0, 0, 50 * Speed * Time.deltaTime));
}
}
}
```

Knives Script.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

///Knive throwing game Script by René Pol RP-Interactive.nl© 01-06-2021//////////

public class Knives : MonoBehaviour
{
public float speed;
public bool thrown = false;
public int move = 0;
public GameObject Newknive;
public AudioClip Knive,Hit,Hitmetal;
public bool IsFalling = false;
private GameManager GR;
public Rigidbody2D rb;

void Start()
{
IsFalling = false;
rb = GetComponent<Rigidbody2D>();
move = 0;
thrown = false;
GR = FindObjectOfType<GameManager>();
rb.gravityScale = 0;
}

void OnCollisionEnter2D(Collision2D other)
{
if (other.gameObject.tag == ("Hit") && move == 0)
{
AudioSource.PlayClipAtPoint(Hitmetal, new Vector3(0, 0, -10));
IsFalling = true;
GetComponent<BoxCollider2D>().enabled = true;
rb.gravityScale = 1;
move = 1;
}
if (other.gameObject.tag == ("Log"))
{
gameObject.tag = "Hit";
AudioSource.PlayClipAtPoint(Hit, new Vector3(0, 0, -10));
if (move == 0 && GR.NumberOfKnives > 0)
{
Instantiate(Newknive, new Vector3(0, -4, 0), Quaternion.identity);
}
rb.isKinematic = true;
rb.velocity = Vector2.zero;
transform.SetParent(other.transform);
move = 1;
}
}
```

```
void Update()
{
if(IsFalling == true)
{
transform.Rotate(new Vector3(0, 0, -50 * 10 * Time.deltaTime));
}
if(transform.position.y < -10.5f)
{
rb.velocity = Vector2.zero;
rb.gravityScale = 0;
thrown = false;
if ( GR.NumberOfKnives > 0)
{
Instantiate(Newknive, new Vector3(0, -4, 0), Quaternion.identity);
}
Destroy(this.gameObject);
}
if (Input.GetKeyDown(KeyCode.Space) && thrown == false && move ==0 &&
GR.NumberOfKnives >0)
{
GR.NumberOfKnives -= 1;
AudioSource.PlayClipAtPoint(Knive, new Vector3(0, 0, -10));
thrown = true;
}
if(thrown == true && move ==0)
{
transform.Translate(Vector2.up * speed * Time.deltaTime);
}
}
}
```

Game Manager Script.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;


///Knive throwing game Script by René Pol RP-Interactive.nl© 01-06-2021//////////

public class GameManager : MonoBehaviour
{
public float NumberOfKnives;

void Update()
{
if(Input.GetKeyDown(KeyCode.R) && NumberOfKnives == 0)
{
SceneManager.LoadScene("SampleScene");
}
}
}
```

Omnom Script.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

/// All about ropes by René Pol //////////

public class OmNom : MonoBehaviour
{
public Sprite Openmouth, Eat;
public AudioClip eating;

void Start()
{
this.GetComponent<SpriteRenderer>().sprite = Openmouth;
}

void OnTriggerEnter2D(Collider2D other)
{
if(other.gameObject.tag == ("Candy"))
{
Destroy(other.gameObject);
AudioSource.PlayClipAtPoint(eating, new Vector3(0, 0, -10));
this.GetComponent<SpriteRenderer>().sprite = Eat;
}
}
}
```

Rope Script.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Rope : MonoBehaviour
{
/// All about ropes by René Pol //////////

public Rigidbody2D hook;
public GameObject linkPrefab;
public Candy candy;
public int links = 7;

void Start()
{
GenerateRope();
}
```

```
void GenerateRope()
{
Rigidbody2D previousRB = hook;
for (int i = 0; i < links; i++)
{
GameObject link = Instantiate(linkPrefab, transform);
HingeJoint2D joint = link.GetComponent<HingeJoint2D>();
joint.connectedBody = previousRB;
if (i < links - 1)
{
previousRB = link.GetComponent<Rigidbody2D>();
}
else
{
candy.ConnectRopeEnd(link.GetComponent<Rigidbody2D>());
}
}
}
}
```

RopeCutter Script.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

/// All about ropes by René Pol //////////

public class Ropecutter : MonoBehaviour
{
public AudioClip cutting;

void Update()
{
if (Input.GetMouseButton(0))
{
RaycastHit2D hit =
Physics2D.Raycast(Camera.main.ScreenToWorldPoint(Input.mousePosition),
Vector2.zero);
if (hit.collider != null)
{
if (hit.collider.tag == "Link")
{
AudioSource.PlayClipAtPoint(cutting, new Vector3(0, 0, -10));
Destroy(hit.collider.gameObject);
}
}
}
}
}
```

Candy Script.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Candy : MonoBehaviour
{
/// All about ropes by René Pol //////////

public float distanceFromChainEnd = 0f;

public void ConnectRopeEnd(Rigidbody2D endRB)
{
HingeJoint2D joint = gameObject.AddComponent<HingeJoint2D>();
joint.autoConfigureConnectedAnchor = false;
joint.connectedBody = endRB;
joint.anchor = Vector2.zero;
joint.connectedAnchor = new Vector2(0f, -distanceFromChainEnd);
}
}
```