GRABBING for GMO (Get Me Out)

Basic mechanic:

The whole grabbing mechanic is supported by one physic element: a hinge joint. It's literally a joint that connects two Rigidbodies together. It is mostly used to rotate doors around a wall to open or close it.

We will use hinge joints for the grabbing mechanic of GMO – Get Me Out. A few reasons why:

- The Physics is handled by Unity, so it feels way better
- The mechanic is handled by Unity, so there are far less bugs
- It is handled by Unity, so it is way easier.

For the grabbing to work, both the player and the "anchor" (aka the object you grab) must have a rigidbody **and** a hinge joint.

We will detail how to set them up in the different situations we can face.

Disclaimer: all the different grabbing scripts should be attached to the dummies of both players. The grabbable objects should have (as a child) a trigger collider (make sure isTrigger is checked/true) determining an appropriate interaction zone, and an appropriate tag referenced in the grabbing script.

Grabbing objects:

Grabbing objects is the easiest. Both objects have a hinge joint, there connected bodies being each other (eg. vyle -> player and player -> vyle). Pseudo code is as follows:

1. If player collides with interaction transform of the object
2. Set the position of the object on the head of the player (reuse my old script)
3. **Set its rotation to Quaternion.Identity** for the objects to be face up
4. Add a hinge joint to the player (the object should already have one, or not, in that case add one to it)
5. Complete the parameters of the hinge joint
   https://docs.unity3d.com/ScriptReference/HingeJoint.html

Grabbing players:

Same thing as the grabbing object, though you have to add the hinge joint to both players, deactivate the control input of the player that is grabbed, and add the functionality that if the player on top jumps then the hinge joint is destroyed.

Swinging:

A bit more complicated. There should be a swinging rope, composed of small rigidbodies (like chains), each element having a hinge joint. The connected body of the hinge joint

should be the rigidbody of the chain / rope element above it. The top chain should have the ceiling / an anchor / an empty game object with a rigidbody as the connected body. This ceiling / anchor should be kinematic.

The more chains you will have, the damper / smoother the swinging will be.

Each chain should use a spring, have a damper of 50, anchor axis of Vector3.back (0, 0, -1). Everything else should be left as default, except if it makes the swinging better.

The bottom chain, so the end of the rope, should have a trigger: it is an additional trigger collider determining the interaction transform that enables the swinging.

When the player enters the trigger collider:

- Add a hinge joint to the player
- Set its anchor axis to Vector3.back
- Set its connected body to the end of the rope
- Set its useSpring to true
- Set the damper of the spring to true.

I invite you to modify and test the settings of the hinge joint to obtain a better swinging.

```
void Swing()
{

    // Add Hinge Joint on the player with correct settings
    HingeJoint joint = gameObject.AddComponent<HingeJoint>();
    joint.axis = Vector3.back; /// (0,0,-1)
    joint.anchor = Vector3.zero;

    joint.useSpring = true;
    JointSpring hingeSpring = joint.spring;
    hingeSpring.damper = 50;

    // Connect the player to the rope
    joint.connectedBody = rope.GetComponent<Rigidbody>();
}
```

Then, when the player hits the control “Jump” you should do the inverse process: remove the hingeJoint.

Disclaimer: you cannot just hide or disable the hinge joint. You have to remove it and re-add it each time.

Current bugs:

- The swinging slows down too much / stops when entering the center / minimum y value. Maybe a mass or spring problem.
- I don’t have the exit of the swinging set up yet.

For more references, check out the Robin Test Scene or ask Robin directly.