Documentation P3
Opening objects of daily use.

Milestone 1

Milestone 1 Description

- Creating an editor plugin for a screw joint between 2 mesh actors
- Screw joint constraints the movement in one direction
- Destroy the constraint after moving the lid by a fixed degree

The milestone was set mainly to have also time, getting used to the unreal engine.

Work of Milestone 1

This Milestone has been completely finished. It was possible to break the lid fixed at 60 degrees. Other Degrees couldn't be possible, since it changes after 180 degrees to a negative value and the lid should only turn one direction. Figure 1 shows how the degree on swing 1 of the physical constraint works. It shows that if the rotation is to the left, there will be a negative value, on the right a positive once.

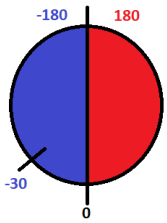

Figure 1

To make sure that the lid can't be turned on the left, it gets a 30 degree offset and a limit of 30 degrees. The 30 degree limit makes it possible to turn 30 degrees in both directions, left and right, so the offset makes it possible to turn it only into the right direction for 60 degrees and back. The offset isn't visible from outside. When the number of degrees is reached it breaks the constraint.

To reach all this a extended version of a physical constraint actor is used. To get this actor into a plugin the description in the following source was used:

https://forums.unrealengine.com/development-discussion/c-gameplay-programming/50761-adding-an-actor-class-to-plugin

The last posts of Mhousse1247 helped to get it into a plugin.

To get the normal plugin folder structure of unreal, public for the header file and private for the cpp file, it's just necessary to replace the line "PrivateIncludePaths.Add("TestPlugin/Private");" with "PrivateIncludePaths.Add("TestPlugin/Public");". Also it was of course necessary to exchange "TestPlugin" with "UCorkScrew" which is the name of the plugin.

Author: André Osse

Milestone 2

Milestone 2 Description

- Parameterize
    - Angle (standard value = 60°)
    - Turning way (standard value = right)
    - Height of threaded area
    - Number of turns to open
    - Approximate lead as ratio height/no of turns
- Make lid travel linearly up/down on the threaded axis while rotating it

Work of Milestone 2

Problem description for turning

Since it has negative values for a left turned swing and positive values for a more right turned swing, though it changes the sign from positive to negative after passing 180 by the right or 0 by the left and other way around, there was a way necessary to calculate the real turned value.

To make sure it isn't possible turning to the other side, than it should be possible, there also must be a way to limit the turning for one way as in milestone 1, but set it free when its turning in the right direction.

Solution for turning

To get the lied turn more than one time there is a quite simple concept of calculating the real turned value. In order to do this, keeping the limit solution from milestone one and avoid problems with the negative values, the solution was to count every half turn. For the right direction the limit of 30 and offset of 30 can be kept. For the left direction the limit is the same but the offset must be on -30 degrees, so it can turn on the left 60 degrees.

On both versions the first thing to check is, in which direction the lid gets turned. If it gets turned on the right over the degree of 0, and hasn't passed any 180 yet, it should set the limit free. So the lid can be turned as much on the right as it wants. If it moves back it should stop if there has been no 180 rotation anymore and if it gets over 0 again it should turn the limit back on. Same on the left side just the other way around.

Figure 2 shows the spots the 180° counter is used for the right-turn-opening. The spots for the right-turning is as shown on the picture on -30° and 150°, so exactly 180° from each other. The left-turn-opening value is on 30° and -150°. The calculation part is just different for each turning way.

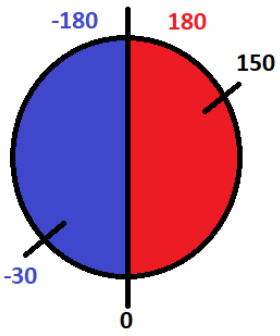

Figure 2

The concept on the left-opening-turn is the same as right, just the calculation must be different caused by the turning way and the spots. So just the right way will be explained here.

The calculation for the real degree number uses the counter of the 180° completed values, multiplied by 180° so the real turns are calculated and adds the current degree number, which are over the 180° spots. So if the lid turned 3 times a half circle around it has turned 540°. If it has 550°, the multiplying of the 180° turns aren't enough, it has to add 10° which are over the last 180° spot.

To get this value it's just necessary to watch how far it is turned yet and set up a calculation. For three different spots on the circle there must be 3 different calculations. The areas are shown in figure 3.

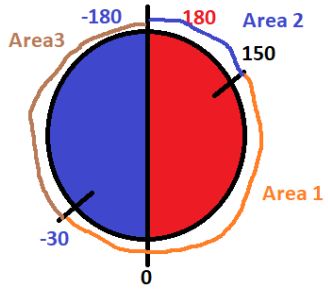

Figure 3

Each Area needs a own calculation, which takes care of the negative or positive degree value and also the different spots of 180° counters.

The simplest example calculation is for Area 1: First it needs to be checked if the current degree is inside this area, if it is just calculate the degree number + 30 degrees. The reason of the 30 degrees is by the spots and of course the offset, which is the reason for the spots.

The other turning way, so the left-opening-turn, has a familiar calculation from 30 degree to -150 degree. Since the offset is -30 degrees we need to subtract 30 of the current value. Than we need to change the value from negative to positive which is in the calculation simple made with an equivalent calculation:

```
// If the current swing is between -150 and 30.
else if (CurrentSwingDegree > -150 && CurrentSwingDegree < 30)
{
    CurrentSwingDegree = 30 - CurrentSwingDegree;
}
```

Figure 4

Since it subtract the value, which is negative if it's turned more than 30 degrees to the left, it gets the real degree over the spot of 30. If it hasn't turned more than 180°, the 180° counter is on 0. So it multiplies 180° by 0 and gets 0 to add to the CurrentSwingDegree value.

To edit the turn direction in the editor there is a simple Boolean. If the constraint is activated as bottle opening constraint at all, the Boolean named "RotateDirectionRight" makes the rotation to the right possible if it is true, else the direction is to the left.

The number of turns to open can be set by the "ScrewAngle" parameter, which is simply the number of degree it should break on. So if it should turn more than 1 time the full way around, the value just must be chosen over 360°, where ever it should break the constraint.

Author: André Osse

Problem of moving the lid up and down

Since the actual version works with offset and needs the lid to activate simulate physics but the bottle not to do it, it didn't work to set the lid higher. Though it might be a problem that there is no way founded yet to set the relative position from the lid to the constraint up, so it might get problems if the rotation is different and it uses the worlds usual rotation. For example if the rotation is slightly 45 degree so the bottle with the lid is diagonally, the worlds location Z coordinate just switches the lid up and down from the worlds ground, not from the rotation of the bottle, which could make really weird looks. There has been no solution found yet.

Author: André Osse

Tutorial

First set 2 mesh actors in the world directly under each other, like in figure 5, and activate the physical simulation at the top one.

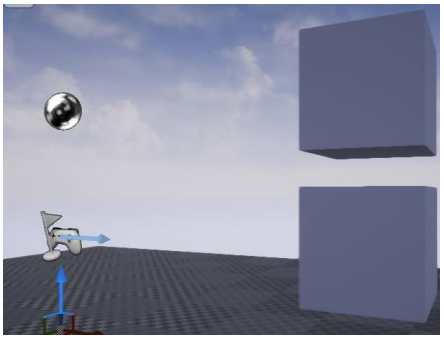

Figure 5

Go in the project folder on the unreal editor into “UCorkScrew C++ Classes” –> “UCorkScrew” -> “Public”, as you can see in Figure 6.

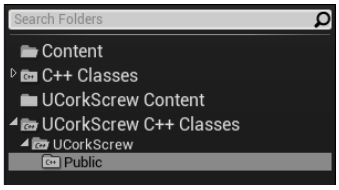

Figure 6

Now you can see the RealisticOpeningActor, which is basically the extended PhysicsConstraintActor with the new futures. You can now move it inside the world like a usual PhysicsConstraintActor. You can see it in Figure 7 and 8.

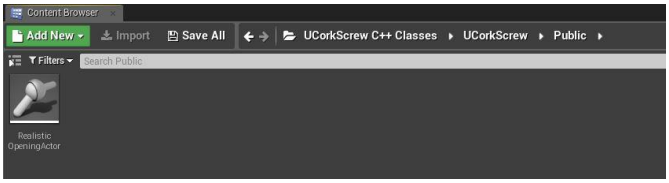

Figure 7

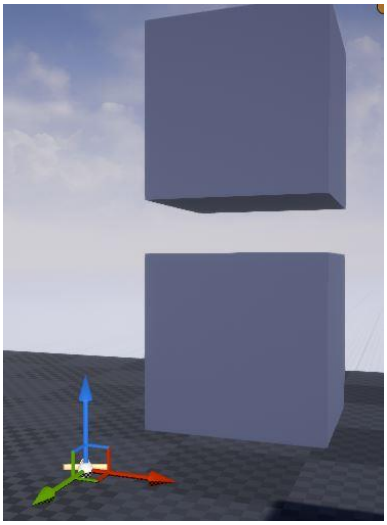

Figure 8

Author: André Osse

Now you can connect the lower one with the bottle and the lid with the top one just be clicking on the fields in the properties. It opens a drop-down menu showing all actors in the world. Click on the once you want to use as bottle, so the lower once. Figure 9 shows how it should look like.

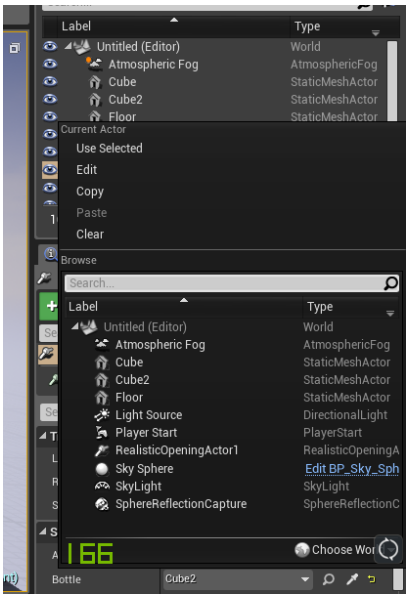

Figure 9

After connecting the Bottle, I moves the realistic constraint actor at the spot the bottle is located on. If u select another object and then the RealisticOpeningActor again you can see it is on the right position. You can also instandly see that the bottle cube gets a red wireframe. If you now add the lid as well it gets blue and a small blue wireframe between those two. Shown in Figure 10.

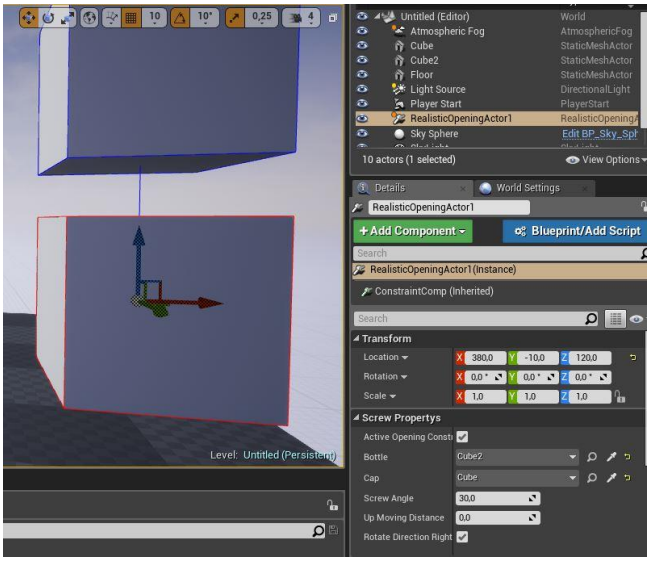

Figure 10

Now you can just change the other properties of the constraint, like the ScrewAngle in any value you want. Only blocked are lower values than 1 for this value. If you start playing now the lied won't fall, and if you fly against it, so it reaches the ScrewAngle after some time it will break the constraint and the lid will fall off.

Author: André Osse

Known Bugs

Only works if the lid as the physic simulation turned on. Still no working way for moving up and down. Doesn't take the same rotation as the bottle yet.

If the turned degrees are over 10° per tick it gets probably strange behavior. Reason why it's not going to be fixed is, that it shouldn't be possible to turn it more than 10° per tick for a human. Though depends on the PC used, but on PCs that should be ready for VR it shouldn't be possible.

Features

The constraint works visual as a usual physical constraint. Also it sets the position of the bottle automatically.

Future work

- Getting the up/down moving to work
    - Also if the rotation is different it should work
    - Calculated by the turned degrees
- Getting friction/dumping force
- Getting a good debug mode that shows actual values.

Author: André Osse