# GoverPod Notes

CBP 10-02-23

Unreal on the Lab Machines

Download the Unreal 4 Assets zip. Click on the zip and drag the folder onto the Desktop which will unzip. Open Unreal 4, and hit **More** then **Browse** and find the file **Myriapod.uproject** which you should open. A number of things may happen depending on which computer you are working on. Most likely you will be faced with this dialogue box, in which case select **More Options**.
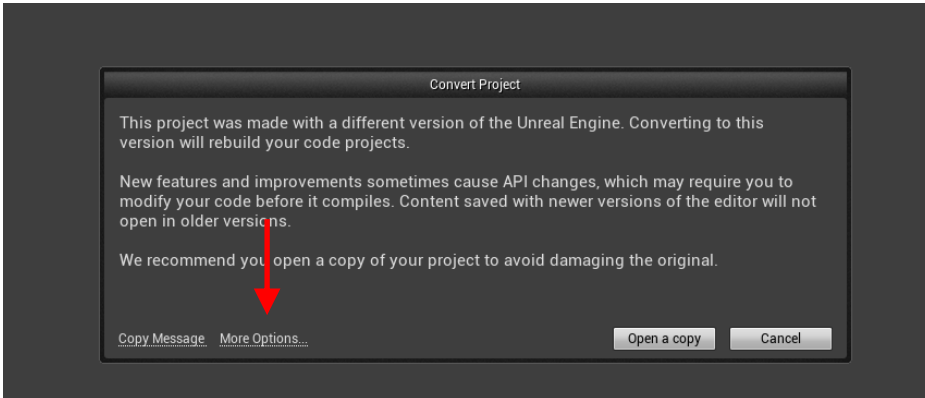
Now a second dialogue will pop up, and choose **Skip conversion**.
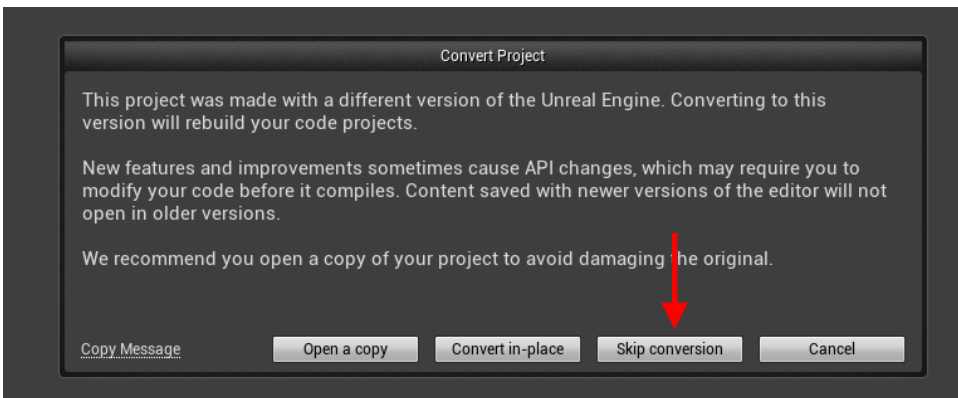
Hopefully the level will open

Some Background Information.

This is a hexapod programmed to be in its 'tripod' gait where LF, LR, RM have the same goal angle which is out of phase with RF, RR, and LM. It's named after Mike Gover, former student who created the components in Blender.

The Unreal physics engine is used for (i) the interaction of the feet with the ground and (ii) the constraining links between the Head, Tail and Middle segments.

The top-view below shows a moment during the tripod gait. Red cylinders show where the legs are connected to the body segments, they rotate about an axis through the cylinders. The green cylinder locates the feet which move up and down.
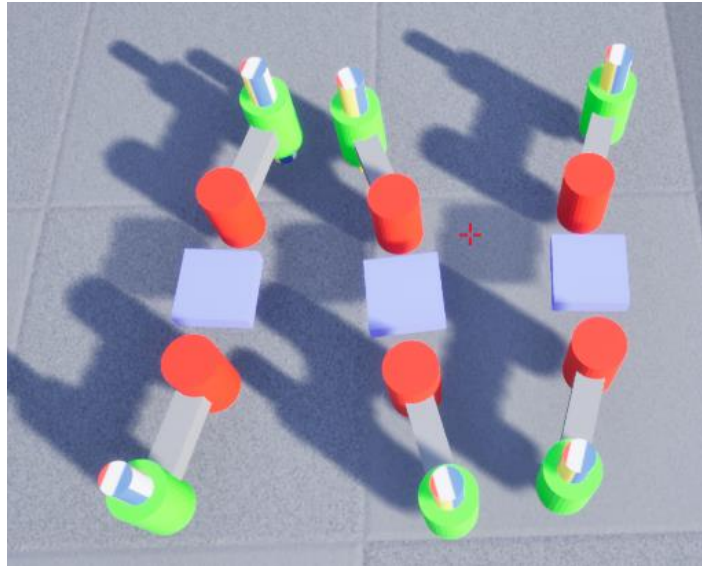
Below is a screen-shot of the feet at the same moment as the figure above. Inside the green cylinders you can see the multi-colored feet which move up and down.
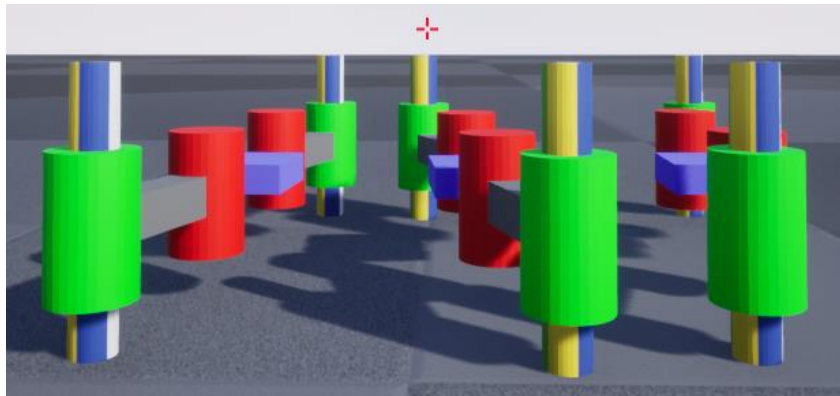
Legs and feet are moved through procedural (not neural) code. First two phases are computed, 180 degrees apart.

```
phase1 = sin(((2 * PI / drivePeriod) * time));
phase2 = sin((2 * PI / drivePeriod) * time + PI);
```

These are converted to lift and swing drives, for the feet and legs respectively.

```
lift1  = liftDriveAmp  * calcLift(phase1);
swing1 = swingDriveAmp * calcSwing(phase1);

lift2  = liftDriveAmp  * calcLift(phase2);
swing2 = swingDriveAmp * calcSwing(phase2);
```

These drives are sent as torques to the legs; this makes use of Unreal physics engine. Code for the left side is shown, you can see that the middle leg is driven 180 degrees out of phase from the front and back legs.

```cpp
LeftFrontLeg->AddTorqueInDegrees(FVector(0, 0, swing1), NAME_None, true);
RightFrontLeg->AddTorqueInDegrees(FVector(0, 0, swing1), NAME_None, true);

LeftMiddleLeg->AddTorqueInDegrees(FVector(0, 0, swing2), NAME_None, true);
RightMiddleLeg->AddTorqueInDegrees(FVector(0, 0, swing2), NAME_None, true);

LeftRearLeg->AddTorqueInDegrees(FVector(0, 0, swing1), NAME_None, true);
RightRearLeg->AddTorqueInDegrees(FVector(0, 0, swing1), NAME_None, true);
```

The values of swing are computed from the phase sine functions like this.

```cpp
double AMAS22_Hexapod_G::calcSwing(double phase) {
    double swing;
    if (phase > driveThreshold)
        swing = 1;
    else if (phase < -driveThreshold)
        swing = -1;
    else
        swing = 0;

    return swing;
}
```
Leg drive

```cpp
double AMAS22_Hexapod_G::calcLift(double phase) {
    double lift;
    if (phase > 0)
        lift = 1;
    else
        lift = -1;

    return lift;
}
```
Foot drive

The foot drive is simple, when the phase is > 0 then lift is 1 (up) else -1 (down). The leg motion is a little more clever, the leg swings forward when the phase is above a threshold, and swings back when it is below the -ve threshold. This is shown below.

The green dashed line shows the threshold has been set to 0.5.

The red dashed lines show how the sine produces the Leg Swing drive, using the threshold value.
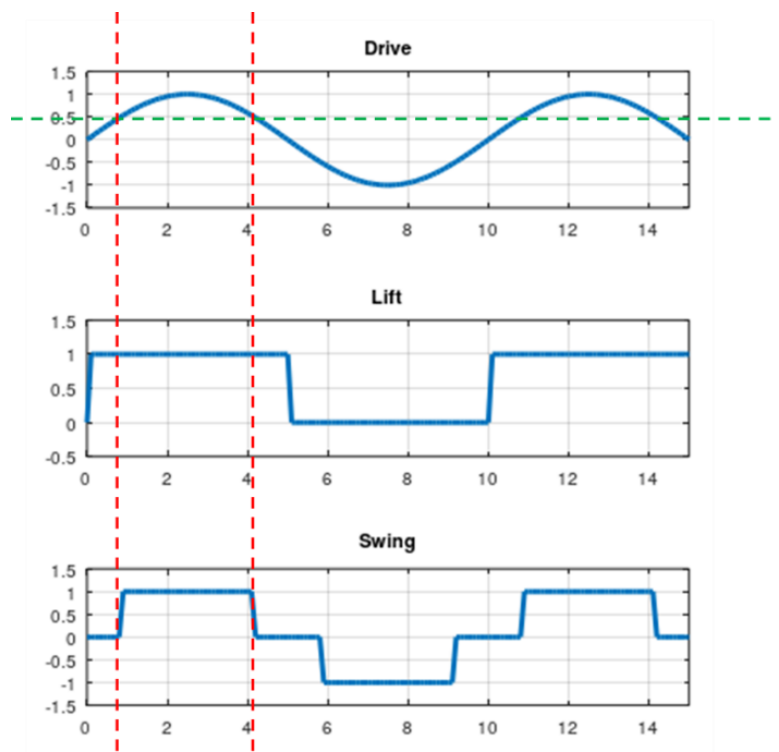
This model has a huge number of parameters, default values are shown below, grouped as indicated by colors.

Yellow shows the important parameters which control the motion of legs and feet. Green are concerned with the joints between the segment and blue are component masses. It's probably best to leave the blue ones alone.

◢ MAS22 Hexapod G		
Rotation Joint Stiffness	1.0	🟩
Drive Period	2.0	🟨
Lift Drive Amp	6000.0	
Swing Drive Amp	2000.0	
Drive Threshold	0.5	
Body Angle Limit	10.0	🟩
Leg Angle Limit	15.0	
Head Mass	0.5	🟦
Midl Mass	0.5	
Tail Mass	0.5	
Leg Mass	0.1	
Foot Damping	0.01	
Foot Mass	0.05	
Run Time	1000.0	
ZMotion Limit	10.0	🟩
Linear Joint Stiffness	1.0	
Articulation Type	yawOnly ▼	