**Tutorial 2 - Blender Game Engine Basics**

close blender... just close it... notice how it doesnt prompt you to save.

save often, save often, save often

=======

1. enter one of the side views

2. grab the default cube and move it up a bit and rotate it so it isnt parallel to all the axis (teach the r+x, r+z, r+y combos)

3. create a plane below the cube, scale it and bring it down a bit.

4. go into the camera's view

5. select the cube and then look closely down on the "panels" at the top of the buttons window.

6. click on the purple packman. This is the game logic panel

7. change the cube from static -> rigid body

8. adjust the form setting to something like 2

9. click on bounds and select "box" from the pulldown menu that appears

10. hit 'p' on the keyboard

=======

1. duplicate the cube a bunch of times (make sure the bounds don't overlap though)

2. hit p again

3. play with this. Use other shapes, use other bound types, edit the floor so it isn't flat.

=======

1. ok delete all your dynamic objects and make a flat floor if you messed it up.

2. create a new cube and make it sit just above the floor nice and straight

3. parent the camera to the cube and position it behind the cube.

4. go to the tab you used to make it a rigid body but this time make it just dynamic

5. set it to have box bounds

6. explain breifly about logic: sensors -> controllers -> actuators

7. add 4 sensors.

8. for each, click on the tab that says always and set them to keyboard instead (notice how they change their appearance and more options are available.

8.5. if you cant see all your sensors you can actually pan this window with mmb and even zoom in or out by holding down ctrl or pressing + / -.

9. for each sensor respectively click on the 'key' option and press w, a, s, or d

10. also rename each sensor to be "up", "down", "left" and "right" respectively (this is an important habit to get used to).

11. add 4 controllers and link each sensor key to its controller

12. add 4 actuators. Notice that the default is "motion" which is exactly what we want.

13. link the actuators to their controllers

14. apply motion in the different directions: the numbers represent x, y, and z IN THAT ORDER!

15. what we want is 'w' and 's' to move the cube forward and back and then have 'a' and 'd' rotate the cube left and right.

16. we do this by inputing values into the 'Force' and 'Rot' parameters. rot values should be considerably lower than force values... [ force: 7.0, rot: 0.01 ]

17. test it out. if you fall off the edge you fall forever!

=======

1. ok this is nice but what about the camera? the camera really helps to define a game. Rather than simply parenting to the cube, let's make the camera track the movement of the cube so the view rotates more interestingly.

2. select your camera and alt-p to remove the cube as it's parent

3. now go to it's logic and create 1 of each logic type and connect them all together

4. for the sensor we will be using the "always" but we need to change its pulse type so press the "true pulse mode" option

5. for the actuator, change it's type to "edit object"

6. click on the pull down that currently says "add object" and change it to "track to"

7. now the options have changed a bit. We just need to input the "Cube" object that we want the camera to track. (brief explanation about names)

8. hit 'p' and see what happens.

9. where'd everything go??? well what its doing is it is only tracking on its z-axis so the camera jumps to looking straight forward... we can tell if we add something away in the distance you can see that it is tracking.

10. to fix this we will go back to the camera's logic and click on the "3d" option in the 'track to' actuator.

=======

1. before I let you go off and do your own thing for a while, let's look at a few other sensors/actuator combos and what they can be used for:

cheap way to make your game look cool: 2d filters

note that motion blur filter is not per-object so if the camera moves...

++++++++++

the "add object" actuator:

is probably the most useful out of all of them so let's make our guy shoot stuff:

1. create an object called an "empty" and put it infront of the cube.

2. parent the empty to the cube

3. create a line of logic

4. set the sensor to be 'spacebar' on the keyboard

5. set the actuator to be 'edit object' -> 'add object'

6. now go to a new layer and create a sphere with low divisions and scale it down a bit.

7. make the sphere a rigid body (default bounds are sphere so you don't need to set bounds)

8. adjust its "radius" setting to match its size

9. rename the sphere "bullet"

10. go back to the layer with your empty and type in bullet where the actuator asks for an object.

11. for 'Time' set it to 400 or something

12. now we got to give it a linear velocity. Choose the axis that faces forwards and give it a velocity of around 20.0 and hit 'L' for local so that it will change as the guy rotates.

13. hit 'p' and start shooting with spacebar.

14. if you want to be CRAAAZY then change the pulse mode to true and then hold spacebar.

++++++++++

how about the collision sensor?

this can be used to register what happens during a collision. Let's make him win when he collides with a "goal object":

1. create a new cube and set it as a rigid body with box bounds

2. give it a new line of connected logic

3. for the sensor set it to collision

4. we need to give it a "property" for what object it should be deeming as important to collide with

5. go to our hero cube and add a property here and call that "hero"

6. now go back to the level cube and type hero in the property field.

8. for the actuator go to "scene" -> "add scene" but here too we need to make something

9. go to the top of the blender interface and click on the current scene and hit "add new" -> empty. Name it "win"

10. now create a new camera and create a "text" object aswell

11. then type in "you win" or something

12. convert the text to a mesh by going "alt + c" and selecting mesh

13. go back to the previous scene and enter "win" in the parameter

14. set to camera view and hit "p". see what happens when you collide with the box!

and to make it even more awesome lets make the bullets explode when they hit shit:

1. get rid of that true pulse mode crazyness if you were doing that

2. go back to the layer with the bullet and create a big sphere

3. name the sphere "boom"

4. switch to the bullet's logic, add a new line and connect them but this time add 2 actuators.

5. set the sensor to collision

6. set actuator 1 to be "edit object" (default is "add object") and type in boom and give it a time of 50

7. set actuator 2 to be "edit object" -> "end object"

8. go back to your main layer, enter camera view and hit 'p'

=======

1. everthing else is just learning sensors, properties, and actuators. lets just play around with them for a while.

Feel free to ask me questions.

don't use the mouse sensor right now it requires some python to get working.