

What is Blender, what can I do with it, and why use it?

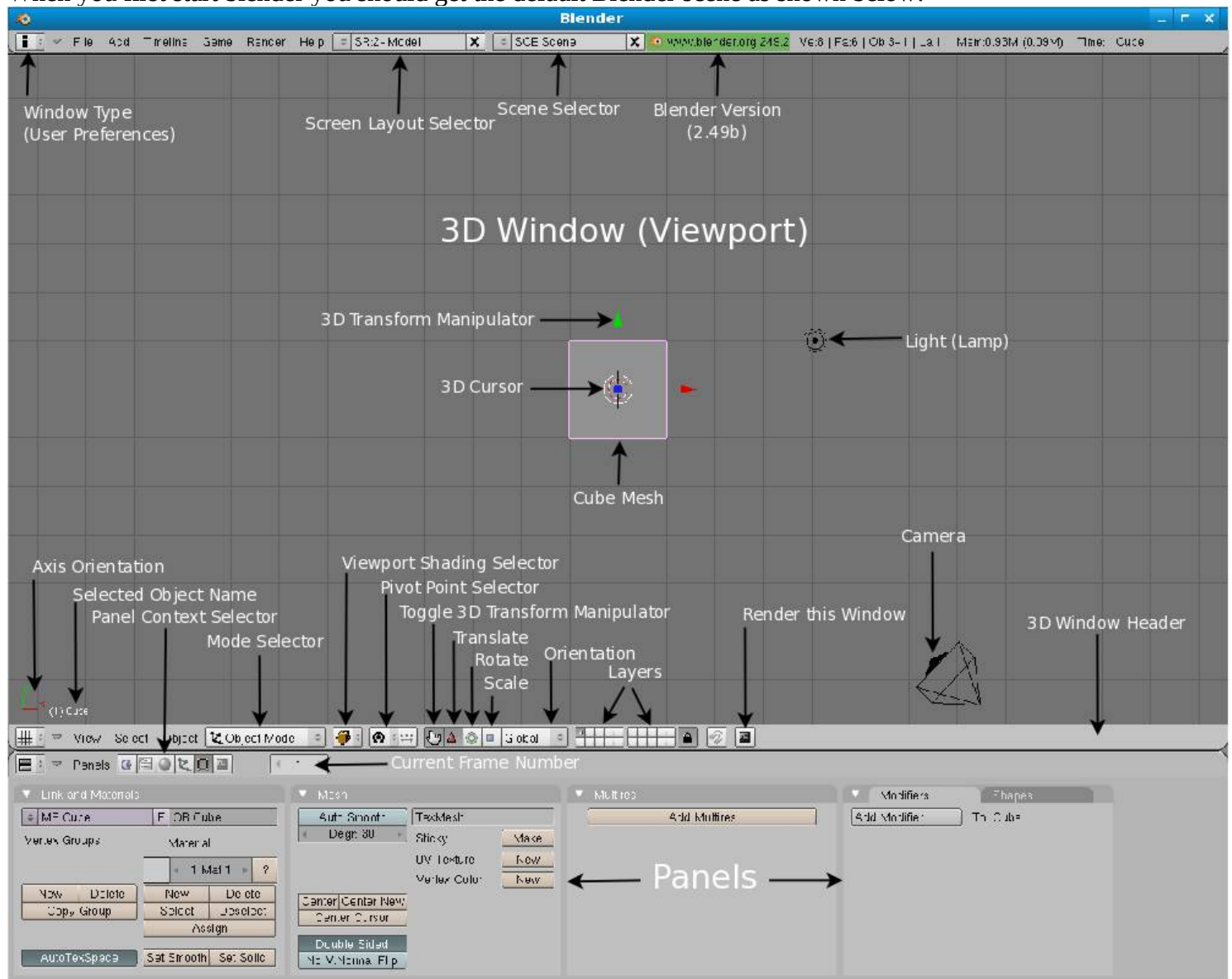
Blender is a free open source 3D content creation suite available at <http://www.blender.org>. Blender includes tools for 3D modeling, rendering, compositing, animation, simulation, and an integrated game engine.

That's fine, but **what can I do with it?** Blender is a complex piece of software and does take some time to learn how to use it. Once you have some basic understanding of the interface and working in 3D, what you do with blender depends very much on what you want to do and create. Blender can be used to create 3D models, make animated movies, special effects in videos, create games, render images to create logos, and much more.

Why use blender? Blender has the same features and many commercial products such as Maya, 3D Studio Max, Lightwave, and Cinema4D but without the costs associated with those packages. Blender is free, it is extensible using the python programming language and above all, blender has a very active and supportive community which provides tutorials, videos, PDF magazines, manuals and more. Blender also runs on many operating systems including Windows, Mac OS X, Linux, Solaris, and Irix. Blender also runs on very minimal hardware.

Blenders Interface (Default Scene)

When you first start blender you should get the default Blender scene as shown below:



The top menu (File, Add, Timeline, etc) is actually the header part of the User Preferences Window. You can put your mouse right under the menu bar to get a double-arrow, right-click and drag down to move the menu and display the User Preferences panel shown below. Once you have set your preferences you can right-click and drag up to hide the panel again.



One thing that I like turn on is the “View Name” option. This displays the orientation (Top Ortho, Front Persp, Left Persp, etc.) of the 3D viewport in the top left corner of the viewport. When you have multiple viewports this helps to know from which direction you are viewing your objects. Other things that I might set would be the to “Emulate Numpad” in the “System & OpenGL” settings if I am using a laptop that does not have a separate number pad.

The 3D Window also called a 3D viewport contains the default cube mesh, a light source (of type lamp), a camera, and the 3D cursor. When an object is selected you will also see the 3D Transform Manipulator. This allows you to perform translations, rotations, and scaling operations on the selected object by clicking on the handles of the 3D transform manipulator. You can change the orientation of the viewport with the numbers on the keypad. By default the viewport starts in Top Orthographic view.

The Buttons Window contains button panels depending on which context and sub-context has been selected from the Panel Context Selector. The main contexts include the following, the descriptions are not all inclusive but just examples of what you might find in each context:

- **Logic (F4)** – changes to the Logic context and includes logic bricks for use with the game engine
- **Script** – changes the panel to the script context
- **Shading (F5)** – changes to one of the Shading sub context context where you can set properties for lamps, materials, textures (F6), radiosity, and world (F8).
- **Object (F7)** – changes to Object context for object properties such and the objects name
- **Editing (F9)** – changes to the Editing context for applying modifiers, adding Materials to objects, etc.
- **Scene (10)** - changes to the Scene context where you set properties for rendering your still images or animations.

Modifying the Windows is made possible with blender non-overlapping OpenGL interface. You can put your mouse over the dividers between windows and right-click to open a context menu that allows you to split a window, join or merge windows back together or remove the header panel for a window. You can also left-click on the divider and drag the mouse to resize a window. Splitting the main 3D window into a number of smaller 3D windows each with a different orientation helps when modeling so that you can see what is happening to your model from various views. By right-clicking on the header of a window you get a context menu to allow changing the position of the header to be at the top, bottom, or to not have a header on a window. You are also able to manipulate the layout of the button panels by right-clicking in the panels window you can change the panel alignment to vertical, horizontal, or free.

Getting Started

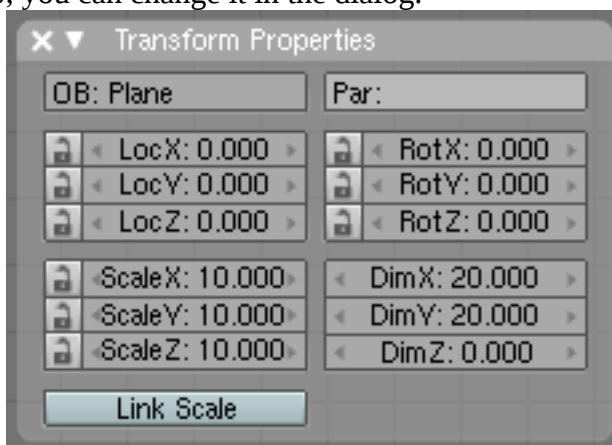
The first thing that you need to know about navigating blender in the 3D viewport is that you use the **Left Mouse Button (LMB) to place the 3D cursor** and the **Right Mouse Button (RMB) for selecting things**. The Middle Mouse Button (MMB) and scroll wheel are used to zoom, pan, and dolly around the 3D viewport. You should also

be aware that blenders interface is created with the idea that you will work with one hand on the keyboard and one hand on the mouse. This means that most of blenders features can be accessed with keyboard shortcuts. This makes it possible to work very quickly once you know the basics.

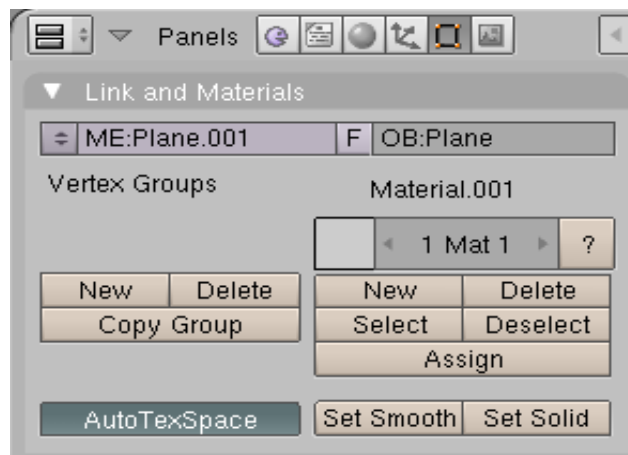
Lets Do Something!

Part 1 - Environment: We need some environment for our game, so lets start by making some simple ground and maybe wall it in so it could be a room.

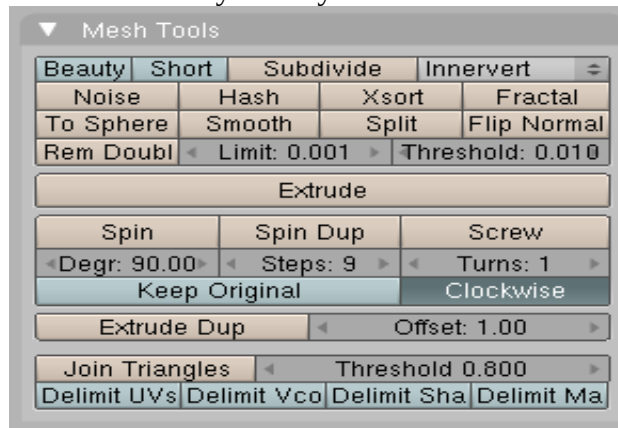
1. Make sure that your mouse cursor is in the 3D Window. We won't be using the default cube mesh so lets delete it. **Right-click** on the default **cube** and press the '**X**' key followed by the '**Enter**' key or by left-clicking on "Erase Selected Object's" to delete the default cube.
2. Press the '**Spacebar**' and select **Add->Mesh->Plane** to add a plane to our scene. This will be the ground for our game.
3. With the Plane selected (Right-click on the plane), press the '**S**' key followed by **10** and '**Enter**'. This will scale our plane by a factor of 10. The plane started as a 2x2 plane and should now be 20x20. You can verify this by pressing the '**N**' key to bring up the Transform Properties dialog. If your plane does not show DimX and DimY as 20.00, you can change it in the dialog.



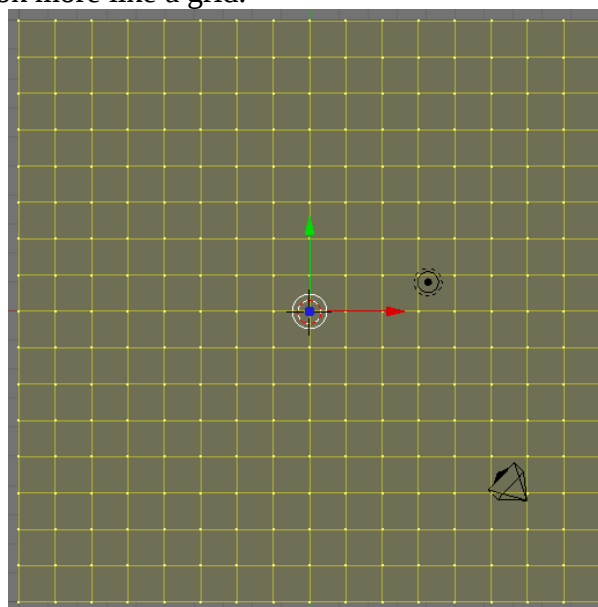
4. After scaling the plane it might be larger than your viewport window. You can center it back into view by pressing '**SHIFT**' + '**C**' or by using your scroll wheel on your mouse to zoom out. Pressing '**C**' will center the viewport on the 3D cursor and using '**SHIFT**' + '**C**' moves the 3D cursor to the origin and centers the viewport.
5. Now lets give our Plane some color. With the Plane selected (Right-Click on the Plane) press '**F9**' to make sure that we have the Editing context. Click the "**New**" button underneath the box with "**0 Mat 0**" in the "**Link and Materials**" panel. This will change the "0 Mat 0" to "1 Mat 1". This creates a new material data block and assigns it to selected Object.



6. **Left-Click** on the grey square to the left of “1 Mat 1”, this will bring up a color chooser. Pick a nice color for your ground, select the color you want and press 'Escape' or move your mouse outside of the color chooser.
7. Since we are calling this our ground maybe we should give it a better name. In the same “**Link and Materials**” panel click your mouse in the box that says “**OB:Plane**” and type “**Ground**” and hit 'Enter'.
8. Our plane is pretty simple right now so lets add some geometry. Press 'Tab' to go into “**Edit Mode**”. You will notice that you have a lot more options now in the 'Editing' panel.
9. Press the 'A' key to select all the vertices of our Ground plane. Press the 'Subdivide' button in the “**Mesh Tools**” panel 4 times, if you click too many times you can undo an action using 'CTRL' + 'Z'.



10. Your plane should now look more like a grid.

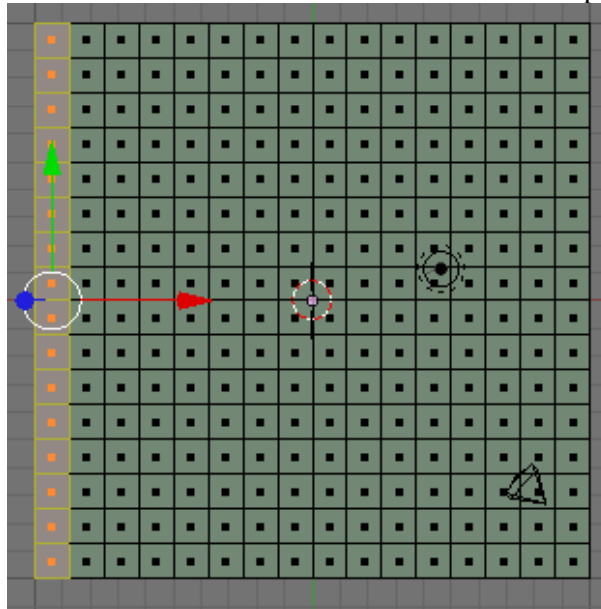


11. Press 'A' to de-select all selected vertices and then press the “**Face select mode**” button or 'CTRL' +

'Tab', '3' to go into Face selection mode.



12. We are going to add a wall around our ground to contain whatever we decide to have inside. In “**Face select mode**” and your mouse in the 3D viewport, press the 'B' key (border select) and holding down the Left-Mouse-Button, draw a box around the first column of faces on the plane.



13. Using the 'B' border select 3 more times, select 3 other side of faces in the perimeter of our plane. You can also individually select faces and add them to the current selection of faces by holding 'SHIFT' while clicking on a face.

14. With the perimeter faces selected lets change our view to the front orientation by pressing '1' on the number pad.

Number Pad Key	3D View
1	Front
2	Rotate Down
3	Right
4	Rotate Left
5	Toggle Orthographic or Perspective
6	Rotate Right
7	Top
0	Camera Perspective
.	User Perspective
Ctrl-1	Back
Ctrl-3	Left
Ctrl-7	Bottom
Shift-2	Pan Down
Shift-4	Pan Left
Shift-6	Pan Right
Shift-8	Pan Up

+	Zoom In
-	Zoom Out

- Now lets make a wall. Press 'E' , select **“Region”**, press '2' and then 'Enter' to extrude the faces as a region 2 blender unit along the surface normal (in our case along the Z axis). Extruding as a “Region” will not add faces between the faces where they are adjacent to each other. Extruding as “Individual Faces” would add a face in between each adjacent face. This might not be apparent at first but can be seen when viewing the model in wireframe mode and rotating the viewport around a little with the middle mouse button held down.
- Press 'Tab' to get out of edit mode and press '7' on the number pad to go back to top view.
- Press 'F5' to go into the **“Shading”** context and the **“Materials”** sub-context panel.
- Locate the “DYN” button in the “Material” panel and select it, you should now see a value and slider named “Restitut”. Set “Restitute” to '0.80', if you click on the value you can type the numbers in instead of using the slider . This is the factor of energy that the material will give back to an object that collides into it.



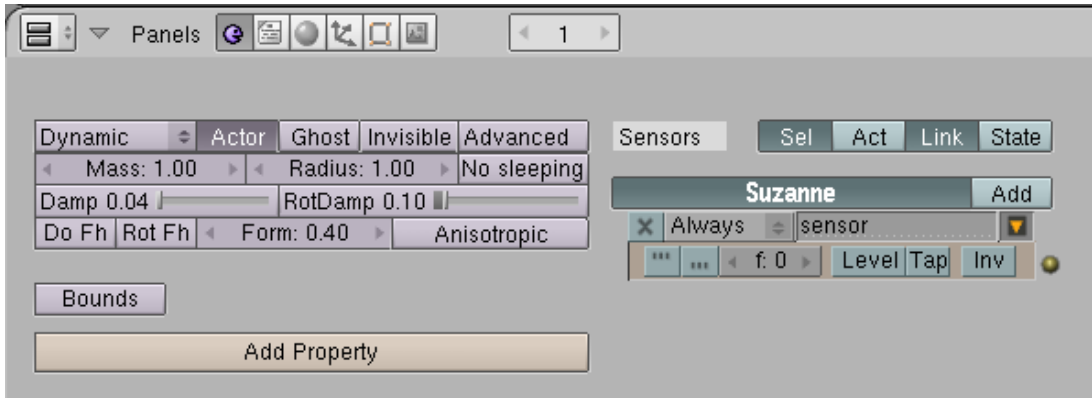
- This would be a good place to save our work. Press 'F2' to open the **“Save As”** dialog (or use the File menu on the “User Preferences” header). Navigate to a directory where you can save files and replace the **“untitled.blend”** with **“tecs_part1.blend”**, Press **“Enter”** and then click on the **“Save File”** button.

Part 2 - Actor: Lets add a simple character to our scene that we can try to add controls to so that we can do something.

- Now lets add an actor to our scene. Make sure that your mouse is in the 3D viewport (top Ortho), that you are in “Object Mode”, and that your 3D cursor is at the origin ('SHIFT'+ 'C').
- Press 'Spacebar' and select **Add->Mesh->Monkey** (This is Suzanne!)
- If you rotate around (holding **MMB** and drag mouse) in the viewport you'll notice that Suzanne is not really oriented right (she's facing up) so lets do the following. Press '7' on the number pad to get back to Top view and with Suzanne selected (Right-click on Suzanne if she is not selected). Press 'R' (rotate), 'X' (constrain rotation to the X-Axis), '90' (rotate 90 degrees), press 'Enter'.
- That's a little better but Suzanne is being intersected by the Ground. Lets move Suzanne up a little. With Suzanne selected, press 'G' (grab/move), 'Z' (constrain movement along the Z-Axis), '2', and press 'Enter' to move Suzanne up 2 blender units.
- Lets give Suzanne some color, Press 'F9' (Editing) panel and press **“New”** below **“0 Mat 0”** in the **“Link and Materials”**, Press the grey square to the left of the **“0 Mat 0”** and select a color from the color chooser.
- While we are here lets make Suzanne a little less blocky, press the **“Set Smooth”** button in the **“Link and Materials”** panel.
- While we are at it lets Press 'F5' to go to the **“Shading”** Context and **“Materials”** sub-context and activate the **“DYN”** and set **“Restitut”** to **‘0.80’** like we did above for the ground plane. Since our ground plane and our actor have restitute set, when Suzanne collides with the ground (or our walls) she should bounce

back!

8. Now lets make Suzanne move. With Suzanne selected, press 'F4' to go to the Logic panel.
9. The top left selector in the Logic panel is currently set to “Static”, change this to “Dynamic”. Lets see what that did. With the mouse in the 3D viewport, press '0' on the number pad to change to the Camera view and press 'P' to activate the game engine.
10. Press 'Escape' to stop the game engine, did anything happen? Suzanne should have fallen down to the ground plane and bounced a little. That's not too exciting so lets keep going.
11. In the Logic panel ('F4') Click the “Add” button in the “Sensors” area. This will create a Logic Brick Sensor.



12. In the drop down that says “Always”, change it to “Keyboard”, this creates a Keyboard Sensor. Click in the box to the right of the “Key” label, it will be replaced with “Press a key”, Press the “Spacebar”. To the right of the “Keyboard” drop-down, click on the sensor box and type in “Jump” and press “Enter” to give our Sensor a meaningful name.
13. Now Click the “Add” button to add a Logic brick for a Controller and then Click Add to add and Actuator Logic Brick. For the Actuator, type “20” in for the Force row in the center column and press 'Enter'. The numbers in the columns stand for the forces in the X, Y, and Z-Axis direction.
14. Connect the Sensor to the Controller by clicking on the little yellow circle to the right of the Sensor and dragging it to the yellow circle on the left of the Controller. Do the same between the Controller and the Actuator so that it looks like the following.

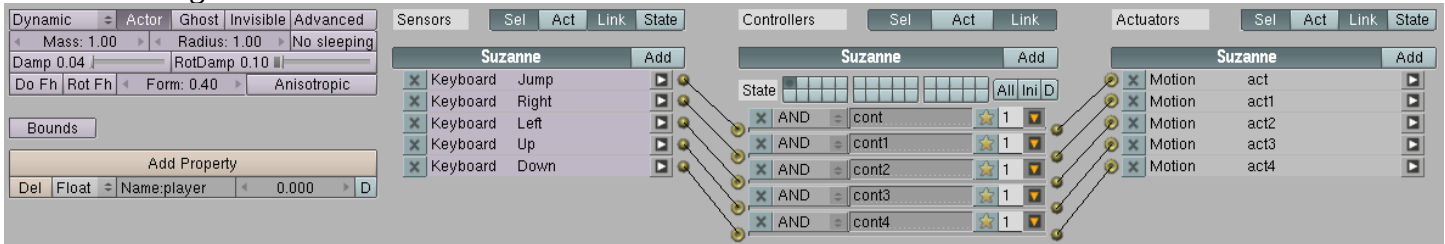


15. Now if we Press 'P' to start the game engine, and press the 'spacebar' once Suzanne stop bouncing, Suzanne will jump up a little, if you hold it down Suzanne will pretty much levitate up until you let up on the spacebar. Press 'Escape' to stop the game engine.
16. Using the same process create 4 more Sensors, Controllers, and Actuators and set them as follows. Since the Logic bricks can get pretty complicated it is a good idea to give each a name, you can do that in the text box next to the selectors for the Sensor, Controller, and Actuator types. You can also click on the little orange triangle at the top right of each Logic Brick to roll it up to save space. If you can't see the entire panel you can Click your Middle Mouse button in the panel and drag it around get a better view.:

Sensor Type	Sensor Key	Sensor Name	Controller	Actuator			
Keyboard	Rightarrow	Right	AND	LinV	10.00	0.00	0.00

Keyboard	Leftarrow	Left	AND	LinV	-10.00	0.00	0.00
Keyboard	Uparrow	Up	AND	LinV	0.00	10.00	0.00
Keyboard	Downarrow	Down	AND	LinV	0.00	-10.00	0.00

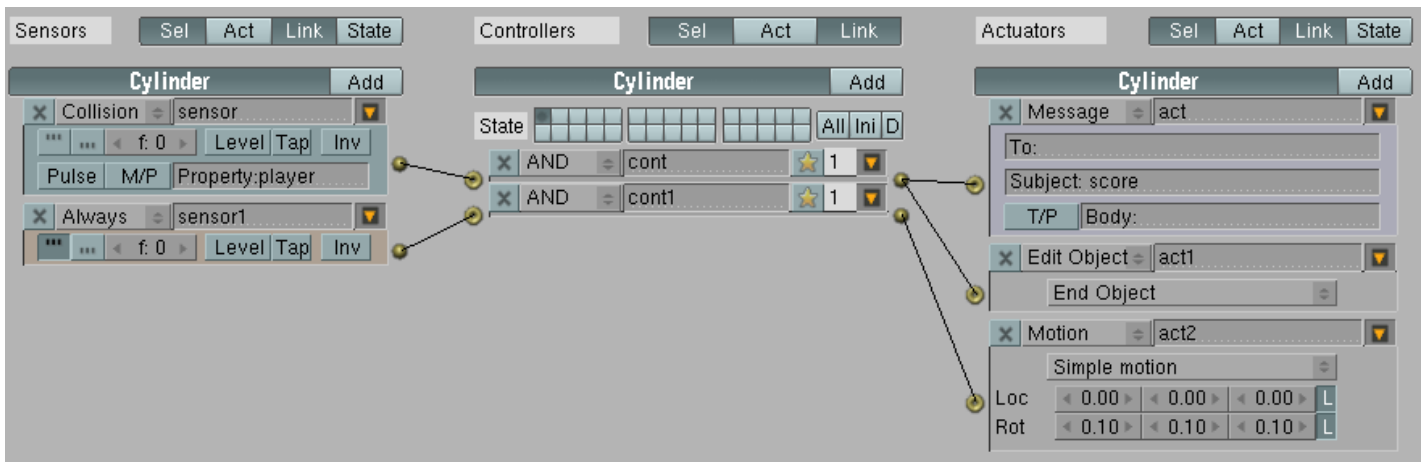
- Now if you press 'P', then you should be able to make Suzanne go right, left, backward, forward, and jump! If you make Suzanne jump over a wall she will just fall forever so just press 'Escape'.
- Now that Suzanne can move a little lets add a property to Suzanne called “**player**”. In the left hand Logic panel click the “**Add Property**” button and in the drop-down left of “Name:” change “**Float**” to “**Int**” and in the “**Name:**” box type in “**player**”. This is case sensitive so make sure it all lowercase. We will use it to identify our player when colliding with objects. So our Logic for Suzanne should look something like the following:



- Lets save our work again, Press 'F2' and this time just press the '+' key on the number pad. This should increment the number in the file name from “**tecs_part1.blend**” to “**tecs_part2.blend**”. You can still manually change it but this is a nice time saver.

Part 3 - Props: What if we are working in a team and we want to use something that someone else has created. Blender can import objects, meshes, even entire scenes from other blender files. So let's try it. We are going to append an object to our scene that so that we have something to interact with besides the ground and walls.

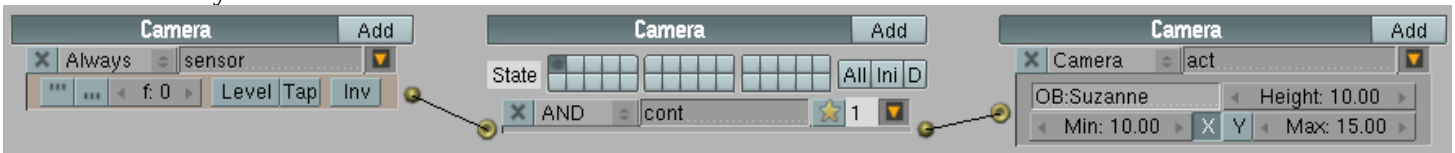
- In the “**User Preferences**” header, click “**File**”, “**Append or Link**”. Click the “**P**” (Parent), until you are all the way to /, then select “**lapps**”, “**tecs**”, “**gameTest.blend**”, “**Object**”, “**Coin**”, and then press “**Load Library**”.
- Did you notice something when you browsed for the .blend file? You where able to go right inside of it and pick the object that we are were looking for. If you look at your scene now you should see that an object has been added. If you press 'P' then you should notice a spinning “Coin” that if Suzanne bumps into it, it will disappear. When we used the “**Append or Link**” function the default is to have it append to your .blend file so once you save your .blend that object will be in there. If you choose to link it, then changes to the Object in the file you linked to would be reflected in your file.
- Lets make a few more by duplicating them. Make sure that your 3D viewport is in “**Object Mode**” and right-click on the Coin object. If you can't see it you might have to zoom out or pan around the viewport.
- With the Coin object selected, press “**ALT**” + “**D**” to create a linked duplicate of the object and put it into “Grab” mode so that you can place it where you want. Maybe constrain it to the X-Axis by typing 'X' and move it over with the mouse so that it's a little bit away from the other one. Create 2 more duplicates and place them like the other one, maybe by constraining by the Y-Axis this time. There are two ways to duplicate objects, “**ALT**” + “**D**” (Linked Duplicate), and “**SHIFT**” + “**D**” (Duplicate). A linked duplicate allows you to modify all of the duplicates by modifying one of them (this is true for edits to the mesh and materials but modifying the logic bricks of one will not modify them for others). A duplicate is an Independent instance and changes to the mesh will not effect other copies.
- Lets try it out. With your mouse in the 3D viewport press '0' on the number pad to activate the camera view and press 'P' to start the game engine. Move Suzanne around and try to collide with the Coins we placed. Press 'Escape' when done.
- Right-click on one of the Coins and take a look at the Logic (F4) bricks for them, can you see what makes them disappear on a collision? What about making them spin? Take note of the “**Message**” Actuator, we'll use this later.



7. Lets save our work again and name it **“tecs_part3.blend”**.

Part 4 - Camera: Wouldn't it be nice if our view of the game moved with us? Maybe like a camera man!

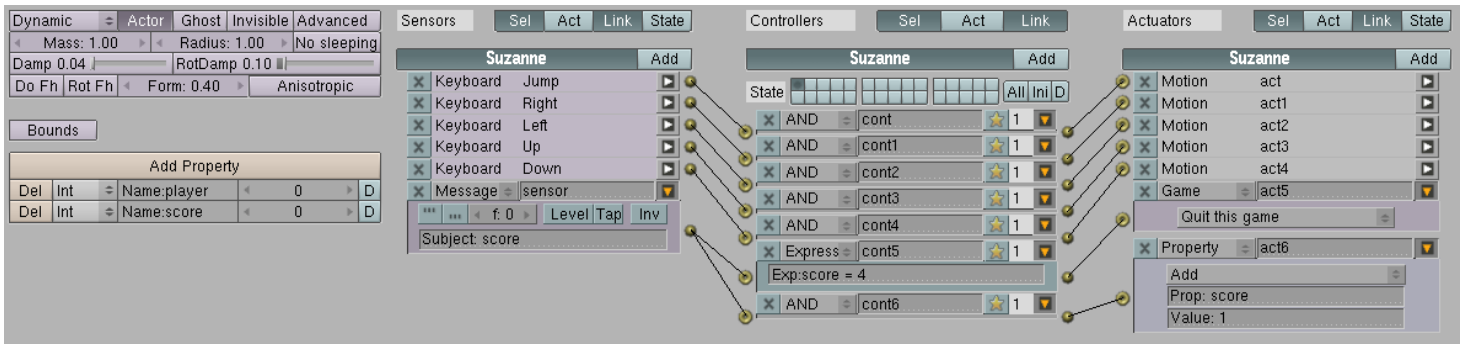
1. In the 3D viewport go to **“Top”** view ('7' on the number pad) and **Right-Click** on the Camera object (It kind of looks like a triangle or pyramid) to select it.
2. Press 'F4' to make sure you are in the Logic panel and add an **“Always”** Sensor type, an **“And”** Controller, and a **“Camera”** Actuator and link them together.
3. In the **“Camera”** Actuator, type **“Suzanne”** in the **“OB:”** box, set **“Height”** to '10', **“Min”** to '10' and **“Max”** to '15'. This tells the camera to follow Suzanne and to stay above our object and to maintain a certain distance from it. The **“X”** and **“Y”** toggle button specify which axis the X or Y axis the camera should stay behind.



4. With the mouse pointer in the 3D viewport, press '0' on the number pad to go into camera view and then press 'P' to start the game engine and see what happens, press **“Escape”** to stop the game engine.
5. Save your work and name it **“tecs_part4.blend”**.

Part 5 – Quitting the game: So far we have been exiting the game engine using the 'Escape' key. It would be nice if we could trigger that if a particular goal is reached. We will add a score property to keep track of the coins collected and require that a certain amount be collected before quitting the game.

1. Right-Click on Suzanne and go to the Logic Panel ('F4').
2. Add a property by Clicking the **“Add Property”** button and set Name: to **“score”** and change the type from **“Float”** to **“Int”**. This is going to hold the number of coins that Suzanne picks up.
3. Add a **“Message”** Sensor and in the subject type **“score”** and press 'Enter'. The actuator listens to **“Message”** Actuators with the subject of **“score”**.
4. Add an **“Expression”** Controller and for the Expression type **“score = 4”** and press 'Enter'.
5. Add a **“Game”** Actuator and from the drop-down selector choose **“Quit the game”** and link the Sensor, Controller and Actuator together. If we ran our game right now it would not yet quit if we collected 4 coins because there is nothing telling our score property to change values.
6. Add an **“AND”** Controller and a **“Property”** Actuator and link them to our **“Message”** sensor. For the **“Property”** Actuator choose **“Add”** from the drop-down selector and put **“score”** in for the **“Prop:”** and **“1”** in for the **“Value:”**. This will update the score property by adding 1 to it's value each time a Message is triggered with **“score”** as it subject.



7. You might be asking what is sending the “score” Message. If you look back at the Coin Object we appended to our game you will notice that the “Collision” sensor on the Coin is linked to a “Message” Actuator which has “score” as its “Subject”.
8. Now if we run start the game engine and collide with 4 coins the game will exit.
9. Lets save our work as “tecs_part5.blend”.

That's it for now: Thanks for you time, hopefully you learned something about Blender!

Additional Resources: The Blender community is very active and there are many tutorials, scripts, videos, and books about blender. The following are links to some that I have found very useful.

- <http://www.blender.org> – This is the main blender website where you can download blender and find the blender manual as well as links to the various blender resources and the blender e-Shop.
- <http://www.blendernation.com> – Main blender blog posting news about blender.
- <http://blenderart.org/> - A PDF magazine dedicated to Blender.
- <http://blenderartists.org> – Excellent forum, if you have a question about blender and how to do something or just want to get advice on your project this is the place to ask.
- <http://www.yofrankie.org/>- The Blender Foundation sponsored open game project.
- <http://durian.blender.org/> - The third Blender Foundation sponsored open movie project following the success of Big Buck Bunny and Elephants Dream.
- <http://www.cdschools.org/54223045235521/blank/browse.asp?A=383&BMDRN=2000&BCOB=0&C=55205> – James Chronister of Central Dauphin High School has created an excellence Blender Basics Classroom Tutorial Book in PDF format.
- <http://ocw.tufts.edu/Course/57> – Neal Hirsig of Tufts University has tutorials both text and video that cover 3D Design with Blender.
- <http://rab3d.com/index.html> – Robert Burk has an excellent Precision Modelling Guide PDF book that details using blender for CAD type work.
- <http://blender-archi.tuxfamily.org> – a wiki devoted to Blender for Architecture.
- <http://www.graphicall.org> – downloads of blender versions in development or testing builds with new features.