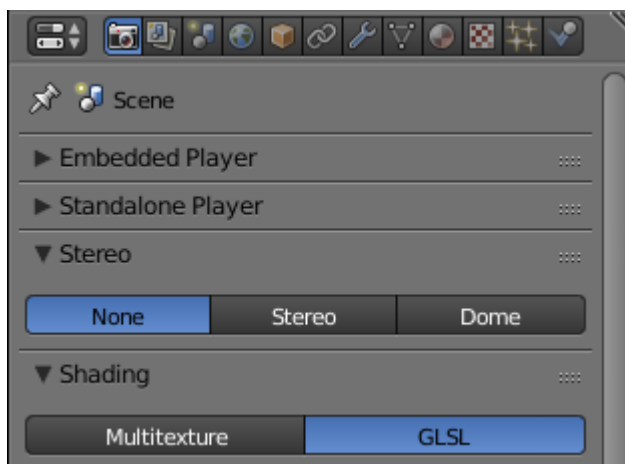
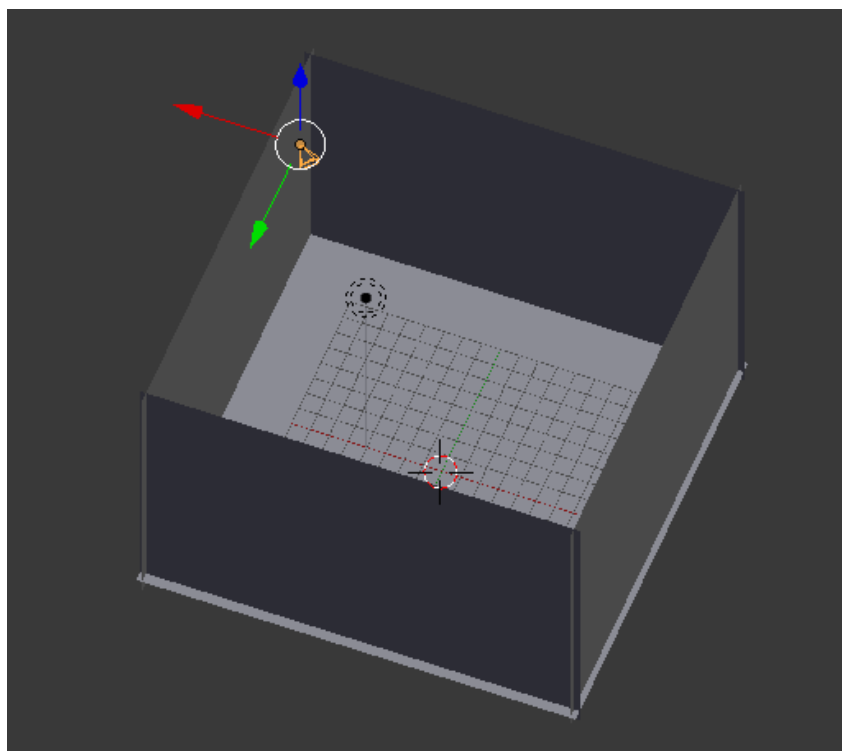


В этом уроке мы немного отвлечемся от игровой логики и Python. Займемся графикой и попутно рассмотрим режим GLSL.

Что нам дает GLSL? В этом режиме можно создавать реалистичные тени, анимированные текстуры и многое другое. Создав новый проект сразу включим GLSL:



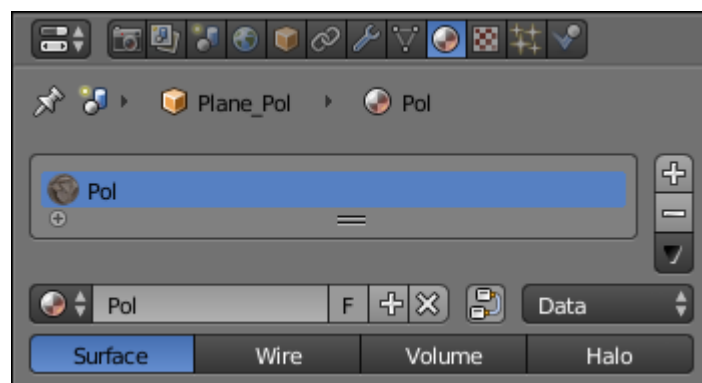
Создадим стены и пол нашей комнаты и выберем для них текстуру:

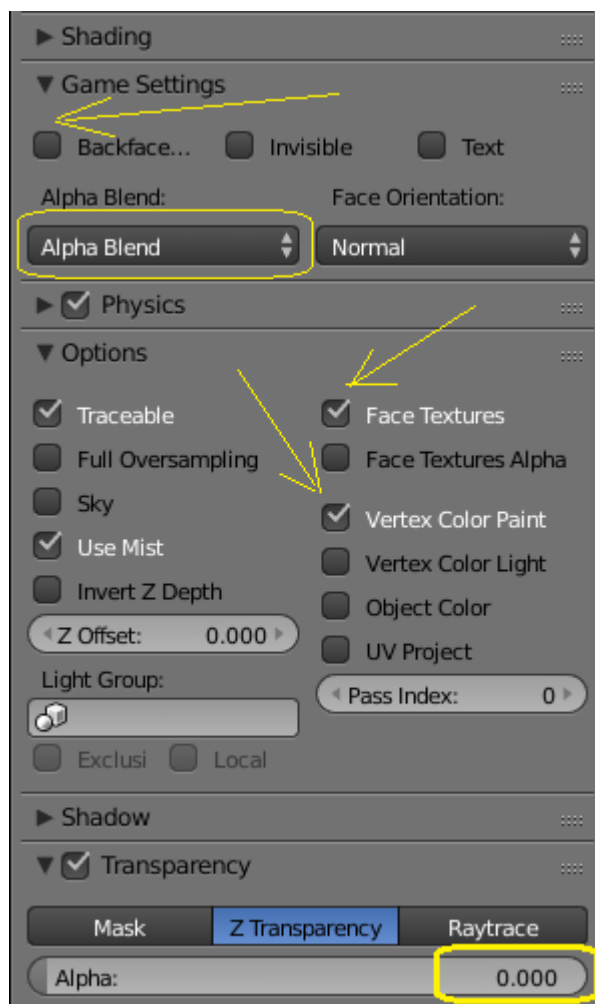


Я выбрал две вот таких:



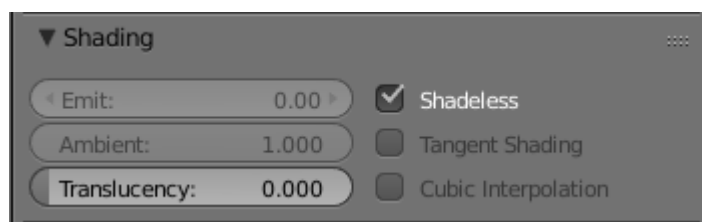
Только текстурировать в этот раз мы будем немного иначе. Выделим плоскость пола и создадим для неё материал с таким же названием:



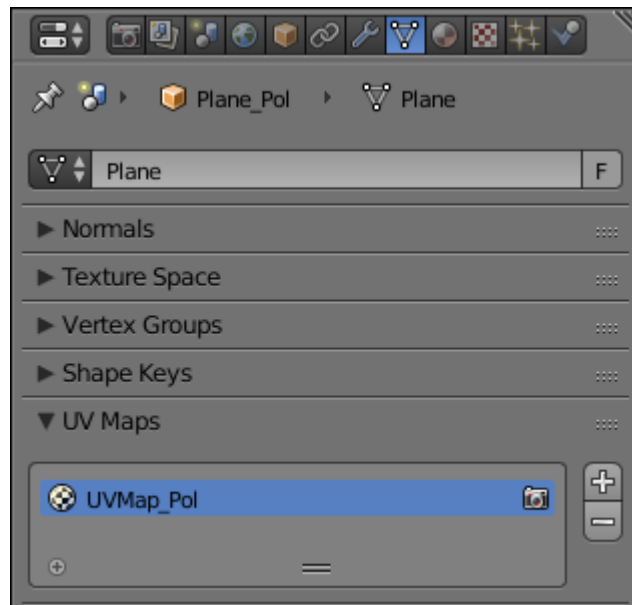


Если у вас хорошая видеокарта, то установив источник света Spot можно использовать красивые тени. Некоторые видеокарты (на старых компьютерах и некоторых ноутбуках, в особенности Intel,) не корректно работают с режимом GLSL. Наша задача добиться одинакового вида на любой машине, поэтому тени мы рассматривать не будем. Тем более, что это дополнительно нагружает процессор.

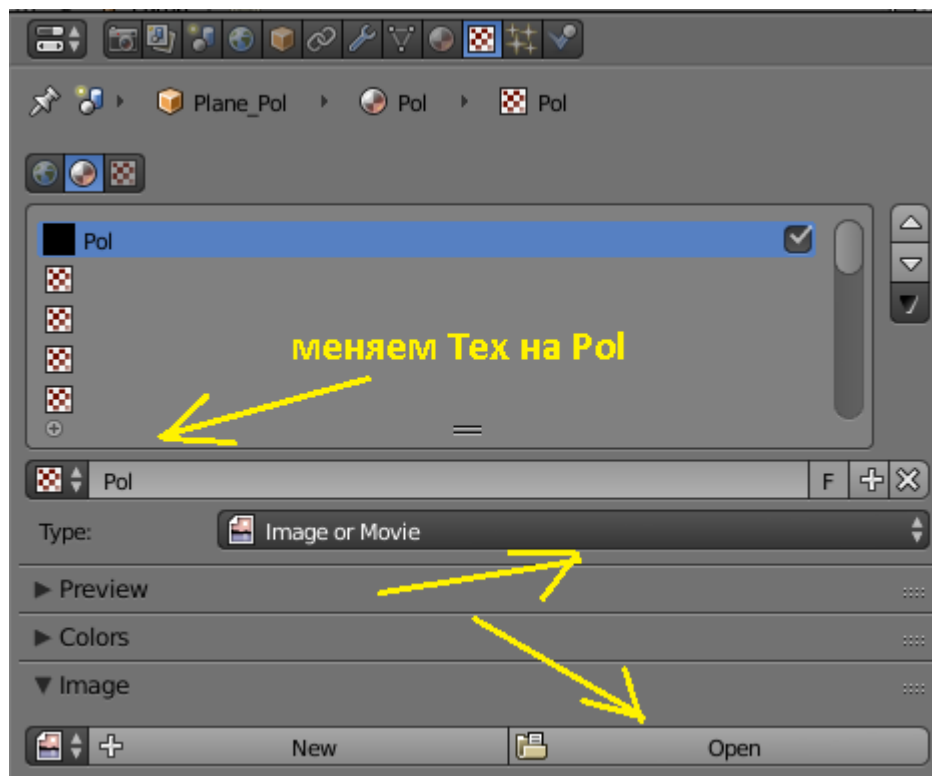
Обязательно включаем Shadeless:



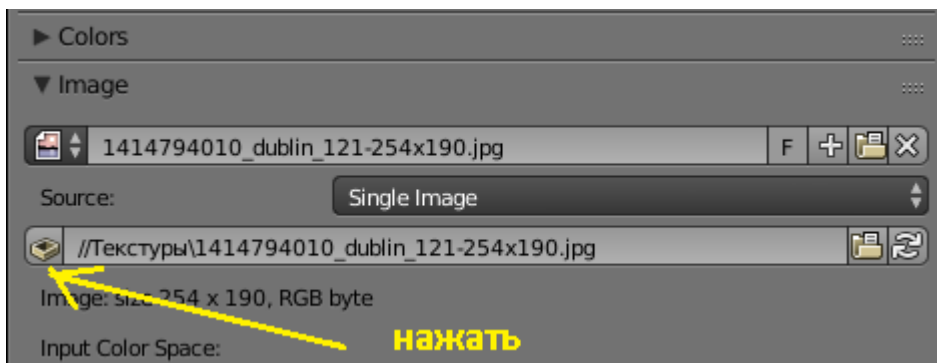
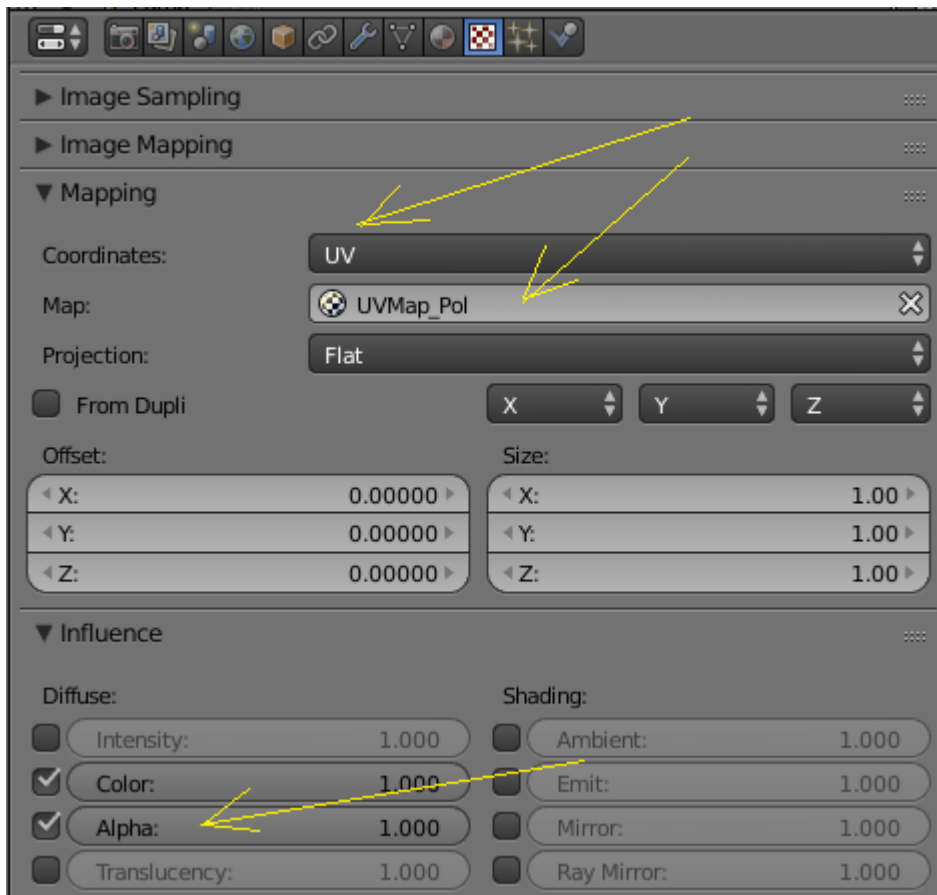
Перейдем на вкладку данных меша и создадим UV (нажав плюсик справа). Выделим название (двойным кликом) и добавим слово Pol :



Перейдем на вкладку текстур и загрузим текстуру для пола:



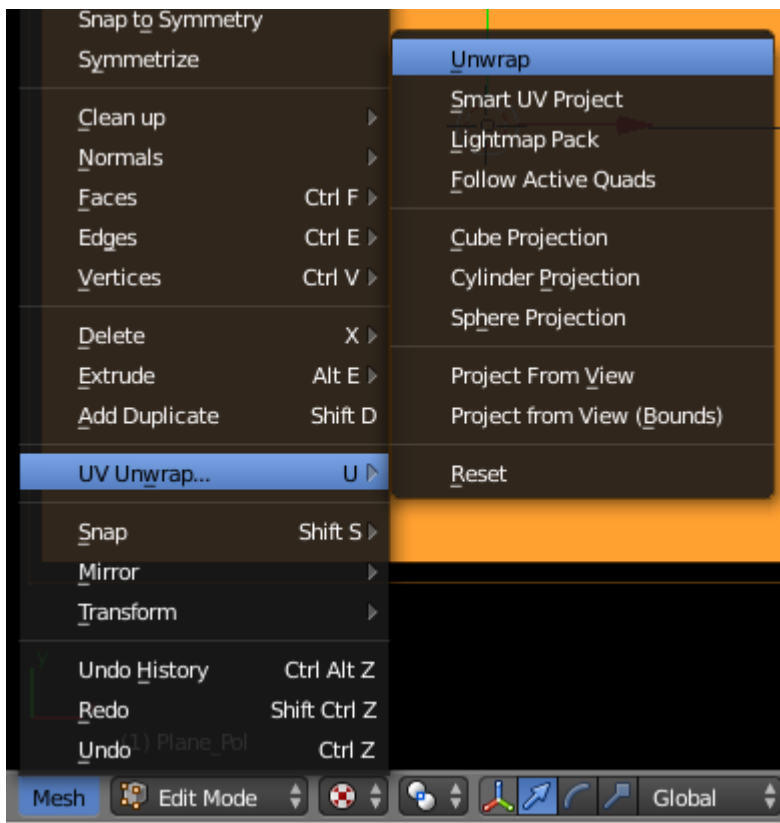
После загрузки картинки изменим настройки:



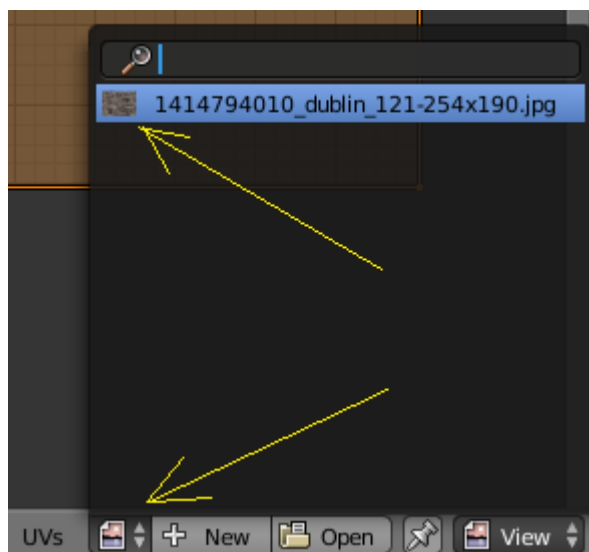
Включаем режим отображения текстур и режим редактирования:



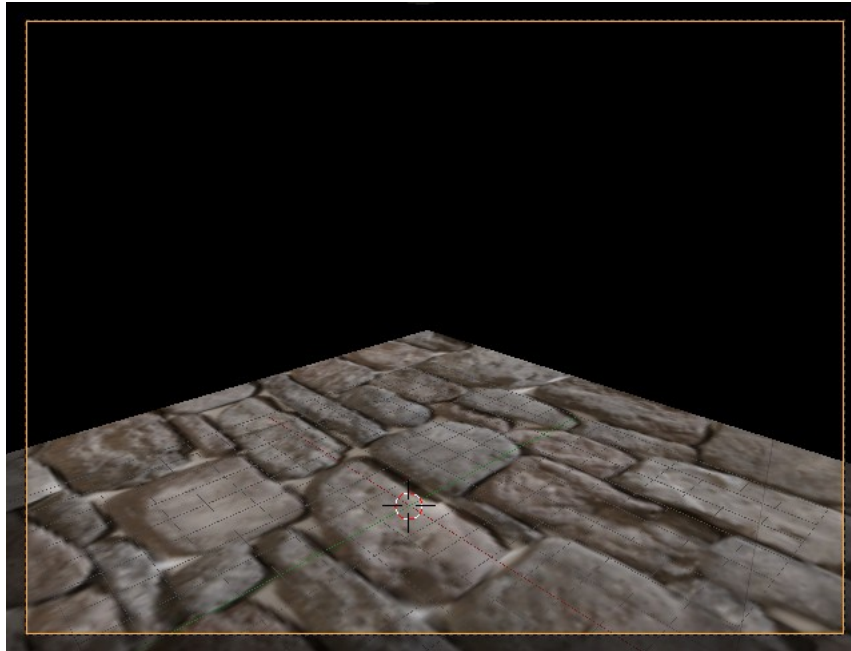
Переходим в редактор текстур (UV Editing). В котором справа делаем вид сверху и ждем :



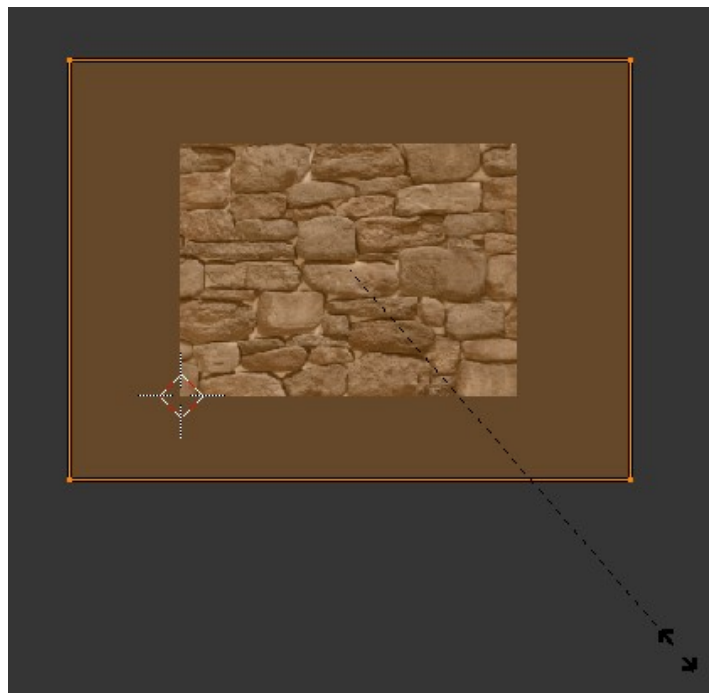
А слева выбираем нашу картинку с текстурой:



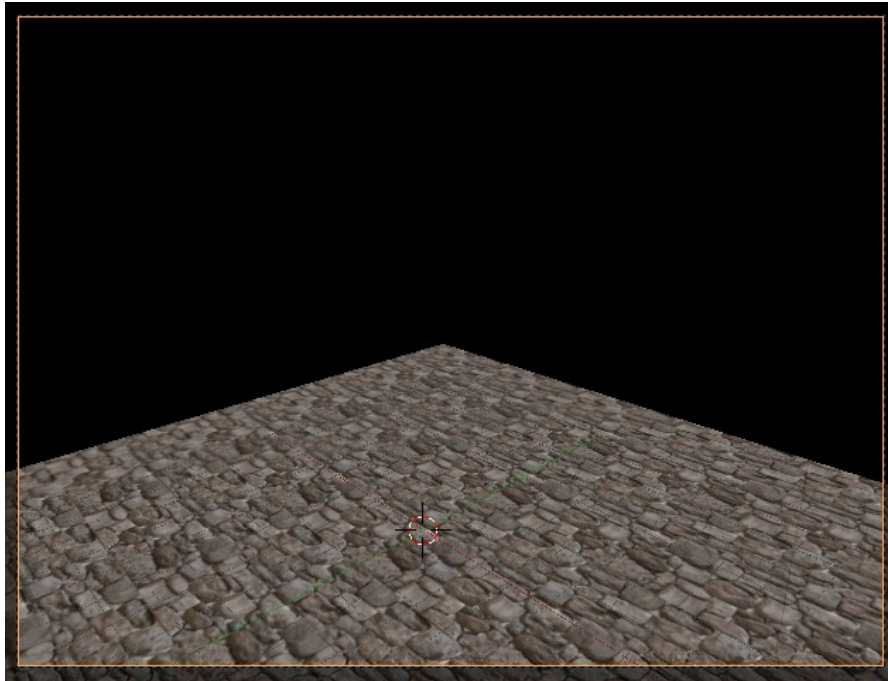
Если после проверки выяснилось, что булыжник на полу слишком велик:



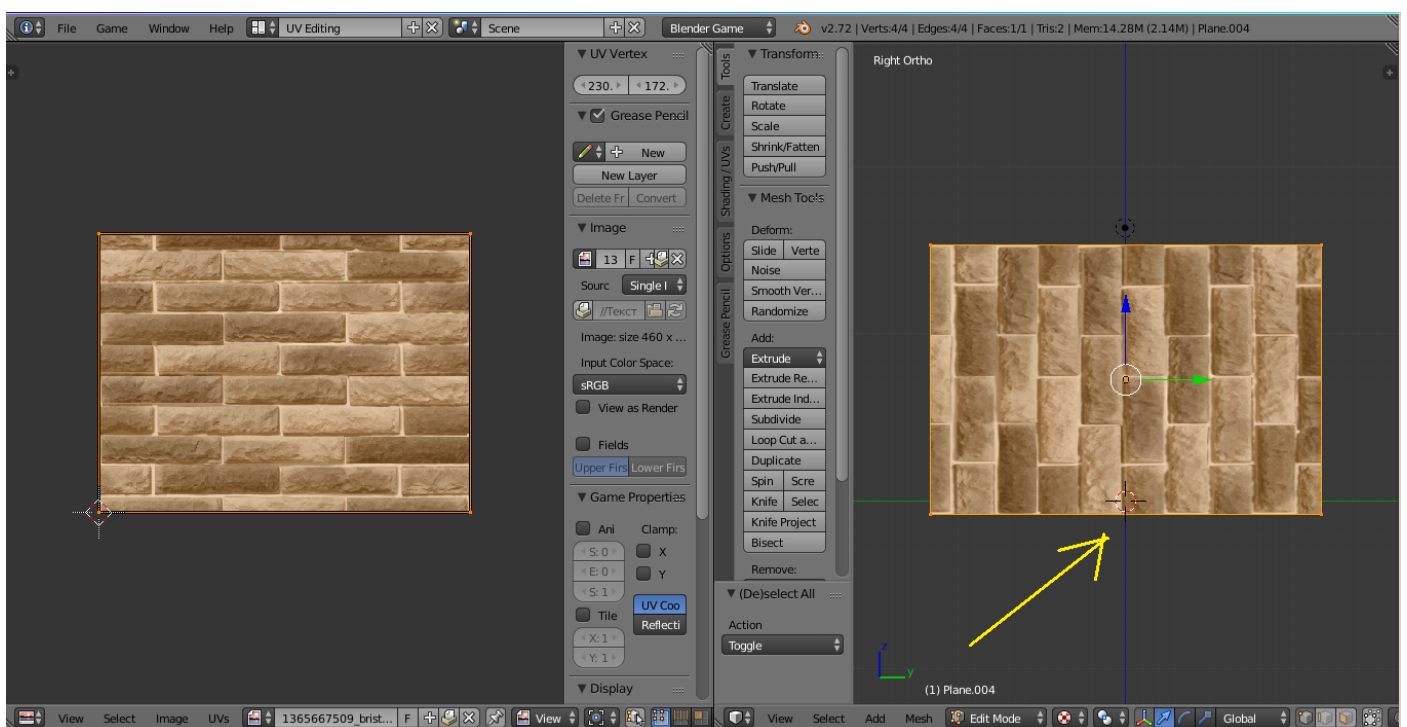
То снова перейдем в UV Editing. В левой части прокручивая колесико мышки уменьшим картинку. После чего (оставив курсор там же) нажмем S и масштабируем (увеличиваем выделенную рамку) настолько, пока нас не устроит вид справа:



Проверяем:

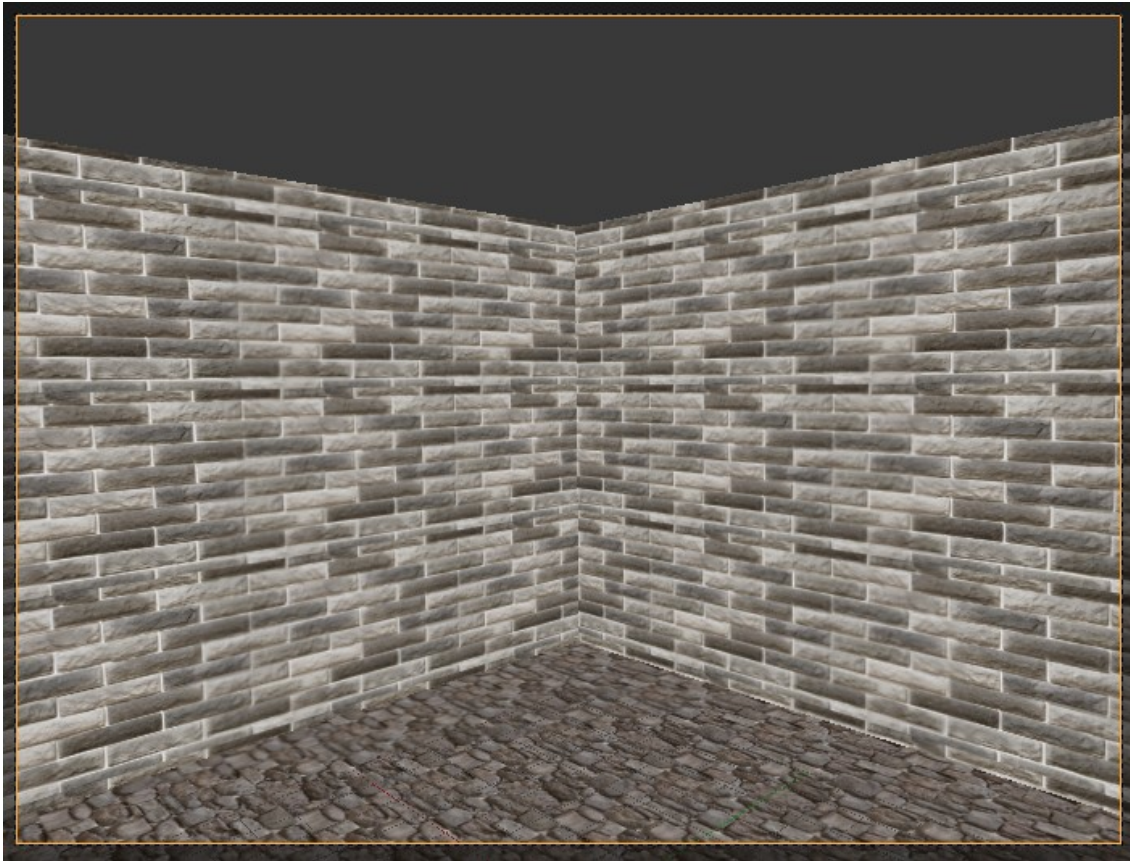


Замечательно! Точно так же поступаем и со стенами. Выделяем плоскости стен и повторяем все шаги, начиная с создания нового материала для стен. Единственной проблемой может стать неверное расположение текстуры:



Но это легко исправить. Слева ждем «А» (выделяем всё), затем «R» и поворачиваем рамку на 90 градусов. А после масштабируем... Готово:





Чтобы окружение лучше воспринималось, добавим по двери на каждой стене. Двери я сделал заранее (тоже в Blender), но в виде картинки png, для экономии ресурсов компьютера:

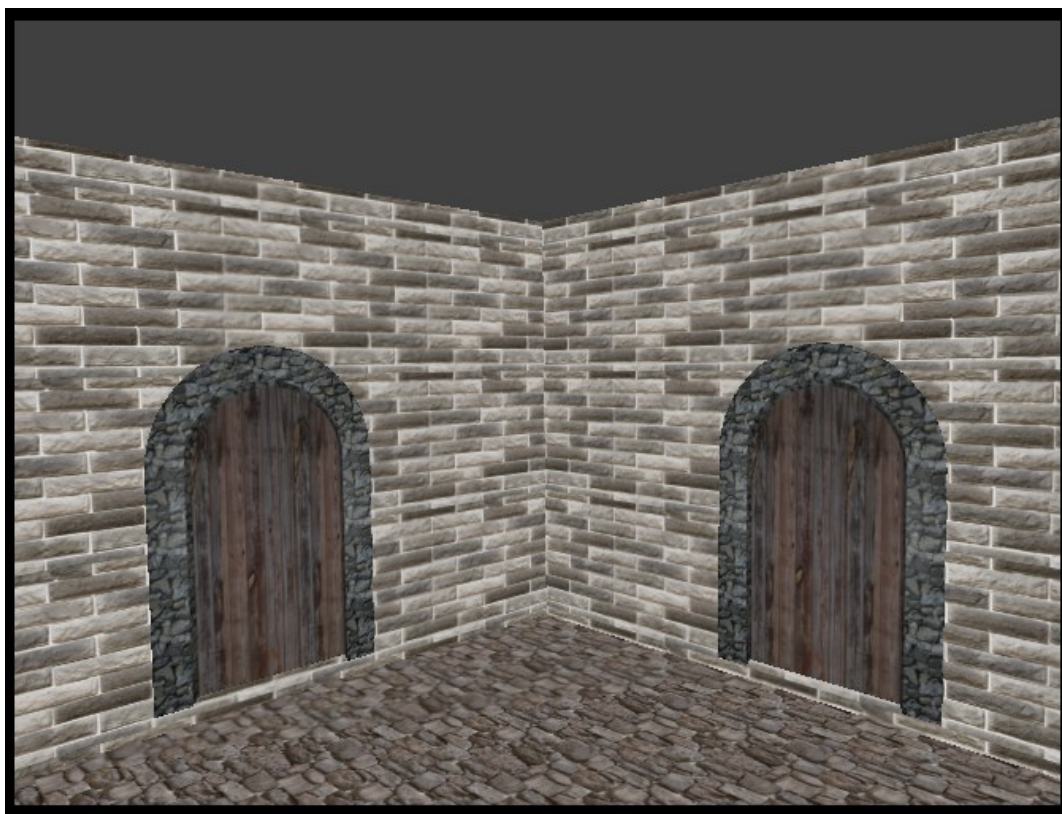


Нам достаточно создать плоскость с новым материалом и наложить на неё текстуру двери все по тому же принципу, что и выше. После этого продублировать плоскость

ещё три раза и прикрепить к стенам. Если после проверки двери будут пропадать, то заменить в настройках материала Alpha Blend на Alpha Clip:



Смотрим, что получилось:

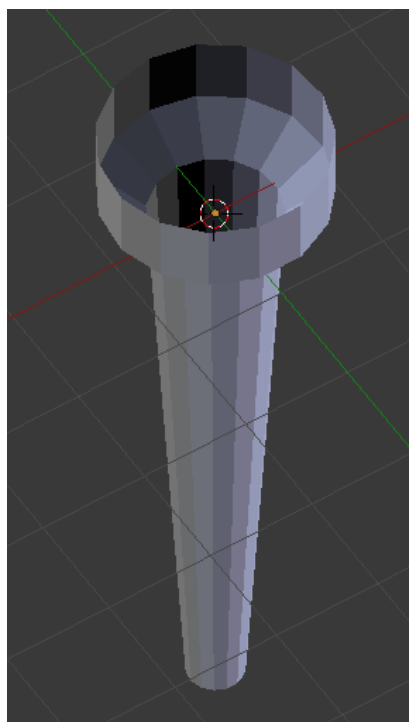


Точно так же делаем потолок. Создаем плоскость, назначаем материал, текстуру и т.д. Ниже показано, как это выглядит у меня:



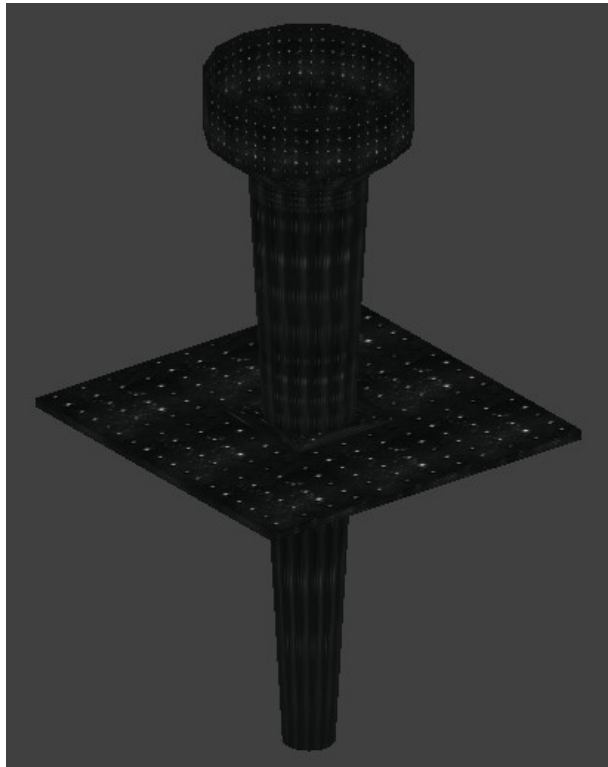
Ну вот мы и подошли к самой интересной теме нашего урока. Чего нам не хватает? В помещении вроде бы светло, но видимых источников света не наблюдается?! Давайте создадим иллюзию их присутствия – расставим по углам четыре факела, огонь которых и попробуем анимировать.

Сначала создадим древко факела. Для этого перейдём на новый слой, чтобы комната нам не мешала, и создадим заготовку из полого цилиндра:





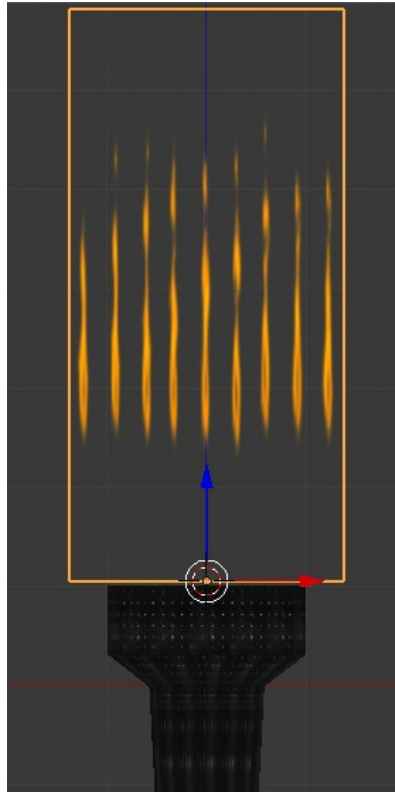
Текстурируем:



Создадим огонь. Для этого нам нужен спрайт последовательности картинок огня. Я нашел вот такой:



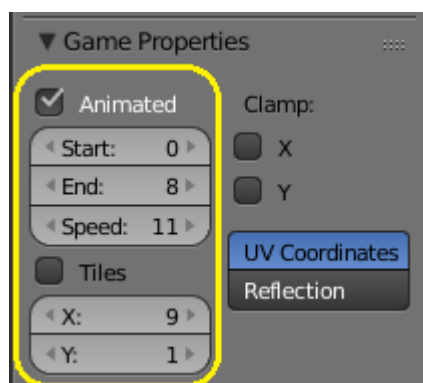
Выделим факел и перейдём на вид спереди. Над ним создадим плоскость. Затем текстурируем её спрайтом, как и в предыдущих случаях:



Идём в UV Editing. В левой части делаем вот так:



Теперь, если проверить нашу анимацию по включению игры клавишей «Р», то выяснится, что анимация работает, но не корректно – просто что-то дергается. Оказывается, пока мы находимся в UV Editing необходимо (при выделенном спрайте) сохранить наш проект. А затем тут же (не выходя из режима редактирования) снова его загрузить (File -> Open...). Почему-то это помогает и анимация начинает работать правильно. Теперь можно понизить скорость кадров до 11:



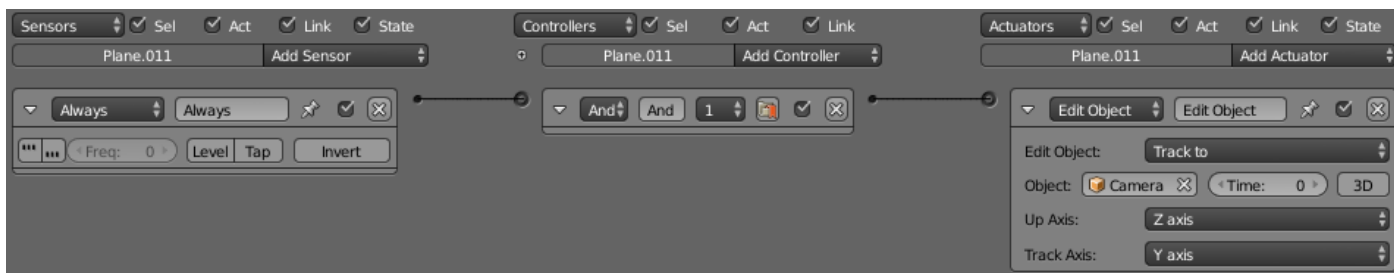
И увеличить плоскость пламени, чтобы оно соответствовало размерам факела ☺ :



Выделяем и огонь и факел, включаем два слоя одновременно и расставляем факелы по углам, предварительно продублировав. Теперь у нас есть горящий огонь в игре:

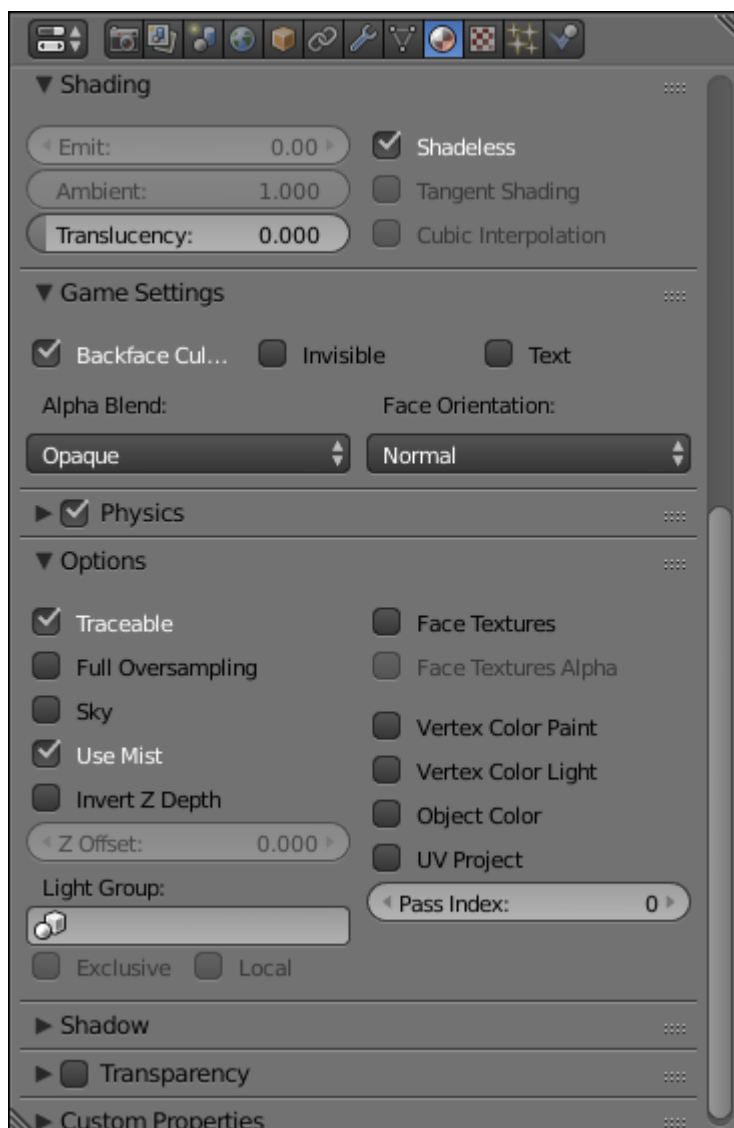


Хотя с факелами это ещё не всё. Нужно зайти в игровую логику и назначить каждой плоскости факела актуатор слежения за камерой, чтобы они были всегда повернуты к камере :

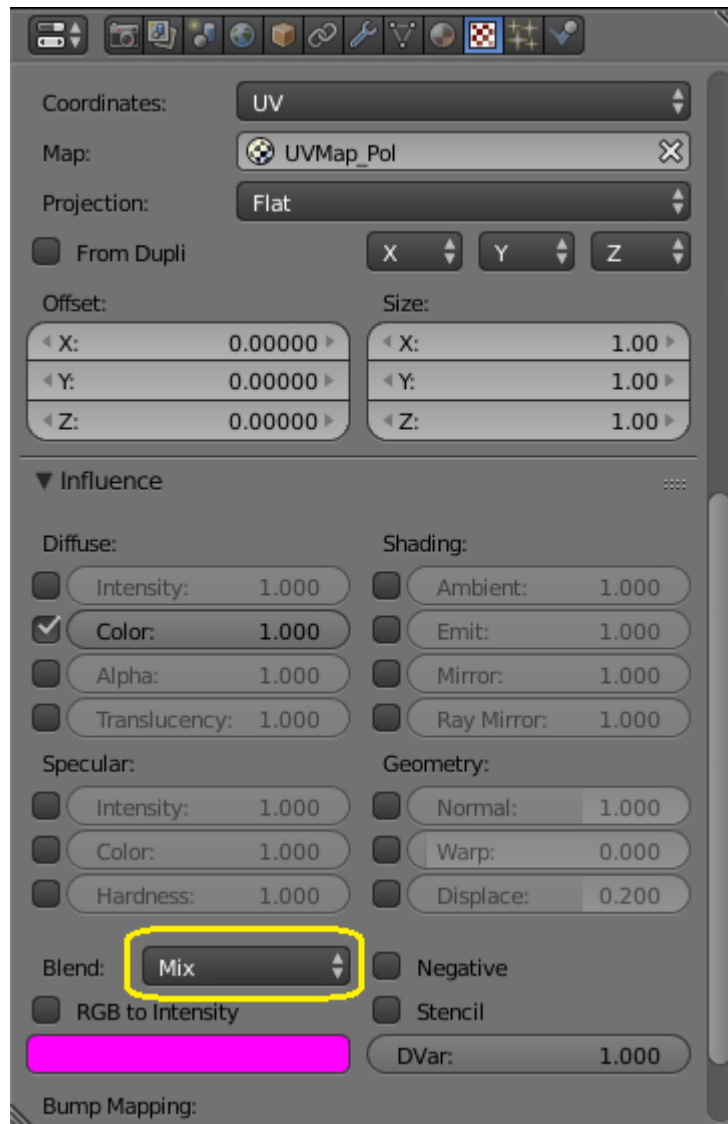


Вот теперь полный порядок. Можно приделать к камере «руки» и заставить её перемещаться по комнате нажатием клавиш со стрелками.

Но прежде хочу ещё раз вернуться к настройке материалов и текстур. Я выбрал такие настройки, которые одинаково хороши и для непрозрачных и для прозрачных текстур. Однако вариантов настроек может быть множество. Если вам не нужно отслеживать прозрачность то можно вообще всё упростить. Вот материал для пола:



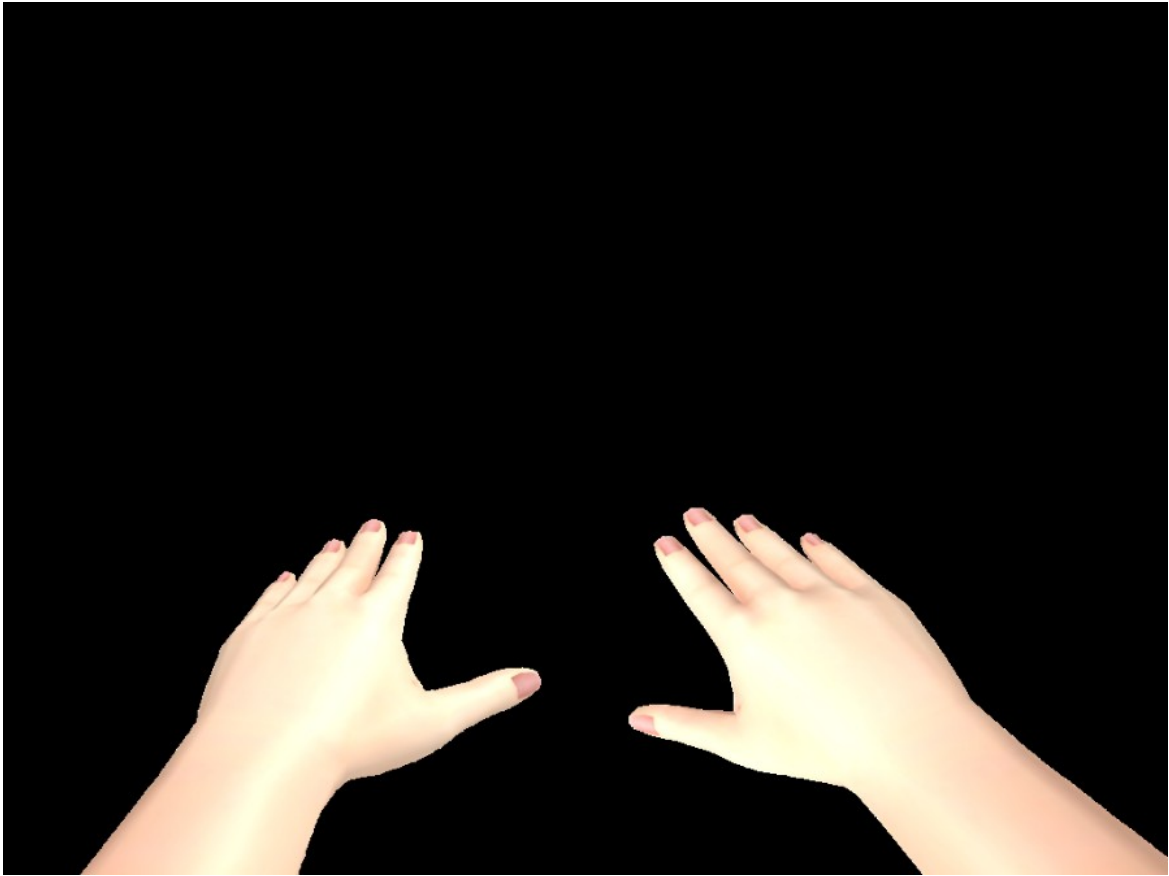
И текстура:



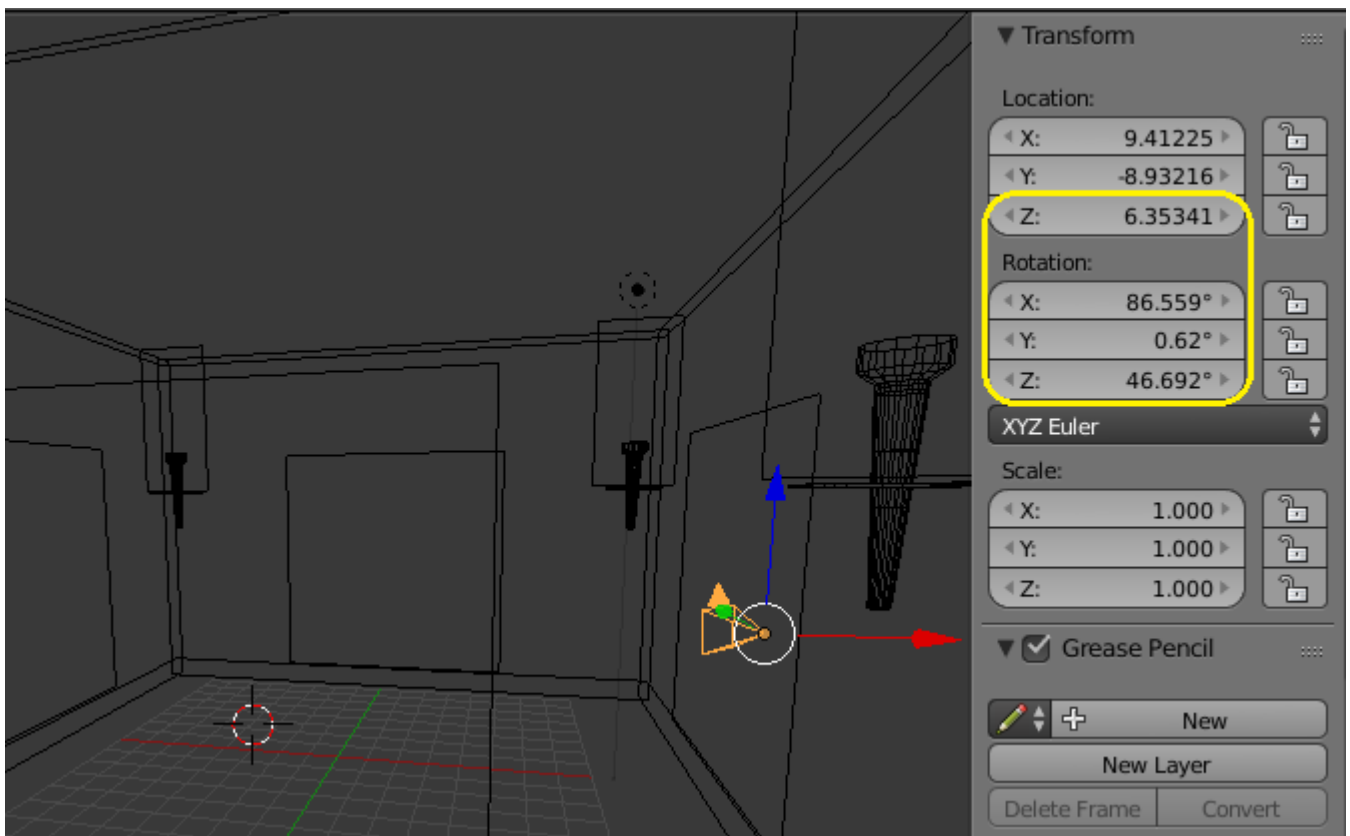
В зависимости от того, какое значение свойства Blend вы выберете в текстуре, так и изменяются настройки свойства Alpha Blend и флажка Vertex Color Paint в материале. Остальное так же, как и выше... Пробуйте, экспериментируйте, ищите свой вариант!

А теперь займёмся руками. Будем делать простейший вариант, опять же чтобы не загружать физику. Сделаем обычную картинку рук в формате png. Размер картинки должен соответствовать размеру камеры, поскольку она прикрепляется именно к ней. Я взял пару картинок трёхмерных рук и сделал вот это:

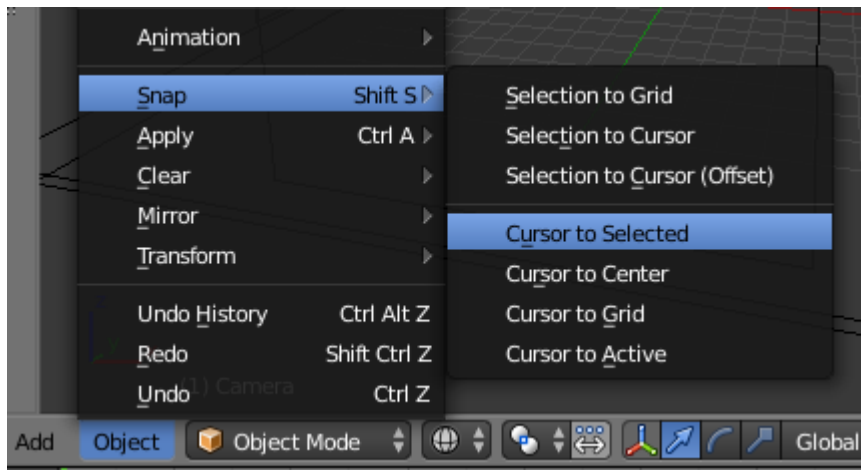




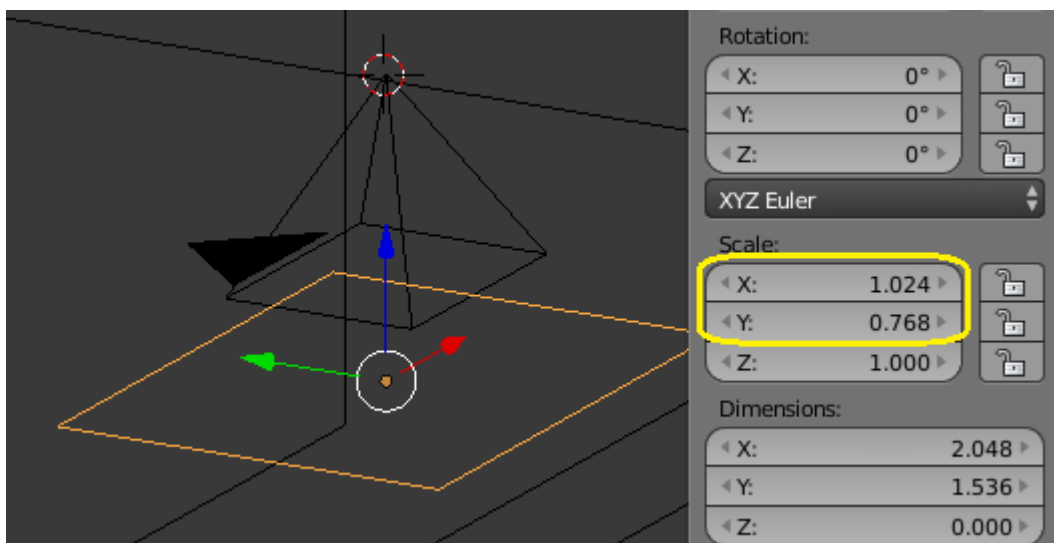
Фон естественно прозрачный. Перейдём на вид Wireframe. Выделим камеру и в полях Location и Rotation запомним выделенные значения:



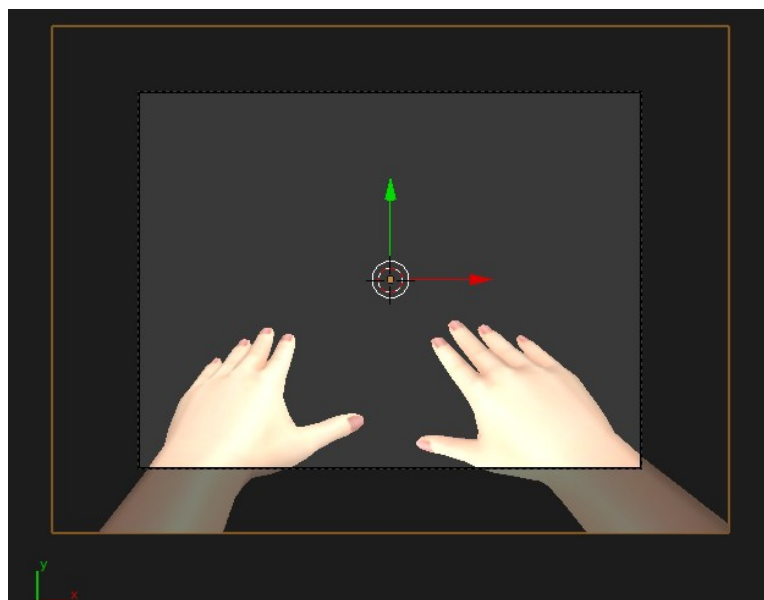
Теперь сбросим их на ноль и переместим курсор в центр камеры:



Создаём плоскость размерами с картинку (или камеру 😊) и опускаем её ниже камеры:



При выделенной плоскости выделяем ещё и камеру. Соединяем по принципу потомок – родитель (Ctrl\_P). Включаем отображение текстур. Включаем вид с камеры. Создаём для плоскости прозрачный материал (как и в случае с факелом) и текстурируем её картинкой рук.:



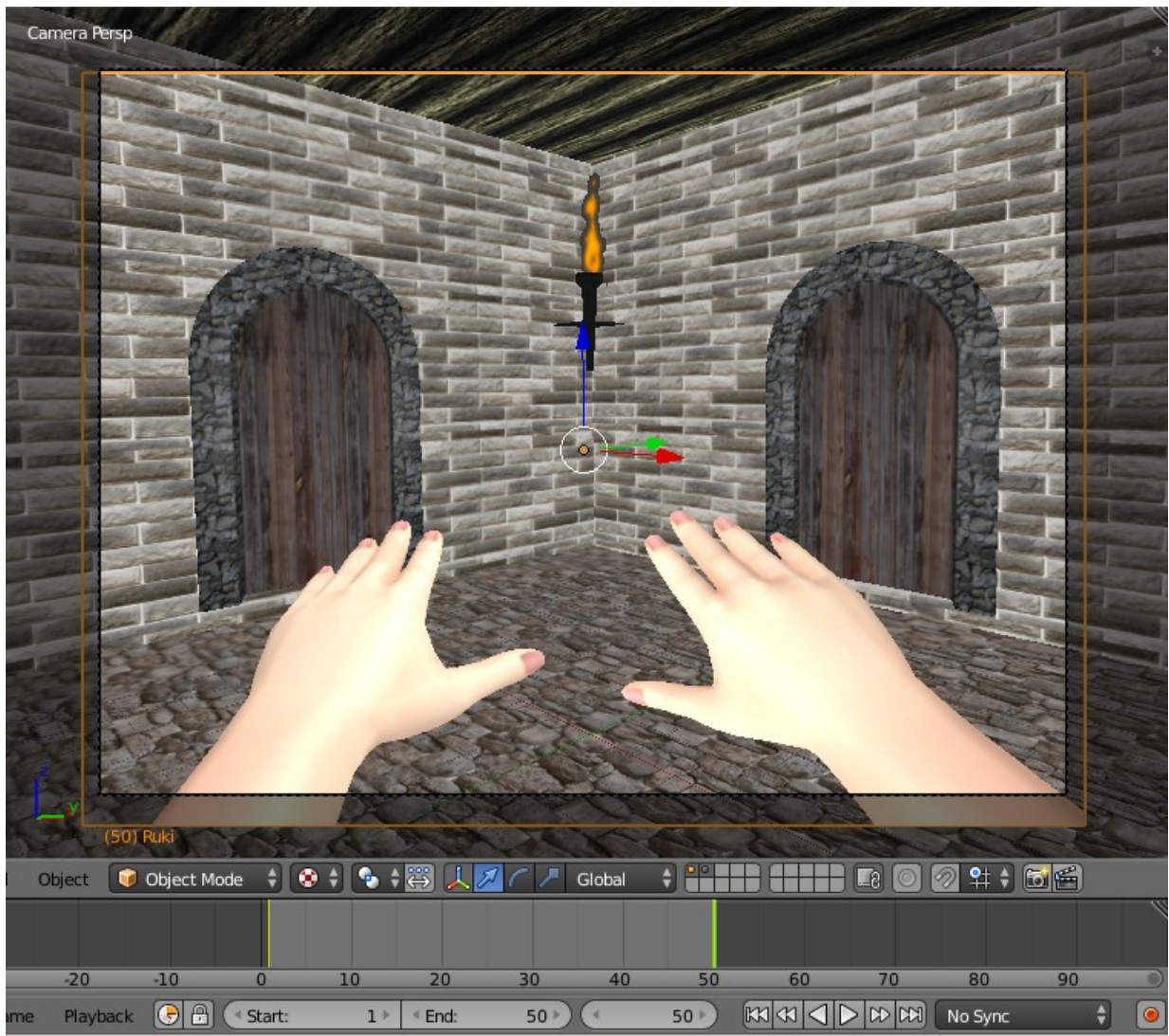
При выделенной плоскости масштабируем её, пока она не станет чуть больше размеров камеры. Вообще, здесь есть свои нюансы. Если вы хотите впоследствии использовать картинку рук, как игровой объект (т.е. чтобы они касались дверей, предметов и т.д.), то есть смысл просто отодвинуть картинку подальше от камеры, а не масштабировать её.

Теперь возвращаем на место все настройки положения камеры (те, что мы сбрасывали на ноль):

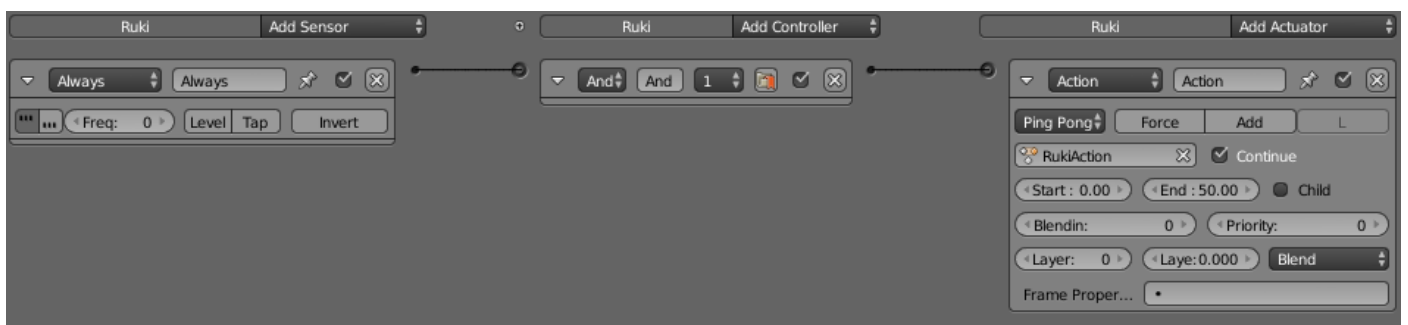


Интересно будет добавить анимацию дыхания. Надеюсь что вы помните, как мы создавали анимацию шлагбаума на прошлых уроках? Так мы создадим иллюзию того, что персонаж живой. Для этого просто создадим кадров 50 (при выделенной плоскости рук!) и два ключа. В последнем ключевом кадре сместим всю плоскость немного вниз:





И в логике рук добавим:

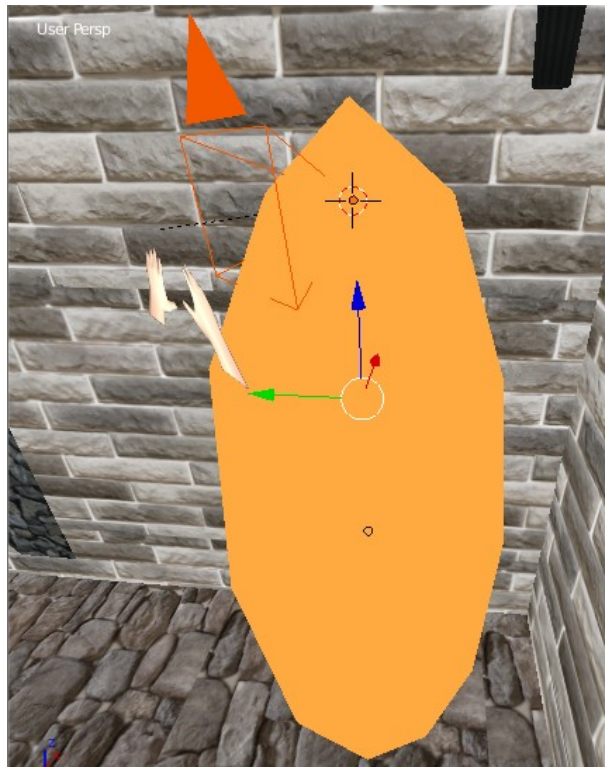


Нажимаем лат.«P» и проверяем. Наши руки должны медленно двигаться вверх – вниз.

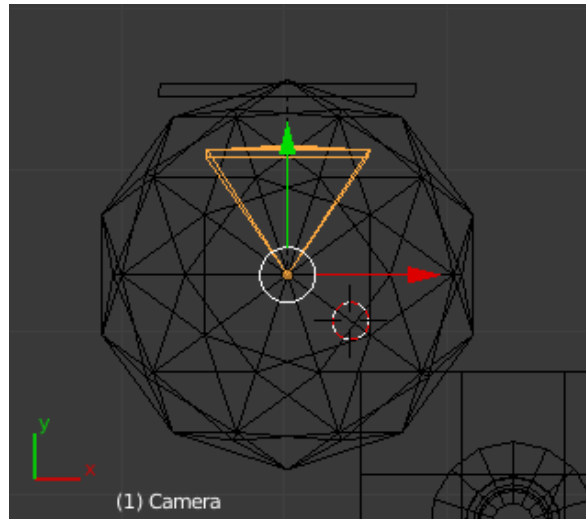
Осталось лишь одно – создать сферу или цилиндр, к которой мы прикрепим нашу камеру и будем этим объектом управлять.

Выделяем камеру и переносим курсор (Object -> Snap -> Cursor to Selected).

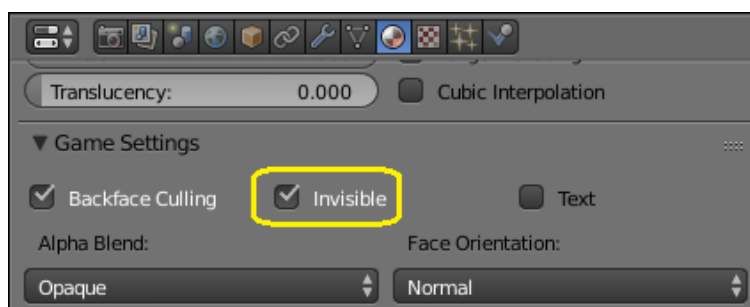
Добавляем объект Ico Sphere и растягиваем его по вертикали. Опускаем чуть ниже так, чтобы камера оказалась на его верхушке. Естественно, сфера не должна касаться пола и быть немного выше его:



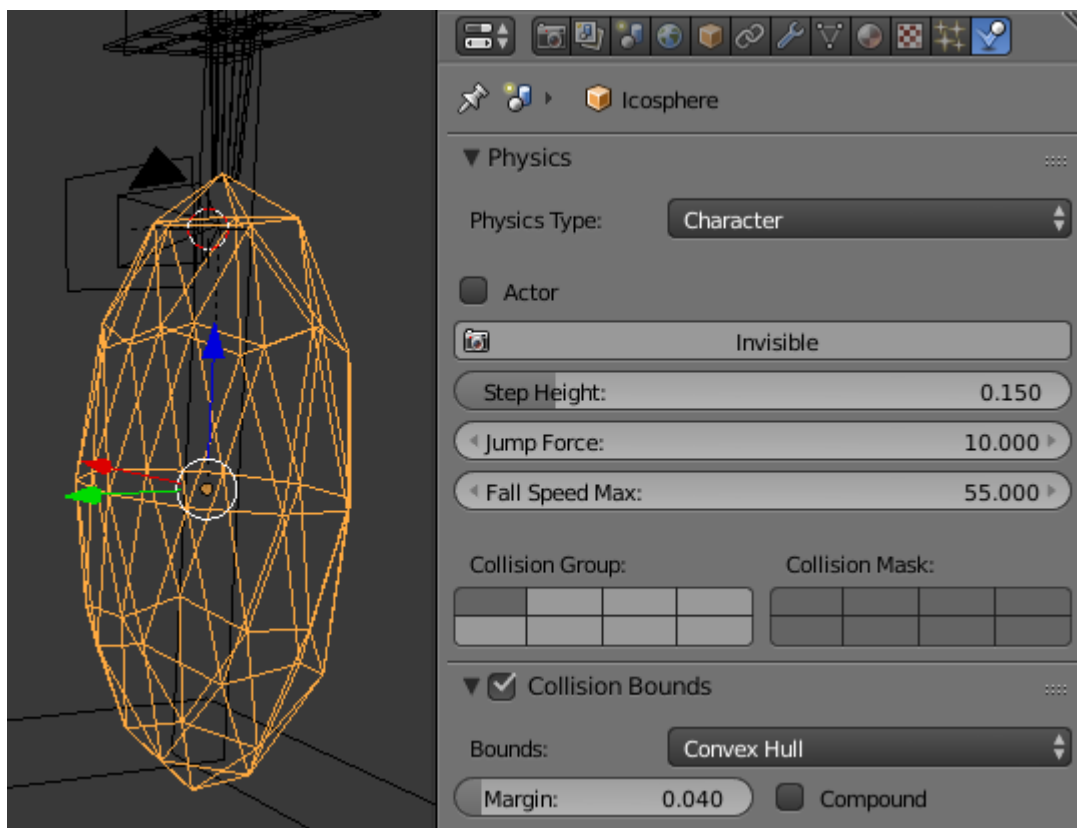
Выделяем камеру и одновременно капсулу (сферу). Создаём связь потомок – родитель. Теперь выделяем камеру и смотрим, что на виде сверху оси камеры и капсулы не совпадают. Значит, разворачиваем камеру по оси Z (должно быть 0), чтобы камера смотрела точно по оси Y:



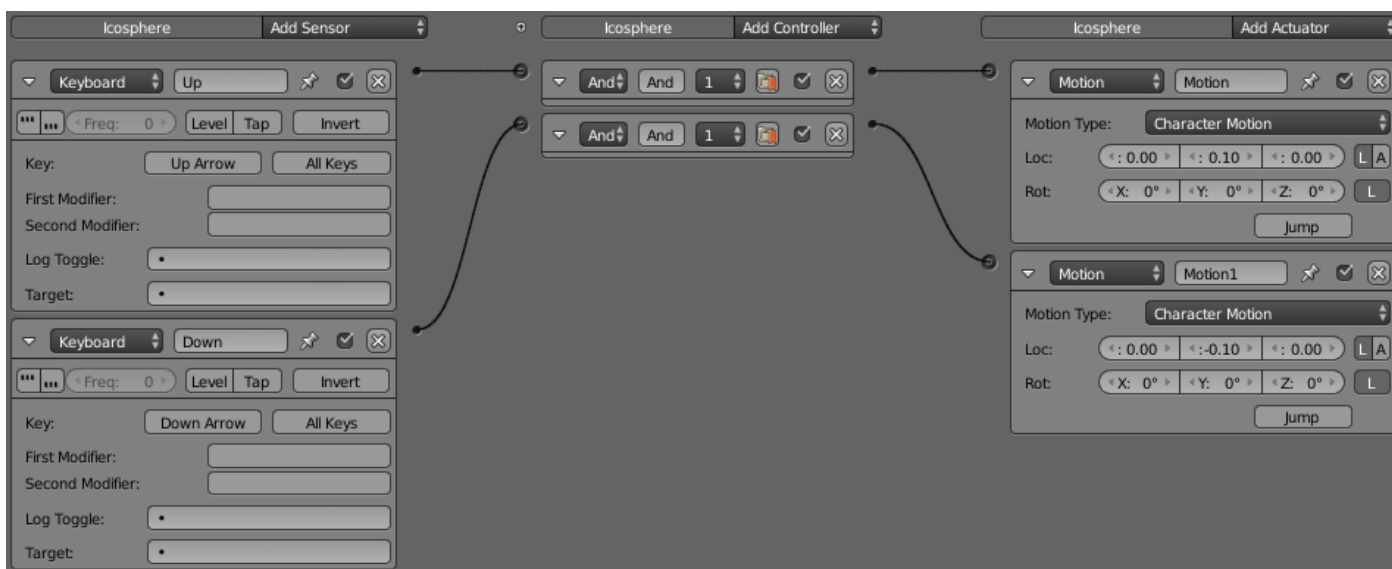
Теперь выделяем капсулу и назначаем ей невидимый материал:



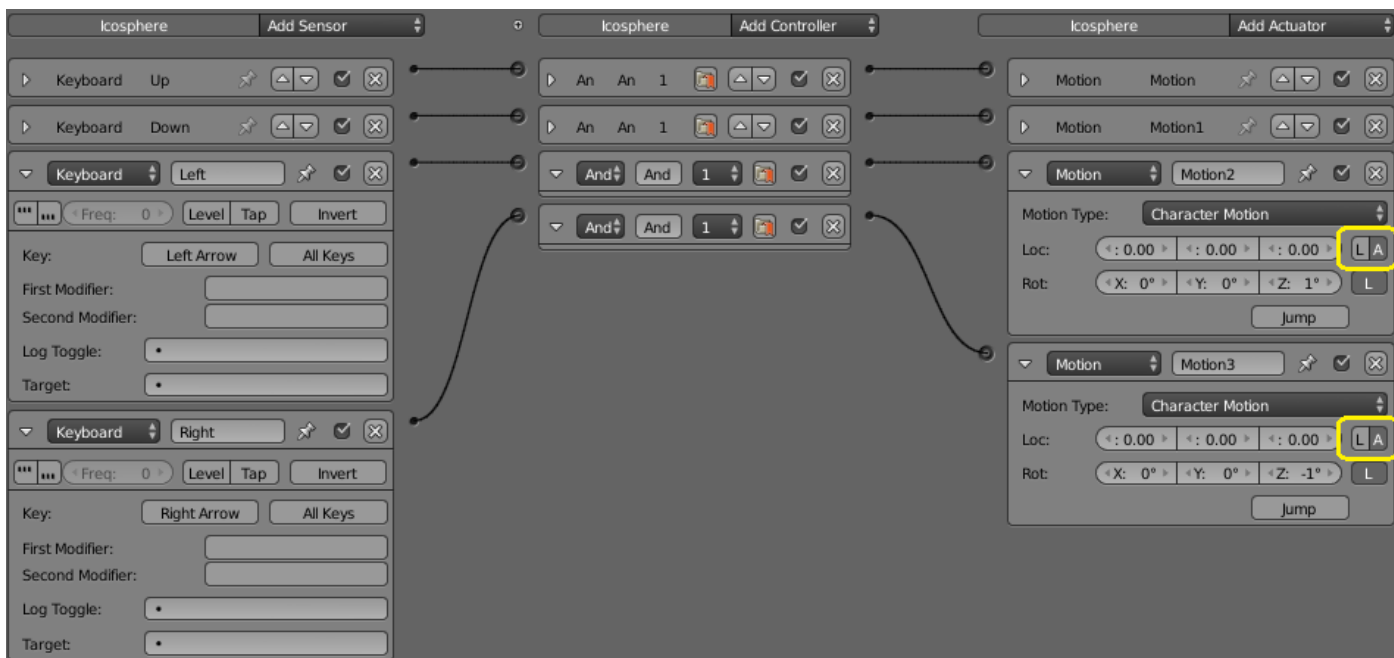
Переходим в физику, где назначаем капсулу персонажем:



Как видно из рисунка, физическая решетка идеально совпадает с объектом. Соответственно должно быть меньше ошибок в обработке столкновений. Наконец, идём в логику:

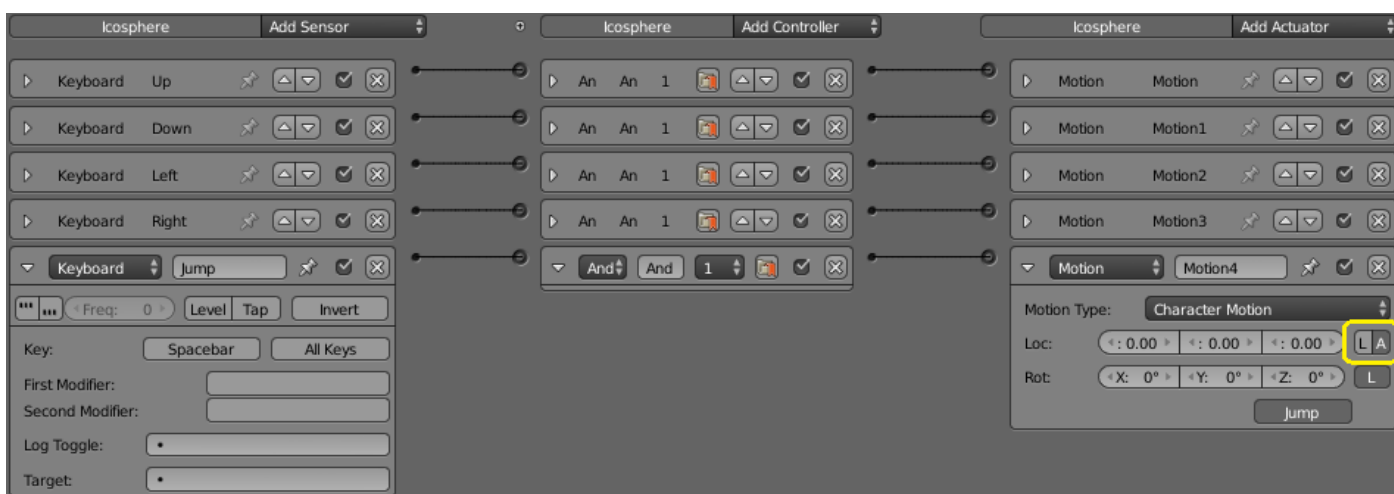


Вперёд – назад и добавляем влево – вправо :



Очень важно включить выделенные кнопки именно так! Иначе при повороте персонаж будет останавливаться.

Ну, и конечно прыжок при клавише пробела:



Готово! Включаем вид с камеры и запускаем игру!





28 июня 2015 года.

Составил **Niburiec** для сайта <http://blender-game.ucoz.ru>