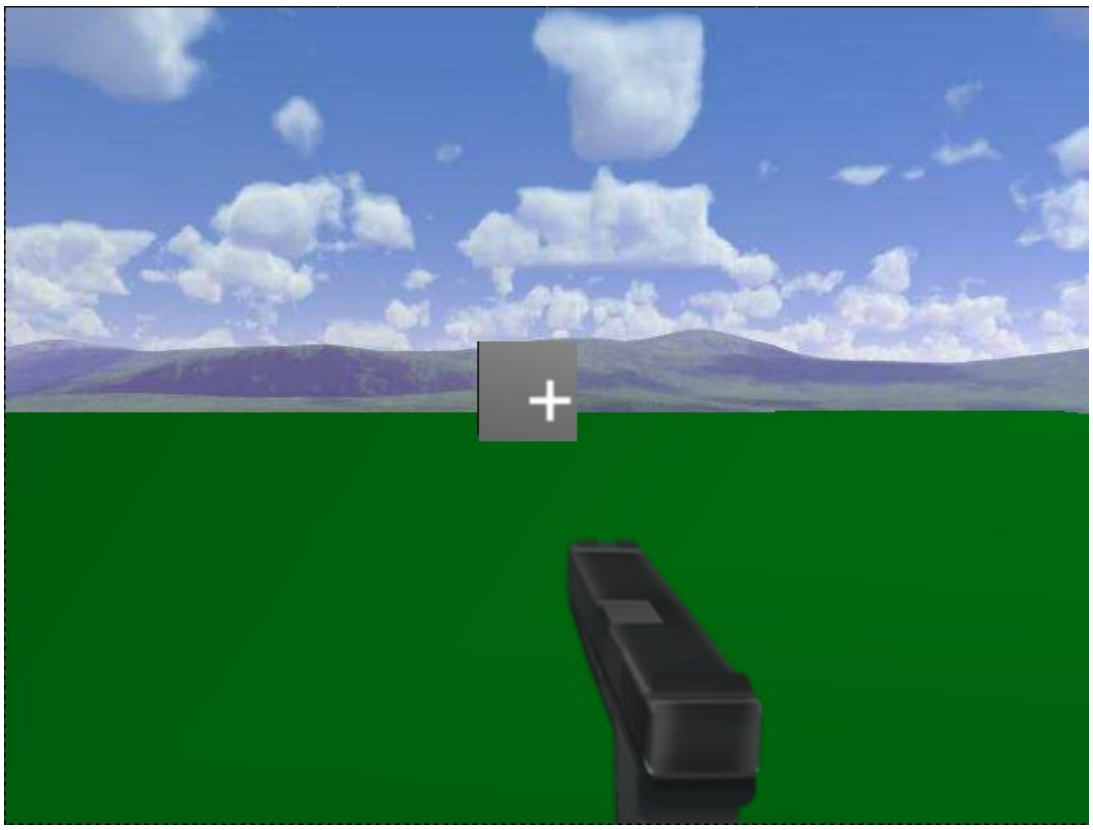
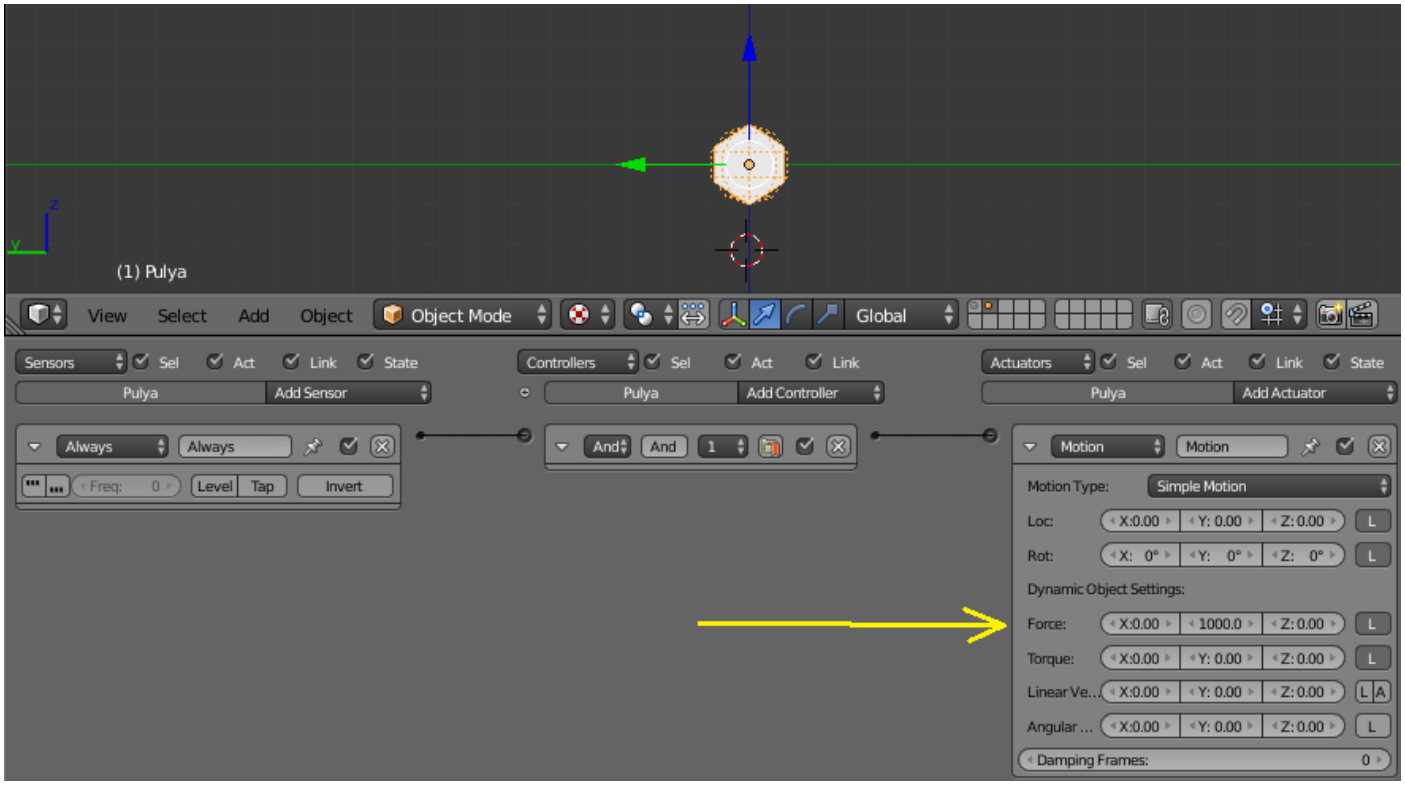
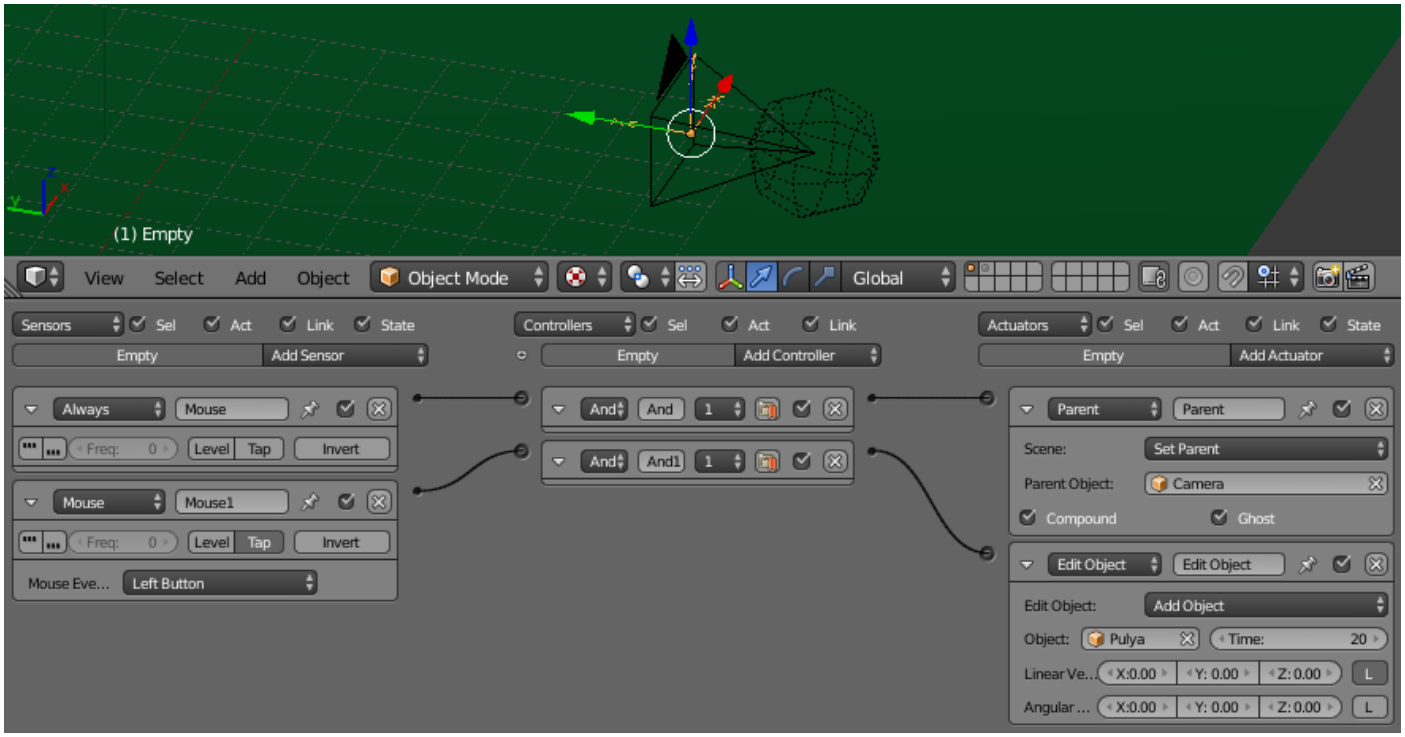
Шутер от первого лица это довольно популярный вид стрелялок. Как создать персонаж от первого лица мы подробно рассмотрели в уроках 26 и 27. Здесь мы рассмотрим вопрос о том, как заставить его стрелять. Есть много вариантов создания выстрела. Мы возьмём самый простой и надёжный – использование пули из дополнительного слоя. И так, создаём «землю», персонажа из камеры и врага в виде кубика. Персонажу в физическом мире присваиваем тип **Character**, а кубику **Rigid Body**. Незабываем, что персонаж должен быть **Actor**. На будущее, если мы не присвоим персонажу (камере) **Actor**, то его не смогут обнаружить такие сенсоры, как **Near**. В качестве же оружия создадим сцену с HUD, куда вставим любую картинку с прозрачным фоном (png). Напомню, если в сцене с HUD у вас нет освещения, то материалу картинки нужно поставить **Shading -> Shadeless** (как и в Background). Иначе вы увидите белый фон ☺. Я себе позволил ещё и сцену Background добавить в виде неба. Здесь тоже нужно подправить материал – сделать его прозрачным. Обязательно запекаем все текстуры, как в предыдущих уроках. В итоге у меня это выглядит примерно так ☺:



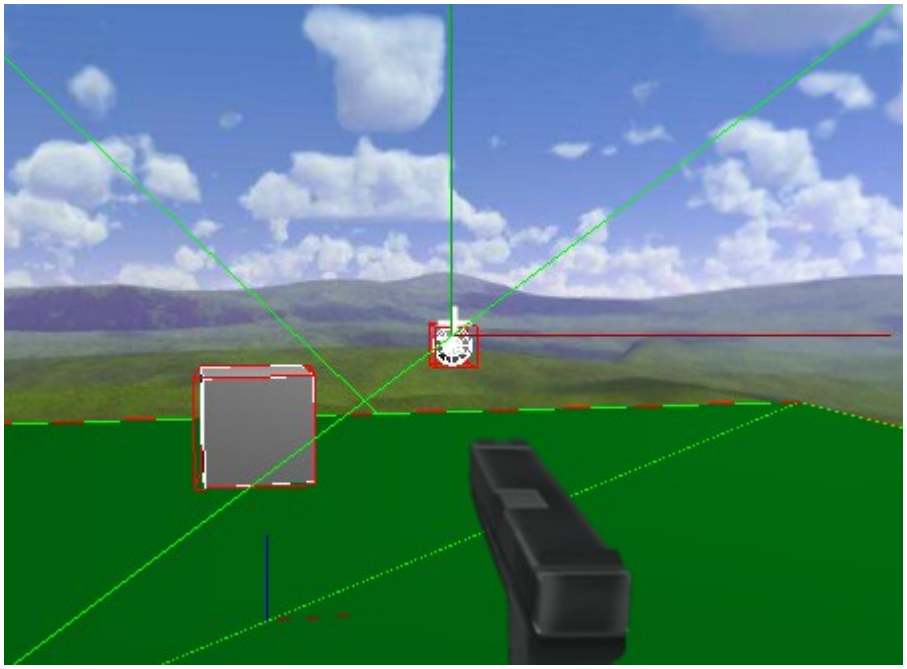
Пора заняться стрельбой. Есть по крайней мере два способа. Самый очевидный это создать динамический объект и выстреливать его в нужном направлении. Но есть ещё один способ – прицеливание и стрельба по лучу. Сенсор **Ray** нацеливается на объект и найденный объект просто уничтожается. Но мы пока будем использовать более простой первый способ. Переходим на дополнительный слой и создаём пулю – куб, сферу, цилиндр, в общем что угодно. Естественно уменьшаем её. Только не перестарайтесь, а то столкновения будут срабатывать не корректно. Логика пули примерно такова:



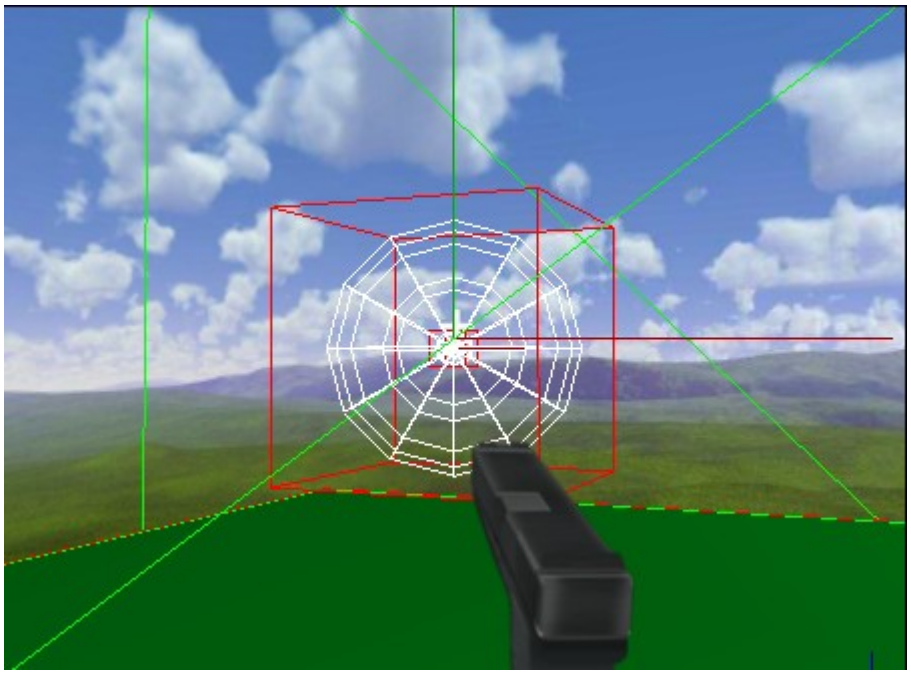
Назначаем импульс (**Force**) пуле. Так и назовём её **Pulya**. Она должна быть либо **Dynamic**, либо **Rigid Body**. Я решил сделать её динамической (чтобы исключить вращение и соответственно лишнюю загрузку просчёта физики). Возвращаемся на основной слой и создаём пустышку перед камерой (**Empty -> Arrows**). Именно в координатах пустышки будет появляться пуля. Привязываем её к камере по типу Потомок -> Родитель. Я решил это сделать программно:



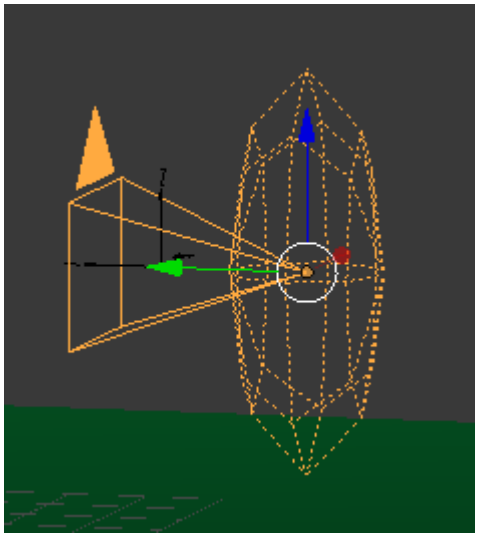
Появление пули происходит по нажатию левой кнопки мыши. И появится она на 20 (миллисекунд, как я понял… хотя могу ошибаться). Время жизни пули напрямую зависит от скорости её полёта. Чем выше скорость, тем меньше нужно делать время жизни. Запускаем и пробуем. При попадании в куб напротив, куб должен смещаться (если он тоже динамический объект):



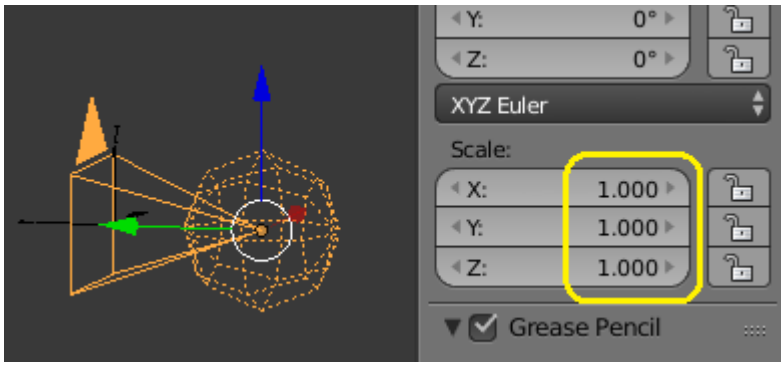
Я намеренно установил отображение физических границ объектов для наглядности. Естественно, если вы отключите в физике пули **Invisible**, то она станет невидимой, но все физические свойства сохранит. Это можно увидеть по границам физики пули:



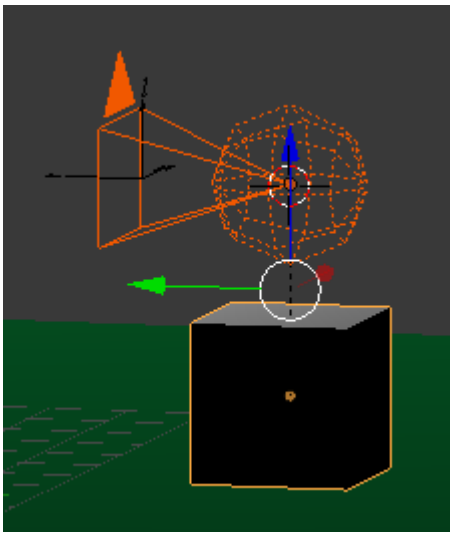
От скорости полёта пули зависит ещё и точность попадания. Поскольку пуля есть физический объект и обладает массой, то и лететь она будет по параболе. Это можно использовать в игре с пушечными ядрами, например ☺. Осталось лишь назначить кубу столкновение с пулей и уничтожать его при попадании. Собственно на этом можно было бы и завершить, однако думаю что физические свойства персонажа (камеры) нужно рассмотреть немного подробнее. Дело в том, что если мы растянем физические свойства камеры по вертикали, как на рисунке:



то пуля при повороте камеры не будет появляться по центру. Она всегда будет смещена в сторону. А это для нас не есть хорошо ☺. Ведь стрельба превращается в лотерею – попаду или нет ☺. Опытным путём я установил, что размеры физики камеры должны быть всегда по умолчанию, т.е. равны 1:

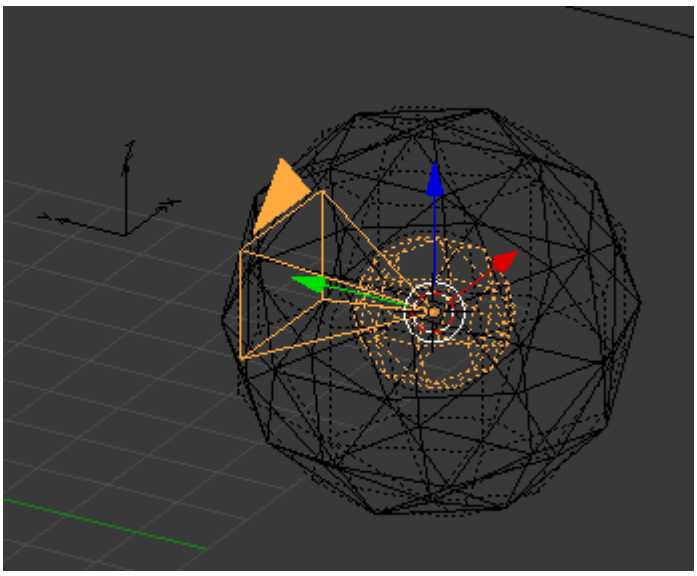


Это гарантирует вылет пули без смещения. Тогда как нам быть с ростом персонажа? Это до какой же степени нужно уменьшать размеры противников, чтобы они нормально выглядели в камере? Выход, как может показаться, лежит на виду. Достаточно создать кубик, например, и прицепить к нему камеру:

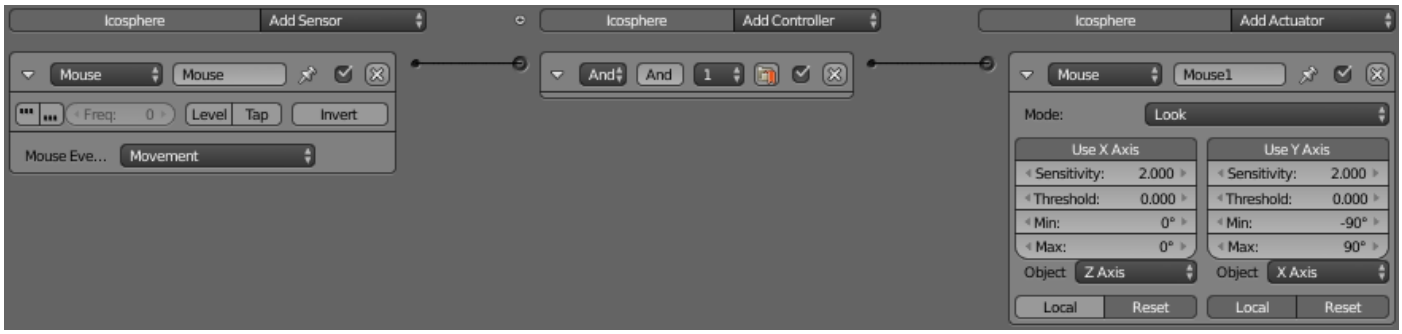


Но, я думаю, это не самое лучшее решение. Очень сложно настроить синхронное управление мышью. Допустим, если я хочу чтобы персонаж вращал голову мышью и двигался вперёд с помощью правой кнопки мыши… Либо пришлось бы управлять кубом с клавиатуры, а камерой – мышью. Что, на мой взгляд, не очень удобно. Так как же быть? Можно поместить камеру внутрь другого объекта,

например, сферы. Соединить их «родственными связями», сделать камеру **No collision** , а сферу **Character**(хотя лучше просто **Dynamic**). Главной будет сфера, а камеру сделать потомком:



Управление вращением и движением полностью передать внешней сфере:



Ну и конечно сделать её **Invisible**. Может быть придётся вынести пустышку немного вперёд… Попробуйте растянуть сферу по вертикали, как изменится её поведение. Или попробуйте заключить камеру в куб… Это будет заданием для самостоятельного решения ☺. Экспериментируйте! Посмотрите, как изменится поведение при смене типа столкновений… Это очень интересные опыты ☺.

26 октября 2015 года.

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