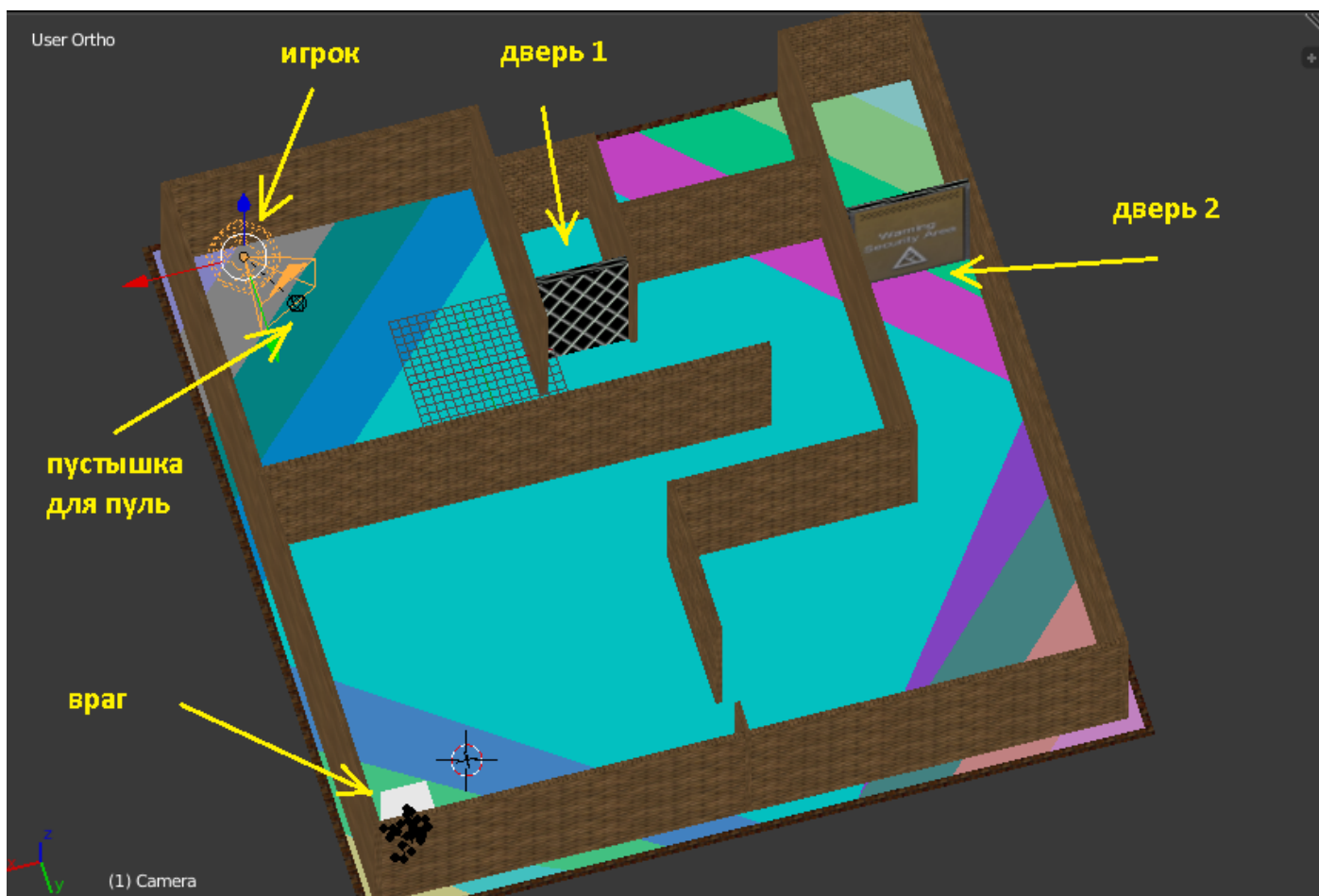
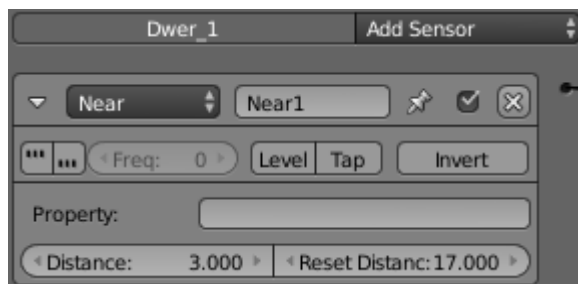


Пришло время собрать нашу «стрелялку» воедино и посмотреть что из этого выйдет ☺. До «крутой» игрушки она конечно сильно не дотягивает. Но зато мы включили почти все основные элементы. У нас есть игрок, возможность стрельбы и настоящий враг. Пускай наш враг пока не наносит вреда игроку, но это довольно легко устроить, используя всё тот же принцип столкновений материалов. Точно так же довольно просто сделать подсчёт патронов или урон здоровья, даже не прибегая к написанию скрипта на **Python** (смотрите прошлые уроки).

Для данного примера я построил небольшое помещение и оснастил его двумя автоматическими дверями. За одними дверями впоследствии можно расположить оружие или патроны, а за другими – переход на следующий уровень или телепорт. Дверями управляют сенсоры **Near**. Чтобы они правильно срабатывали нужно помнить, что радиус обнаружения исходит из центра масс. И если дверь открывается вверх, то и центр масс уезжает туда же ☺. При этом обнаружение прекращается. Поэтому расстояние в свойствах сенсоров нужно определять опытным путём:



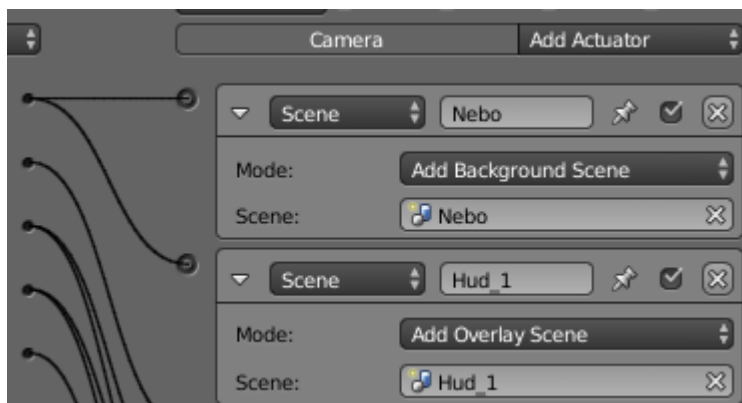
Например, для первой двери у меня вот так:



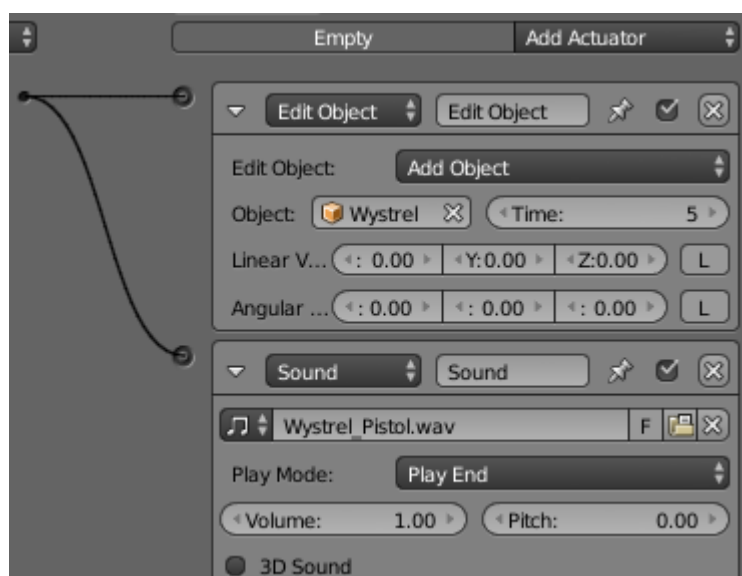
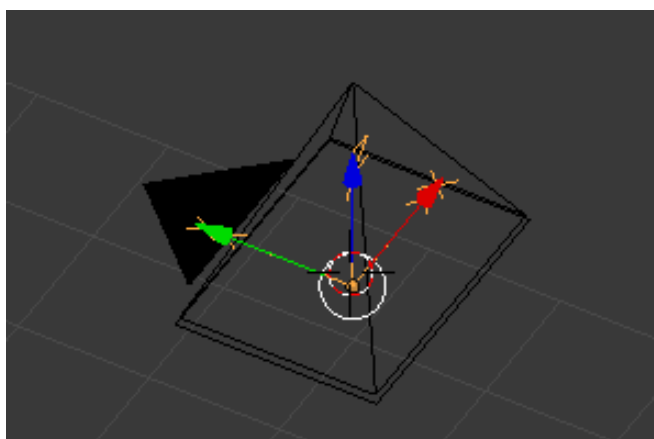
Для игрока я не стал изобретать велосипед и сделал столкновения камеры обычной сферой, просто увеличив её. Пустышка для пуль тоже сфера. Сама пуля тоже сфера. Кроме того, я не стал делать её

невидимой для наглядности. Так вы сможете убедиться, что она действительно есть и является физическим объектом. Потому что отскакивает от стен с рикошетом.

В качестве пистолета я взял обычную картинку. Установил её в новой сцене (а не на другом слое!) и подключил её через **Add Overlay Scene**. И естественно добавил небо:

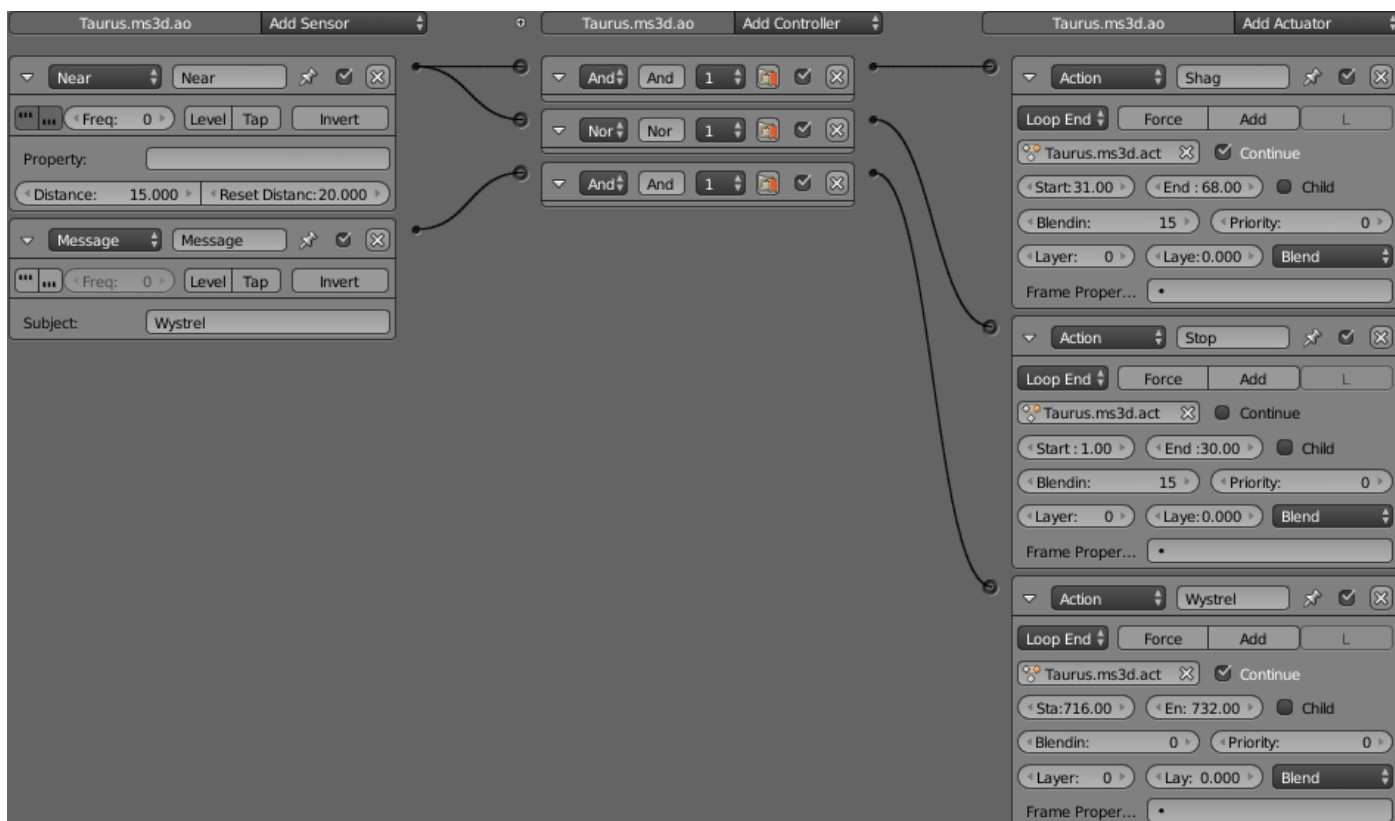


Для вспышки от выстрела я нашел простое и интересное решение. В сцене с пистолетом я сделал дополнительный слой и там разместил прозрачную картинку со вспышкой. А подключаю её при выстреле по принципу пули, просто помещая её всякий раз в пустышку ☺:



В игре есть звуки. Шаги, выстрел, открывание и закрывание дверей, попадание во врага. По желанию можно добавить и другие звуки. Главное не забывать чтобы звуки были в формате **wav**, поскольку этот формат **Blender** обрабатывает без проблем.

Враг заключён в куб. Куб имеет материал и является динамическим объектом. Если его сделать статическим, то возникают некоторые трудности. В частности он будет проходить сквозь стены и т.д. Сам куб перемещается по плоскости, которая является навигационной сеткой (смотри прошлые уроки). Если сенсор **Near** обнаруживает игрока, то куб следует за игроком. У костей модели включается анимация ходьбы. При отсутствии обнаружения куб останавливается, а у костей анимация изменяется на анимацию покоя. При попадании в куб срабатывает актуатор **Message**, который передаёт сообщение в сенсор **Message** костей. У последних меняется анимация:



Заметьте, у сенсора **Near** костей включены обе кнопки (с точками).

Более никаких особенностей у игры нет. Файл проекта небольшой, но запакован в архив **RAR** вместе с картинками, моделью врага и звуками. Я решил так сделать потому, чтобы вам было проще, если вы захотите из всего этого сделать свой проект ☺. Ищите его на странице готовых проектов.

На этом я завершу разбор простейшего шутера от первого лица, поскольку сам не являюсь ярким поклонником данного жанра. На мой взгляд, примеры дают общее представление о способе построения простейшей игры для самостоятельного повторения. Верю в то, что у вас получится гораздо лучше ☺! Возможно, в дальнейшем мы ещё вернёмся к этой теме, если возникнут какие-то новые интересные моменты.

23 ноября 2015 года.

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