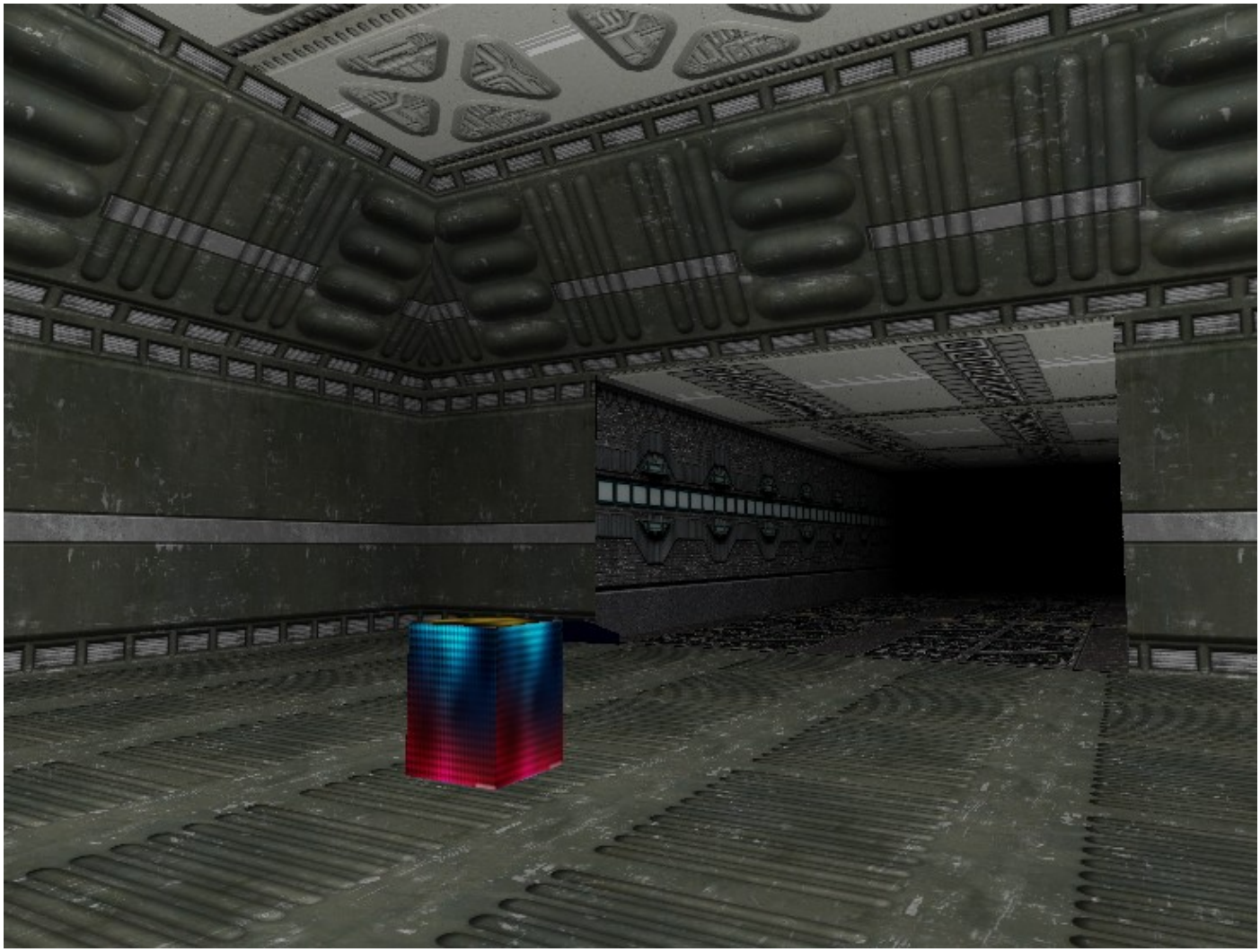
В этом уроке мы начнём знакомиться с замечательной возможностью, которую нам подарили разработчики Blender – использование видео текстур в BGE. Если вы помните, в уроке 26 мы применяли спрайтовую анимацию пламени факела. Хороший способ «облегчить» игру и не использовать частицы. Однако немного удручает необходимость применения режима **GLSL** , который не только усложняет работу процессора, но и может некорректно работать на некоторых типах старых видеокарт. А ведь наша задача должна состоять в том, чтобы создать игру, запускающуюся на большинстве компьютеров. Иначе, какой смысл любоваться своим творением в одиночку ☺. Применение видео текстур позволяет нам оставаться в обычном режиме мультитекстур, но с гораздо более реалистичной анимацией и более широкими возможностями её применения.

И так, рассмотрим модуль **bge.texture**. Он позволяет манипулировать текстурами во время игры. Источники текстур могут быть различными: видео файлы, файлы изображений, видео-захват, буфер памяти, камеры рендеринга самого Blender или несколько источников одновременно. Видео и графические файлы даже могут быть загружены прямо из интернета. Для этого достаточно использовать вместо имени файла его **URL**. Мало того, вы можете применять фильтры на изображениях, что позволяет создавать различные видео эффекты. **bge.texture** использует **FFmpeg** для загрузки изображений и видео. Вот лишь некоторые форматы и кодеки, которые поддерживаются **FFmpeg**:

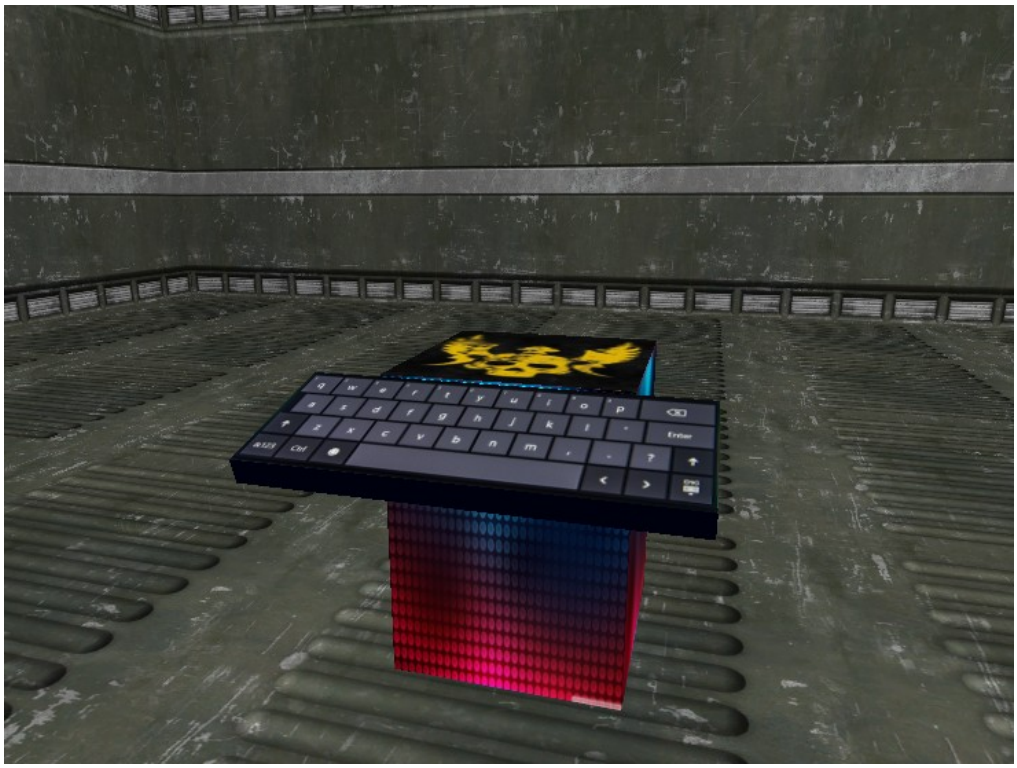
```
* AVI
* Ogg
* Xvid
* Theora
* dv1394 camera
* video4linux карты захвата (включает в себя множество веб-
камер)
* videoForWindows карты захвата (включает в себя множество
веб-камер)
* JPG
```

На первый взгляд довольно сложно для мгновенного понимания ☺. А если учесть, что включается всё это только посредством скрипта на **Python**, то начинающим строителям игр становится вообще страшно ☺. Однако, вы можете мне поверить – всё очень и очень просто. И начнём мы, как и всегда, с разбора примера.

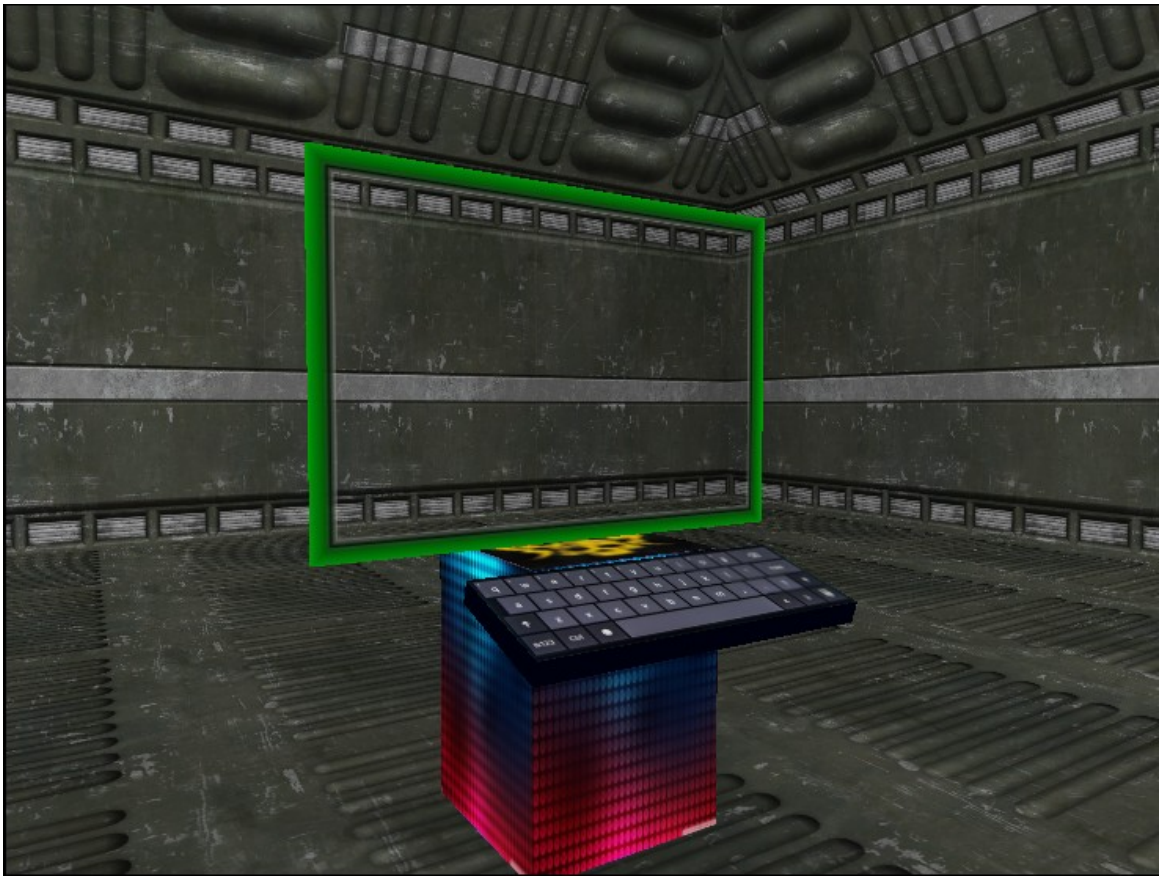
Создадим сцену комнаты с телевизором по центру. Я решил сделать сцену сразу из двух одинаковых комнат, соединённых коридором (с расчётом на последующие уроки):



Поскольку я не использую **GLSL**, я включил тёмный туман, чтобы немного оттенить углы. По центру стоит терминал, на котором и будет отображаться видео:

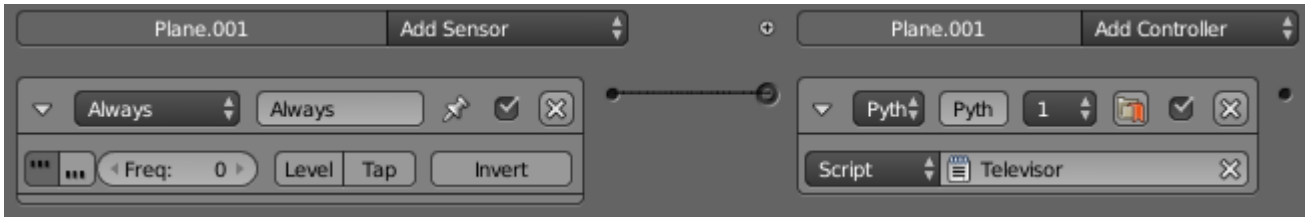


Для этого поставим на терминал панель и назначим ей материал. Текстуру я наложил сразу. Хотя она всё равно потом устанавливается в скрипте, а видео файл будет всего лишь её подменять в процессе игры. Текстура может быть любая. Материал может быть прозрачным или нет - для видео с альфа-каналом это не имеет никакого значения. Я сделал фрагмент видео (**AVI** с альфа-каналом) размером **320/240**. Значит приблизительно таких размеров (кратных им) должна быть и панель, чтобы не пришлось программно использовать масштабирование самого видео:



Замечательно то, что **FFmpeg** сам обрабатывает альфа-канал и выводит прозрачную картинку. А это открывает для нас широкие возможности его применения. Хотя, справедливости ради следует отметить, что просчёт альфа-канала тоже в некоторой степени загружает компьютер. Однако продолжим. Файлы текстуры и видео должны быть в одной папке с blend файлом! Теперь нам нужно уяснить один момент – видео крепится к конкретному объекту, и для каждого такого объекта пишется скрипт запуска видео файла. Сам скрипт очень прост. Весь секрет состоит в том, что код обрабатывает каждый кадр. Значит, для воспроизведения фильма он должен повторяться каждый шаг, т.е. по сути - бесконечно. Описание всех функций модуля **bge.texture** вы найдёте в разделе **BGE и Python**. Здесь же мы рассмотрим лишь некоторые.

Заходим в логику и выделяем объект экрана нашего телевизора. Подключаем сенсор **Always** и пишем скрипт **Televisor** для телевизора:



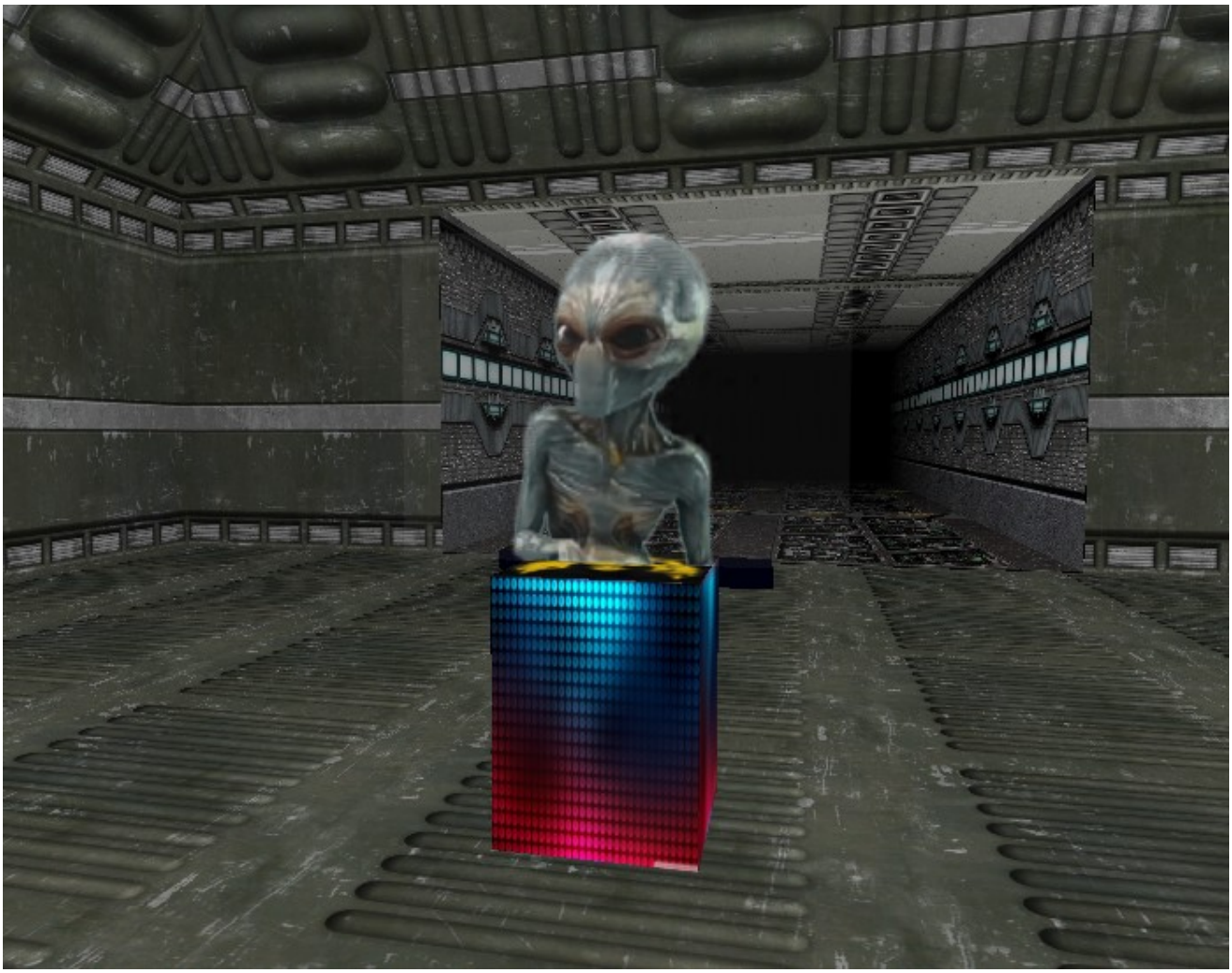

```python
import bge
from bge import texture
from bge import logic

cont = logic.getCurrentController()
obj = cont.owner # получаем ссылку на объект

if not hasattr(logic, 'video'): # делаем проверку атрибута video,
                  # чтобы убедиться, что мы создаем текстуру только один раз
    matID = texture.materialID(obj, 'IMTelevizor.png')
    # ищем материал по имени файла текстуры (Приставка IM обязательна)

    logic.video = texture.Texture(obj, matID)
    # присваиваем объект к атрибуту

    movie = logic.expandPath('//2.avi')
    # получаем полное имя файла видео

    logic.video.source = texture.VideoFFmpeg(movie)
    # Назначаем источник (у нас - это видео т.е.movie)

    logic.video.source.scale = True
    # устанавливаем автоматическое (быстрое) масштабирование

    logic.video.source.repeat = -1
    # устанавливаем бесконечное повторение

    logic.video.source.play()
    # запускаем видео

logic.video.refresh(True)
# вызываем очистку и повторение
```


Как видите, ничего сложного в скрипте нет. Каждая строка закомментирована, поэтому не должно возникнуть сложностей. Но если вы хотите более углублённо разобраться с кодом, то загляните в раздел **BGE и Python**.

Теперь нам осталось запустить игру и «пройтись» по комнате. Как обычно у меня управление мышью (правая кнопка мыши – вперёд). И вы увидите на экране терминала беспокойно оглядывающегося гуманоида ☺. (Смотри мой пример)

Естественно вы должны понимать, что видео с альфа-каналом возможно лишь в двух форматах: **AVI** и **MOV**. **AVI** для альфа должен быть несжатым. Соответственно размер такого файла может быть просто гигантским. Вывод: делать небольшие фрагменты малого разрешения, с расчётом на бесконечное повторение. Где взять видео файлы ? Отвечаю – на YouTube огромное количество файлов. Есть файлы на т.н. «Хромаке», т.е. на синем или зелёном фоне.

Программ, превращающих фон в альфа-канал довольно много. Есть платные и бесплатные. Но, в крайнем случае, простенький ролик вы можете сделать и на самом Blender. На этом знакомство с видео текстурами считаю законченным. В следующий раз мы рассмотрим, как ещё можно применить модуль **bge.texture** ☺.



14 мая 2016 года.

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