

Давайте рассмотрим ещё один скрипт на языке Python для объекта Text, который нам может понадобится. Попробуем вывести на экран линейную скорость объекта при его движении. Этот скрипт очень полезен в гоночных играх. Для начала создадим сцену, как в предыдущем уроке. Там у нас плоскость, объект-актёр, который управляется с клавиатуры стрелками и текстовый объект, прикрепленный к камере. А теперь переходим к скрипту. Для определения линейной скорости объекта в локальных координатах, у Blender Game Engine есть интересная функция – *localLinearVelocity*. Если перед ней поставить имя объекта, то будет измерена его линейная скорость. Результат измерения можно присвоить переменной:

```
vec=car.localLinearVelocity
```

Здесь *vec* это переменная, *car* это объект. Посмотрим весь скрипт:

```
import bge
cont=bge.logic.getCurrentController() # получить текущий контролер
obj = cont.owner # получить выделенный объект

scene = bge.logic.getCurrentScene() # получить текущую сцену
objList = scene.objects # получить список объектов
car = objList["Actor"] # получаем объект «актёр»
text = objList["Text"] # получаем текстовый объект

vec=car.localLinearVelocity # получить локальный вектор скорости
obj["prop"] = vec[1]/30 # присваиваем переменную Float и делим на 30

speed = obj["prop"] # присваиваем обычную переменную
```

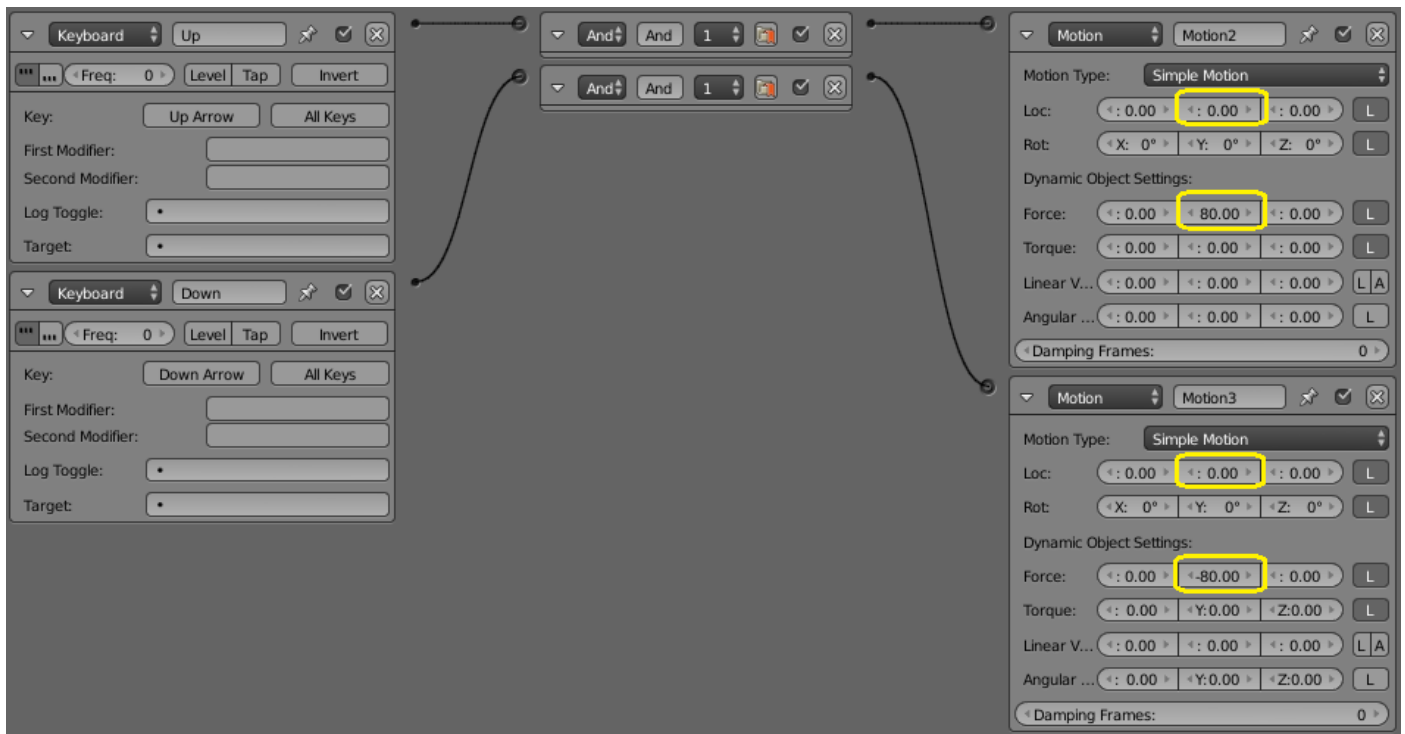
Для чего мы поделили на 30? Если получаемое число слишком велико, а нам нужно чтобы оно уложилось в определённый интервал. Вы можете разделить на 10 или вовсе не делить и посмотреть, что из этого получится. Полученный результат мы присваиваем обычной переменной *speed*. Далее, для вывода на экран в нормальном виде, нам нужно преобразовать результат в целое число (отбросив все цифры после запятой) и превратить цифровую переменную в строковую. Иначе объект Text её не поймёт. Но перед этим введём небольшое ограничение, чтобы корректно отображался ноль скорости:

```
if speed<-0.001: # если скорость меньше 0.001 то присваиваем 0
    speed = 0

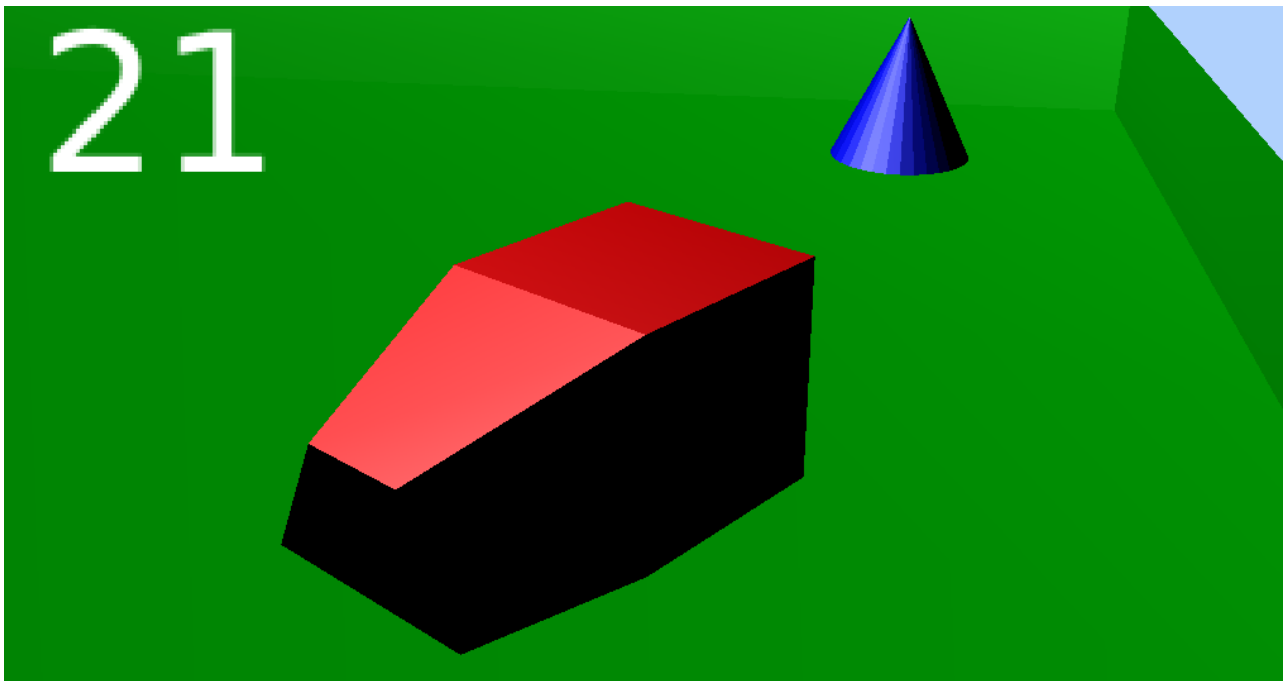
# переводим скорость в понятные цифры
speed = speed*100
speed = int(speed)

# конвертируем цифры в строку
text.text = str(speed) # выводим на экран
```

Однако жать на лат. «Р» ещё рано. В прошлых уроках нас вполне устраивало движение вперёд-назад на 1 по локальной оси Y. Но измерять скорость такого движения нет никакого смысла, т.к. скорость и так известна (она всегда равна 1). Поэтому мы изменим сам принцип движения. Мы больше не будем перемещать объект, а придадим ему импульс. Ведь перемещать объект, изменяя координату на единицу, можно и статический. Зачем же тогда мы делали его Rigid Body? Да потому, что к статическому объекту не приложишь силу импульса, и трение ему ни к чему. Сила применяется просто. Заходим в актуатор Motion (движения) сопоставленный клавише «Вперёд» и меняем значение Force оси Y, а значение Loc ставим ноль :



Тоже и назад. Вот теперь можно жать лат. «Р». Вы заметили, что характер движения изменился? При наборе скорости «Актёр» уже не так послушен – его заносит!? Дело в том, что теперь он обладает инерцией, как нормальный физический объект. А текстовый объект исправно показывает его скорость.



Ниже весь скрипт целиком:

```

1 import bge
2 cont=bge.logic.getCurrentController()
3 obj = cont.owner
4
5 scene = bge.logic.getCurrentScene()# получить текущую сцену
6 objList = scene.objects           # получить список объектов
7 car = objList["Actor"]             # получаем об машины
8 text = objList["Text"]            # получаем текстовый об
9
10 vec=car.localLinearVelocity        # получить локальный вектор скорости
11 obj["prop"] = vec[1]/30            # присваиваем переменную Float и делим на 30
12
13 speed = obj["prop"]                # присваиваем обычную переменную
14
15 if speed<-0.001:                  # если скорость меньше 0.001 то присваиваем 0
16     speed = 0
17
18 # переводим скорость в понятные цифры
19 speed = speed*100
20 speed = int(speed)
21
22 # конвертируем цифры в строку
23 text.text = str(speed)            # выводим на экран
24 |

```

Составил [Niburiec](http://blender-game.ucoz.ru) для сайта <http://blender-game.ucoz.ru>