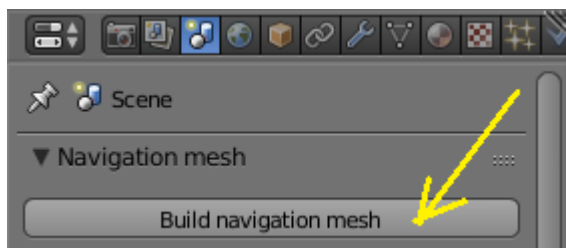
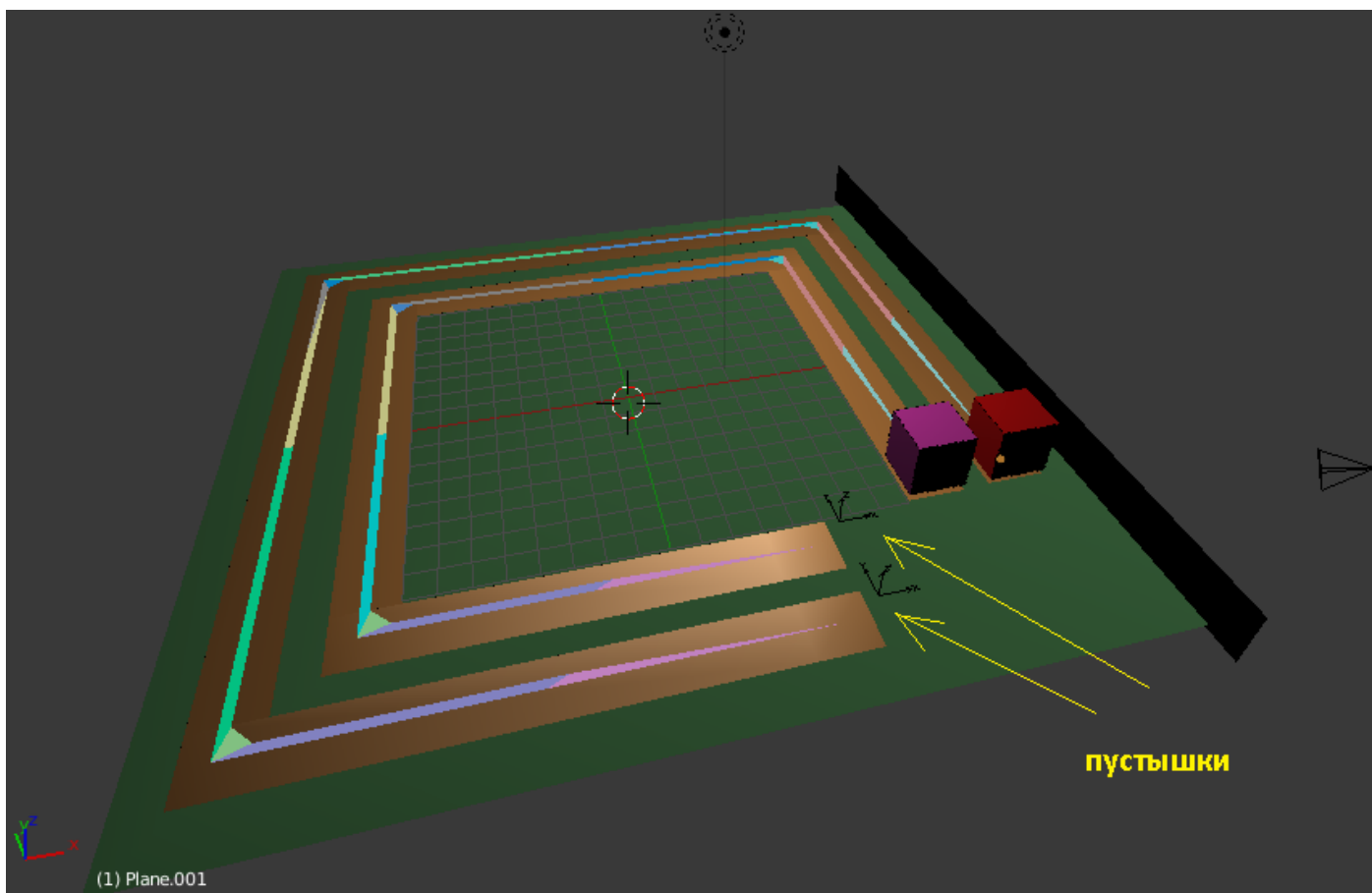


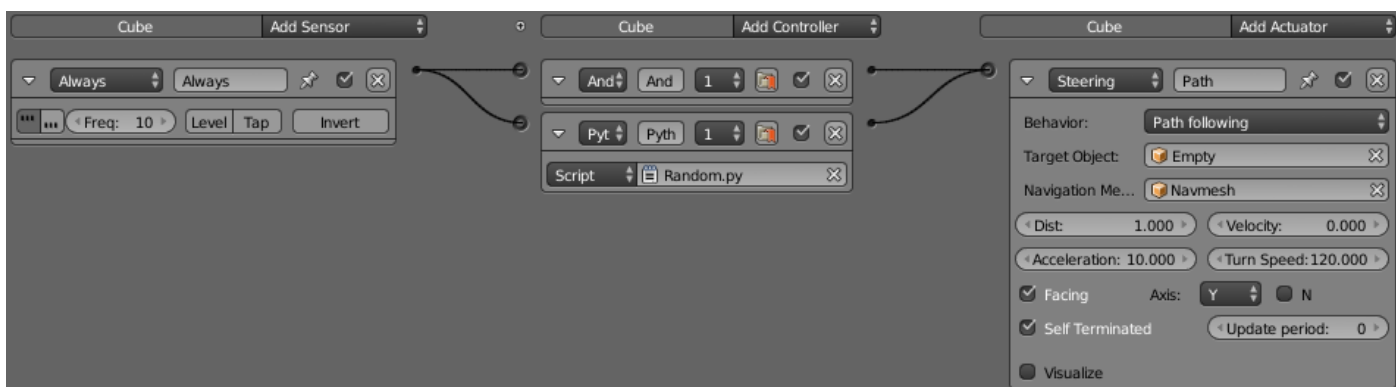
И так, разберём обещанные гонки кубиков. «Делайте ваши ставки, господа! Ибо неизвестно, какой из кубиков придет к финишу первым!» Здесь все просто. Создаем «землю». Два плана пути рисуем любым удобным вам способом и делаем (каждому отдельно) свой Navigation Mesh. Как это делается вы должны помнить из прошлых уроков:



Далее, вначале каждого пути ставим по кубику, а в конце – по точке-пустышке:



Переходим в логику. Выделяем первый кубик:



Для чего нужен контролер And? Дело в том, что в скрипте не прописано управление. Там лишь меняется значение Velocity в случайном порядке. Поэтому нужен контролер And, посылающий постоянные импульсы на актуатор. В поле Navigation Mesh выбираем тот путь, что под кубиком. Для второго кубика делаем тоже самое. Теперь пишем два одинаковых скрипта (но с разными названиями). Для первого:

```
import bge
import random # подключаем модуль случайных чисел
cont = bge.logic.getCurrentController() # Получить контролер

act = cont.actuators["Path"] # Получить путь

ran1 = random.random()*10 # Получить случайное число

act.velocity = int(ran1) # Присвоить значение скорости
```

И для второго:

```
import bge
import random # подключаем модуль случайных чисел
cont = bge.logic.getCurrentController() # Получить контролер

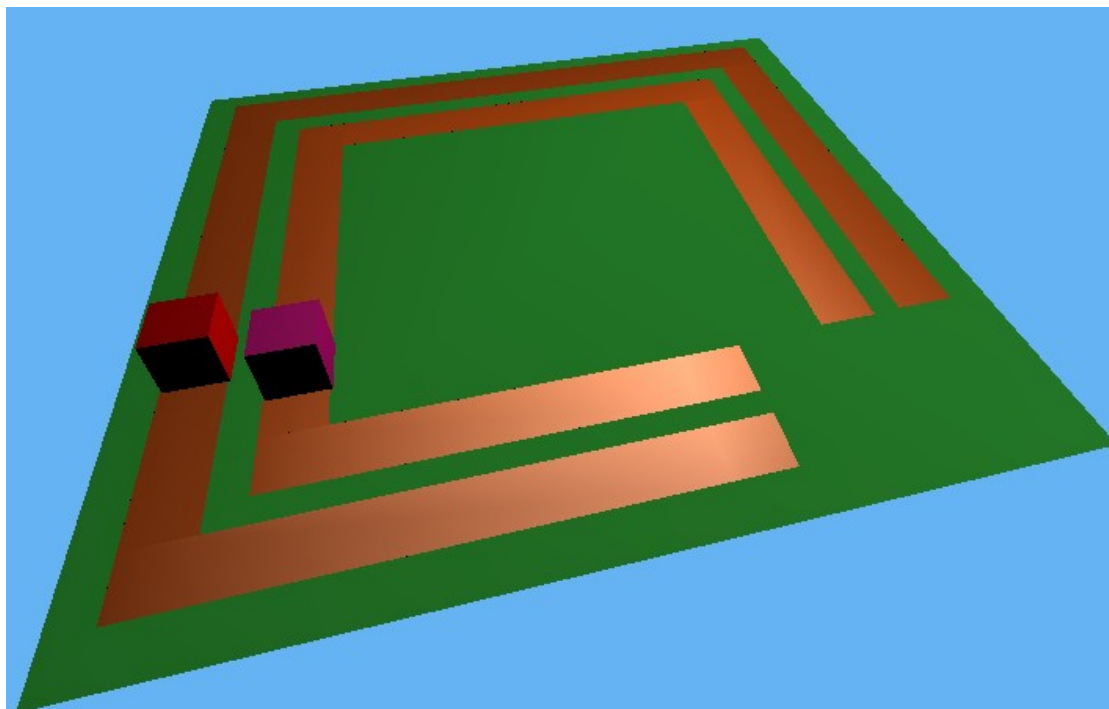
act = cont.actuators["Path1"]

ran2 = random.random()*12

act.velocity = int(ran2)
```

Во втором скрипте мы умножили не на 10 а на 12 не случайно. Просто внешний круг больше внутреннего, значит и изменения скорости нужны в более широких пределах.

Запускаем и смотрим:



Blend файл проекта как и всегда в разделе «Готовые проекты».

24.06.2015.

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