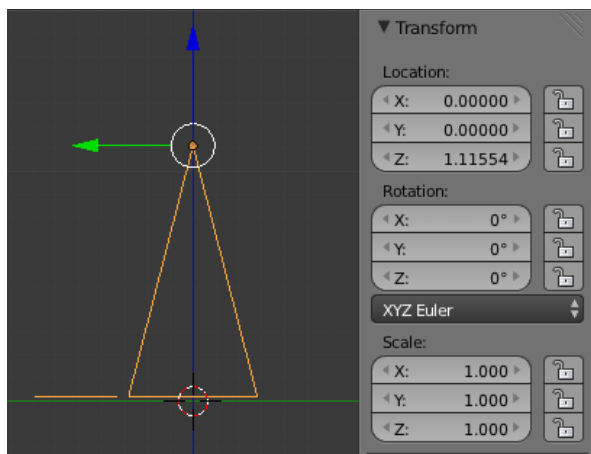


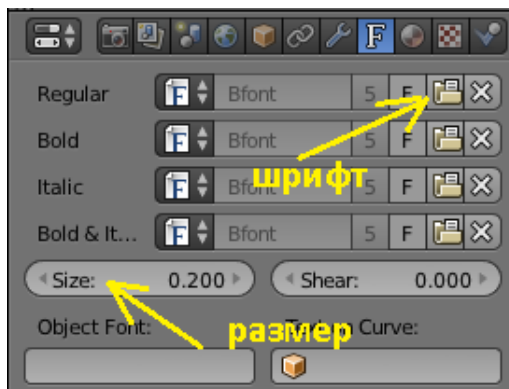
Большинство игр имеют возможность ввести имя игрока. Пришло время и нам с этим познакомиться. Реализовать в BGE ввод текста во время игры не просто. Без скриптов – практически не возможно. Поиски в интернете не дали серьёзных результатов. Оказалось, что по этому вопросу почти ничего нет. Перечитывая то немного, что удалось найти, я пробовал составить пример. Но знаний катастрофически не хватало. Тогда я обратился на форум одного русскоязычного сайта за помощью. Оказалось, что знающие люди есть, просто у них никто не спрашивал 😊. И совершенно бескорыстно получив подробные объяснения, я так же бескорыстно поделюсь с вами.

Не будем ставить задачу создать крутое меню. Давайте создадим заготовку-пример, к которому мы всегда сможем обратиться в последствии, если что-то забудем.

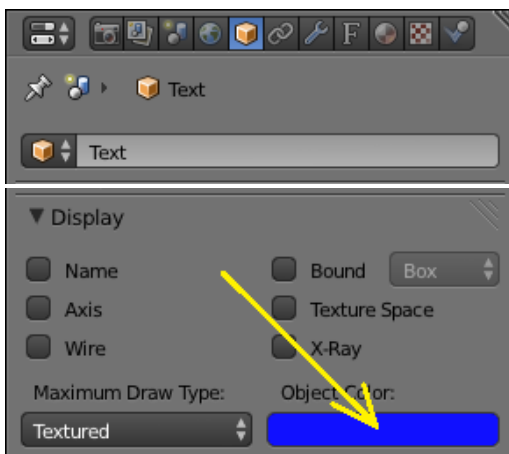
Для начала создадим две сцены. Можно назвать их как угодно, но я оставил по умолчанию. Удалим из них свет, камеры и куб(если он есть). Вставим в обе сцены камеры с нулевой ориентацией и приподнимем над центром:



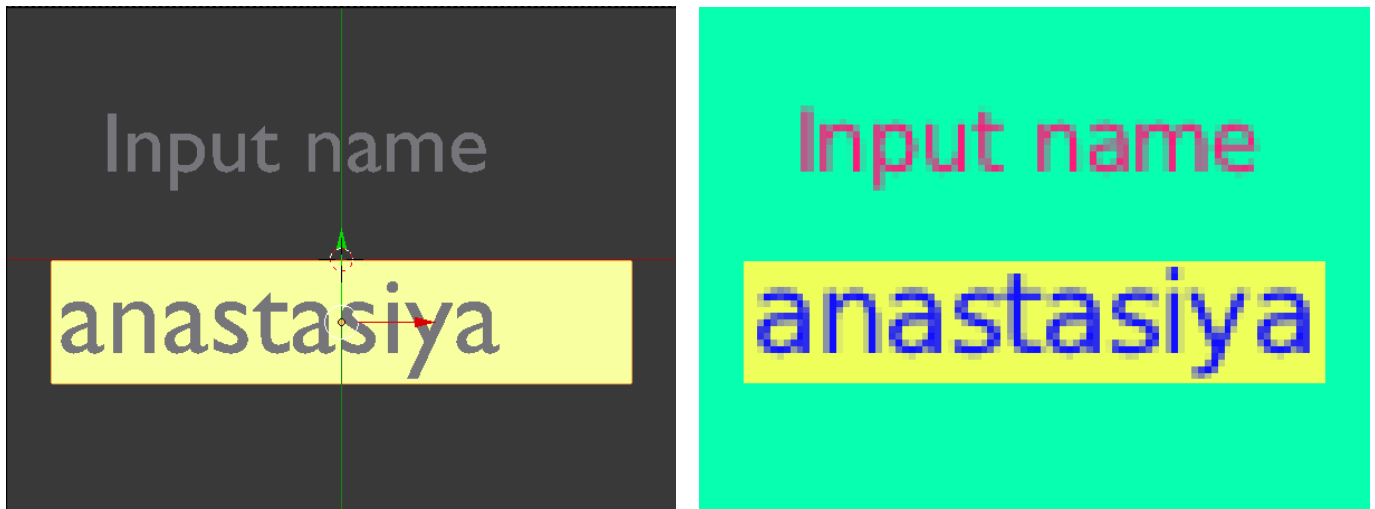
Включаем режим Blender Game и в первой сцене переходим на вид с камеры. Вставим два текстовых объекта, верхний из которых назовём `Input_Text`. Нижний оставим с названием `Text`. Настроим размер обоих текстов на вкладке Text и выберем шрифт:



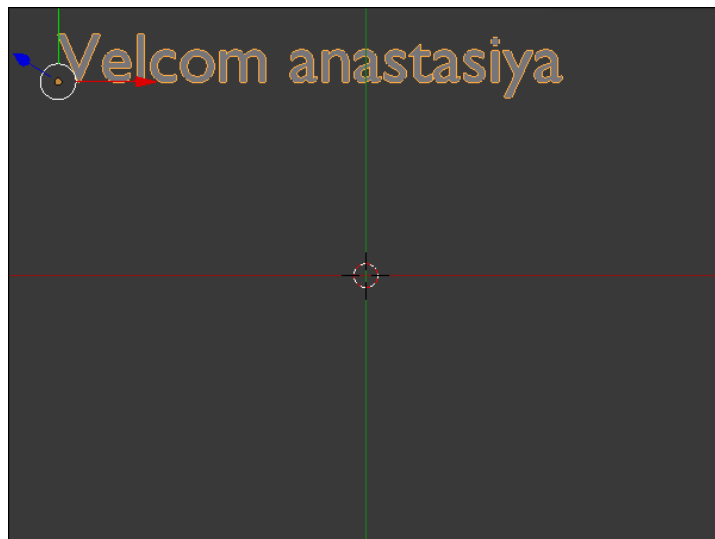
И цвет текста, на вкладке Object:



Попробуйте включить игру (лат. «Р»), чтобы проверить правильность размещения объектов. Иногда такие малые размеры немного не совпадают с реальностью. Теперь, за нижним текстовым объектом разместите плоскость и тоже дайте ей цветной материал. Не волнуйтесь за качество букв. Позже в скрипте мы его улучшим. Ну, и цвет фона мира установите. Обязательно замените буквы в нижнем тексте так, чтобы их было ровно 10. Так гораздо легче выравнивать объекты относительно друг друга:



Во второй сцене так же настроим цвет мира и вставим текстовый объект с именем `Text.002`. Должно получиться примерно так:



Если буквы выглядят уж очень плохо, попробуйте поиграть с размером и расстоянием от камеры. Найдите для себя оптимальный вариант. Чем дальше от камеры, тем чётче буквы.

Переходим в логику первой сцены. Выделяем верхний текст и пишем малюсенький скрипт, с названием `Pervy`:

```
#-*- coding: utf-8 -*-
```

```
import bge
```

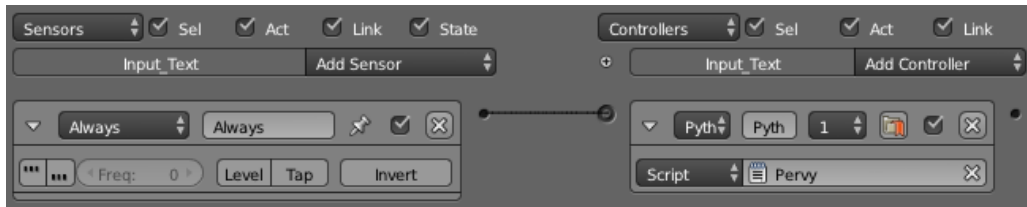
```
scene = bge.logic.getCurrentScene()# получить текущую сцену
```

```
objList = scene.objects # получить список объектов
```

```
text = objList["Input_Text"] # получаем текстовый объект
```

```
text.resolution = 2 # улучшаем текст
```

Вставим сенсор и контролёр:



Теперь выделяем нижний текст. Включив не на долго режим редактирования удалим все буквы:



Вставим в логике сенсор Keyboard (в котором выберем All Keys), контролёр Python и актуатор Scene:



И пишем скрипт для контролёра Python нижнего текста, с названием Vtoroy:

```
# -*- coding: utf-8 -*-
```

```
import bge
```

```
scene = bge.logic.getCurrentScene()# получить текущую сцену
```

```
objList = scene.objects # получить список объектов
```

```
# получить контролёр
```

```
cont = bge.logic.getCurrentController()
```

```
scen = cont.actuators["Scene"] # Получить актуатор
```

```
obj = cont.owner # Получить текущий объект
```

```
text = objList["Text"] # получаем текстовый объект
```

```
text.size = 0.17 # меняем размер текста, если нужно
```

```
#text.color = 0, 0.9, 0.5, 1 # меняем цвет текста, если нужно
```

```
text.resolution = 2 # улучшаем текст
```

```
# получить сенсор по имени
```

```
sen = cont.sensors["Keyboard"]
```

```
# получить сканкод нажатой клавиши
```

```
es = sen.events[0] # первое событие в очереди - сканкод клавиши
```

```
#es[0] - это код клавиши, es[1] - статус клавиши (1- нажата, 3-отпущена)
```

```
if es[1]==1: # если клавиша нажата
```

```
keyCode = es[0] # то получить код этой клавиши, при этом
```


```
if keyCode == 133: # если это клавиша Backspace
```

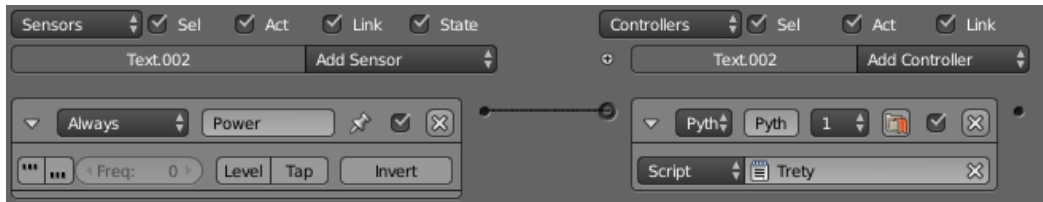
```
text.text = text.text[:len(text.text)-1] # то уменьшаем длину текста на единицу
```

```

elif keyCode == 13: # если это клавиша Enter
    bge.logic.globalDict['gamename'] = text.text # передаём текст в глобальную переменную
    cont.activate(scen) # переходим в следующую сцену
    pass # (пропуск)
else: # иначе
    if len(text.text)<10: # если длина текста меньше 10 знаков
        text.text += chr(keyCode) # переводим сканкод в символ и печатаем

```

Объяснять значение каждой строки нет смысла, ибо в коде я всё написал в комментариях. Переходим во вторую сцену. Выделяем текстовый объект, и в режиме редактирования  удаляем все буквы. Возвратившись в режим объекта переходим в логику, где вставляем сенсор Always и контролёр Python:



Создаём ещё один скрипт для контролёра текстового объекта Text.002 под названием Trety:

```

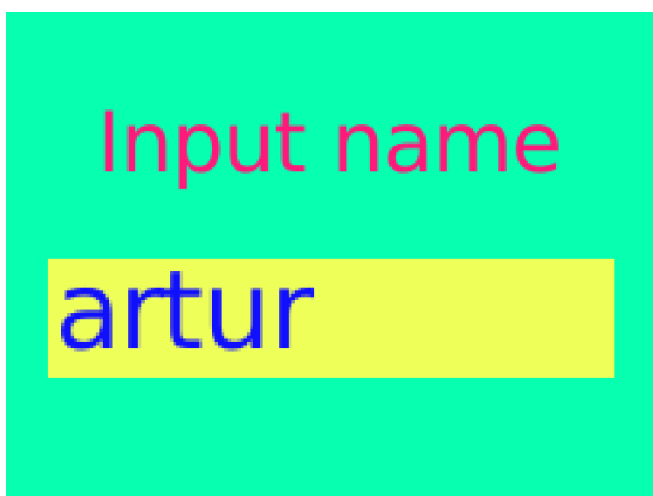
import bge
cont = bge.logic.getCurrentController() # получить контролёр
sen = cont.sensors["Power"] # получить сенсор

scene = bge.logic.getCurrentScene()# получить текущую сцену
objList = scene.objects # получить список объектов
text2 = objList["Text.002"] # получить текстовый объект

# Читаем данные из глобальной переменной и присваиваем их тексту
text2.text = bge.logic.globalDict['gamename']

```

Вот и всё. На первый взгляд кажется сложно. Зато мы научились получать сканкод клавиатуры и создали глобальную переменную в словаре. Он позволяет хранить любые данные и получать их в любой сцене. Переходим в первую сцену, запускаем (лат «Р») и вводим имя (латинскими буквами). Жмём Enter и радуемся, наблюдая имя во второй сцене ☺.



Scene



Scene.001

А что же делать, если нам необходимо вводить буквы русского алфавита? Ведь в данном примере при любом раскладе клавиатуры мы вводим только латинские буквы? Ну, тогда есть два пути. Первый путь это создать собственный шрифт, заменив латинские буквы на русские, чтобы получился тот же сканкод. Зачем нам лишние проблемы 😊? Второй путь гораздо проще – программно, прямо в коде, назначить латинским буквам русские. Великолепный Python и здесь нас может выручить.

Для начала нам необходимо узнать сканкоды латинской раскладки на все клавиши, связанные с русским алфавитом. Для этого можно посмотреть официальную справку по Blender или воспользоваться нашей (только что написанной) программой. Изменить нужно всего одну строчку кода в самом конце скрипта Vtoroy:

```
text.text += chr(keyCode)
```

Где вместо `chr` поставить `str`. Теперь вводим по одному символу и смотрим цифровой сканкод, который необходимо записать в соответствии с буквами. Например, сканкод латинской буквы `Z` – 122. Но эта клавиша соответствует русской букве `Я`. Так и запишем для себя `Я – 122`, и т.д. Таким образом, у нас получится таблица соответствия кодов, которую мы и применим в расширенном варианте скрипта. После строки:

```
if len(text.text)<10:      # если длина текста меньше 10 знаков
```

Вставляем кусок кода:

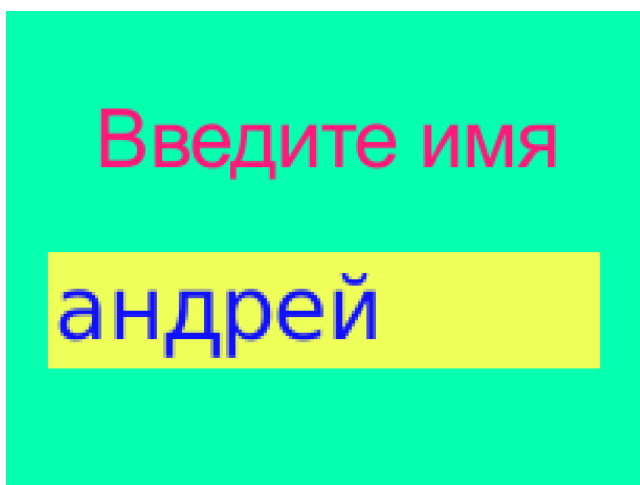
```
scan = keyCode # Назначаем число сканкода переменной
```

```
    if scan == 48:
        text.text += "0"
    if scan == 49:
        text.text += "1"
    if scan == 50:
        text.text += "2"
    if scan == 51:
        text.text += "3"
    if scan == 52:
        text.text += "4"
    if scan == 53:
        text.text += "5"
    if scan == 54:
        text.text += "6"
    if scan == 55:
        text.text += "7"
    if scan == 56:
        text.text += "8"
    if scan == 57:
        text.text += "9"
    if scan == 137:
        text.text += "ё"
    if scan == 113:
        text.text += "й"
    if scan == 119:
        text.text += "ц"
    if scan == 101:
        text.text += "у"
```

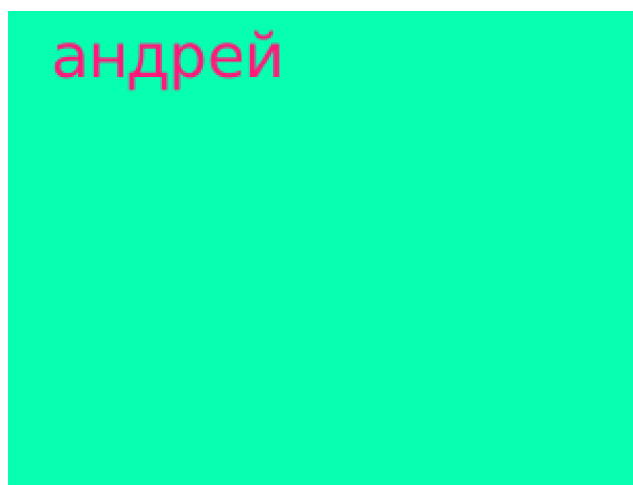
```
if scan == 114:
    text.text += "к"
if scan == 116:
    text.text += "е"
if scan == 121:
    text.text += "н"
if scan == 117:
    text.text += "р"
if scan == 105:
    text.text += "ш"
if scan == 111:
    text.text += "щ"
if scan == 112:
    text.text += "з"
if scan == 141:
    text.text += "х"
if scan == 142:
    text.text += "ь"
if scan == 97:
    text.text += "ф"
if scan == 115:
    text.text += "ы"
if scan == 100:
    text.text += "в"
if scan == 102:
    text.text += "а"
if scan == 103:
    text.text += "п"
if scan == 104:
    text.text += "р"
if scan == 106:
    text.text += "о"
if scan == 107:
    text.text += "л"
if scan == 108:
    text.text += "д"
if scan == 135:
    text.text += "ж"
if scan == 136:
    text.text += "э"
if scan == 122:
    text.text += "я"
if scan == 120:
    text.text += "ч"
if scan == 99:
    text.text += "с"
if scan == 118:
    text.text += "м"
if scan == 98:
```

```
text.text += "и"  
if scan == 110:  
    text.text += "т"  
if scan == 109:  
    text.text += "ь"  
if scan == 44:  
    text.text += "б"  
if scan == 46:  
    text.text += "ю"  
if scan == 32:  
    text.text += " "
```

Многовато, примитивно – я понимаю. Можно было всё собрать в список (массив) и упростить. Но мы ведь с вами «начинающие». А этот вариант проще всего и нагляднее. Пользуйтесь! ☺ И изучайте Python! Возможно, что скоро вы сделаете гораздо лучше ☺.



Scene



Scene.001

Составил Niburiec 27.01.2015. для сайта <http://blender-game.ucoz.ru/>