Разработка пользовательского интерфейса в Unity

Ashley Godbold. **Mastering UI Development with Unity:** An in-depth guide to developing engaging user interfaces with Unity 5, Unity 2017, and Unity 2018

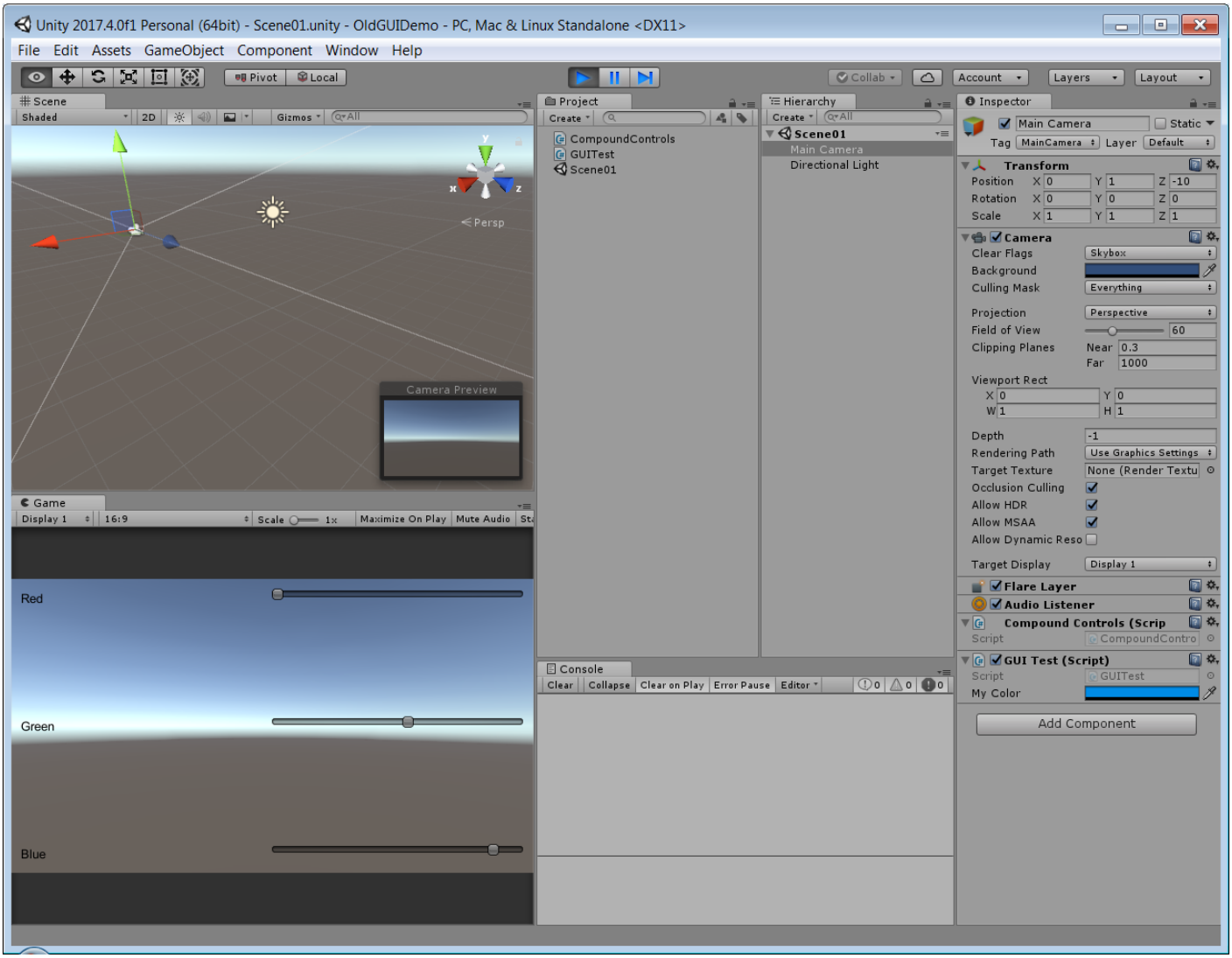
Версии GUI в Unity

До версии 4.6 в Unity использовалась система GUI, ориентированная исключительно на программную реализацию. Сейчас она называется «устаревшей GUI» (legacy GUI), тогда как новая система называется просто UI.

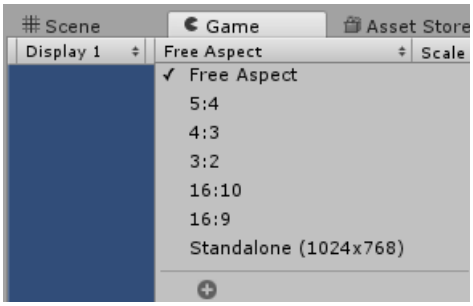
Пример использования устаревшей GUI:

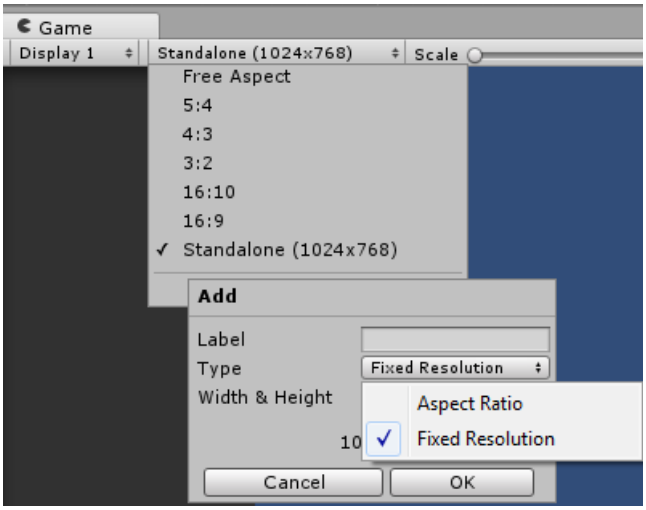
```
using UnityEngine;
using System.Collections;

public class CompoundControls : MonoBehaviour
{
    public static float LabelSlider (Rect screenRect,
        float sliderValue, float sliderMaxValue, string labelText)
    {
        GUI.contentColor = Color.black;
        GUI.Label(screenRect, labelText);
        screenRect.x += screenRect.width;
        sliderValue = GUI.HorizontalSlider(screenRect, sliderValue, 0.0f, sliderMaxValue);
        return sliderValue;
    }
}

using UnityEngine;
using System.Collections;
public class GUITest : MonoBehaviour
{
    public Color myColor;

    void OnGUI ()
    {
        myColor = RGBSlider(new Rect(10, 10, Screen.width / 2 - 10,
          Screen.height / 2 - 20), myColor);
    }

    Color RGBSlider(Rect screenRect, Color rgb)
    {
        rgb.r = CompoundControls.LabelSlider(screenRect, rgb.r, 1.0f, "Red");
        screenRect.y += screenRect.height;
        rgb.g = CompoundControls.LabelSlider(screenRect, rgb.g, 1.0f, "Green");
        screenRect.y += screenRect.height;
        rgb.b = CompoundControls.LabelSlider(screenRect, rgb.b, 1.0f, "Blue");
        return rgb;
    }
}
```

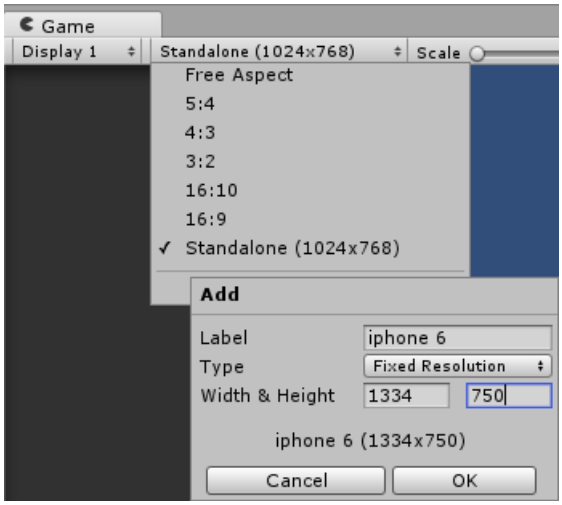
2



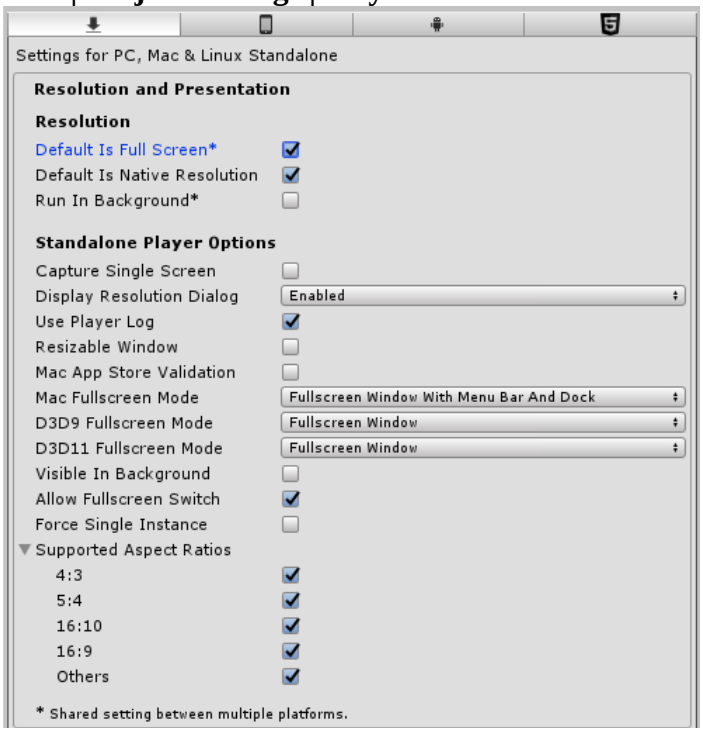
Настройка соотношения сторон и разрешения окна

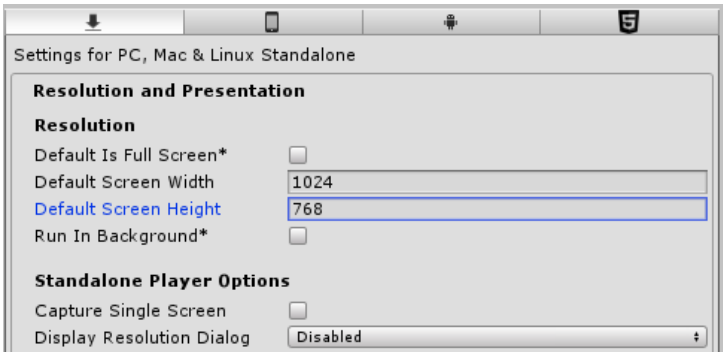






Edit | Project Settings | Player



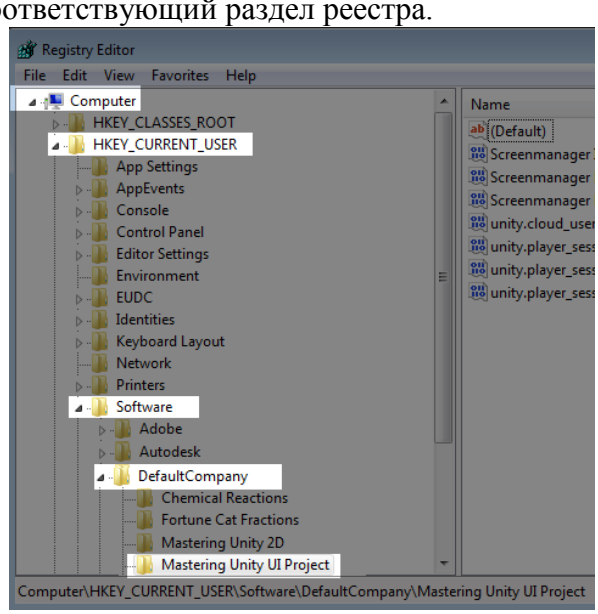


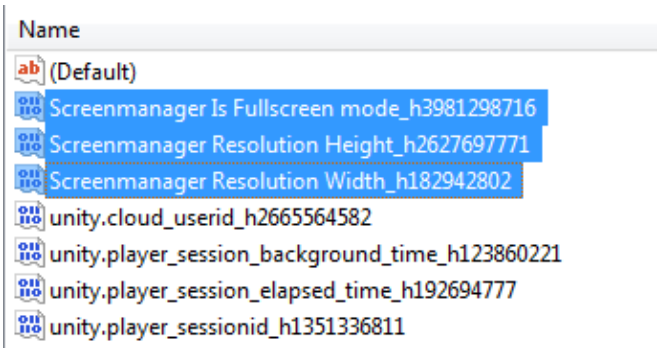
Если снять флажок Default is Native Resolution, то можно настроить разрешение по умолчанию.

Если откомпилировать и запустить программу в Unity 2017, а затем изменить настройки, связанные с разрешением, то они не будут учтены при последующем запуске, так как исходные настройки сохраняются в реестре. Это можно исправить с помощью следующего скрипта (данный скрипт должен быть размещен в подкаталоге Editor каталога Assets):

```
using UnityEditor;
using UnityEngine;
public class DeletePrefs : EditorWindow
{
  [MenuItem("Edit/Delete All PlayerPrefs")]
  public static void DeletePlayerPrefs()
  {
    PlayerPrefs.DeleteAll();
    Debug.Log("delete prefs");
  }
}
```

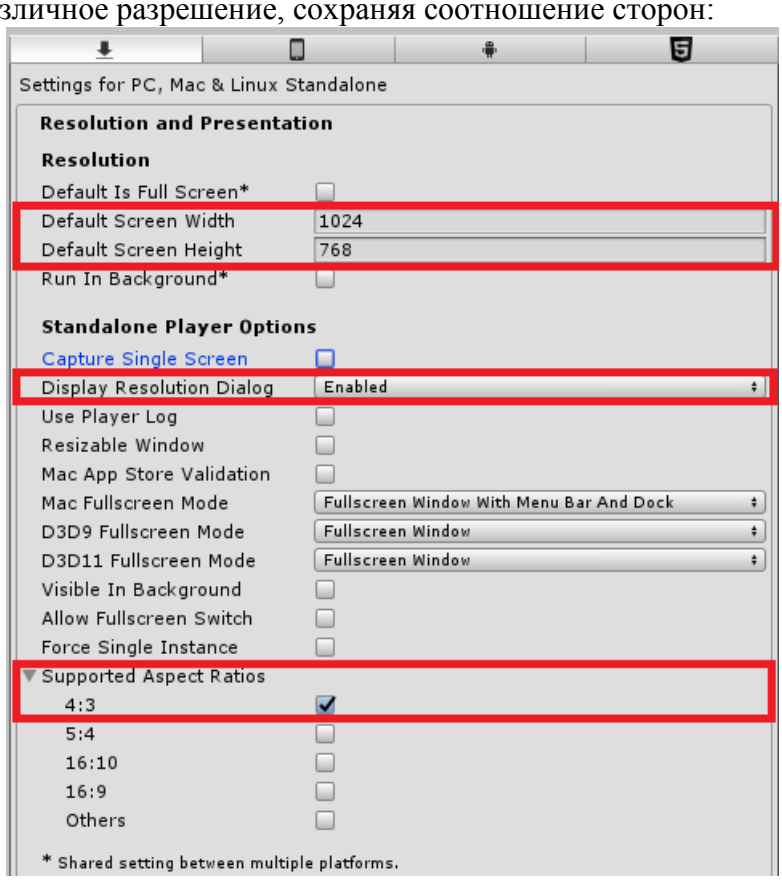
Сразу после создания данного скрипта в меню Edit будет добавлена указанная команда. Иногда выполнение этой команды не приводит к требуемому результату. Тогда надо явно очистить соответствующий раздел реестра.

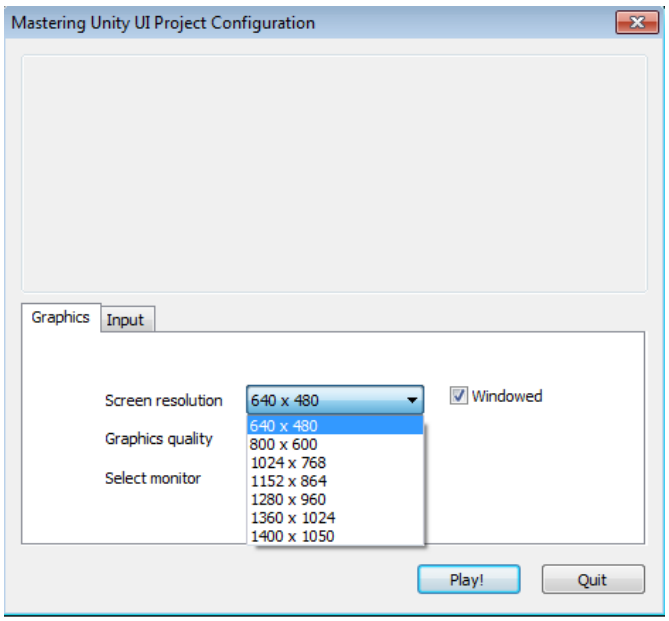




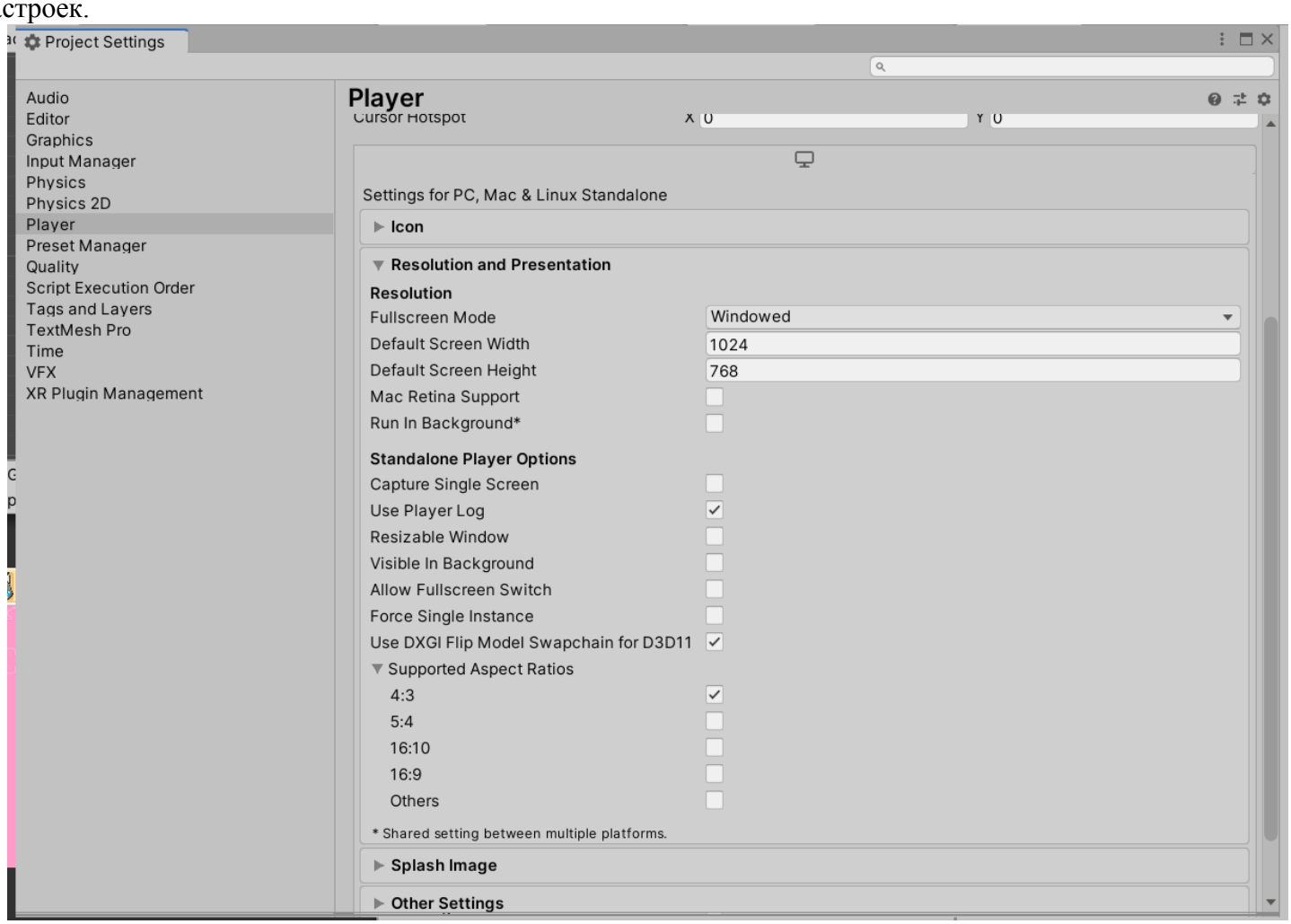
В Unity 2019 отмеченная проблема решена: при изменении стартового разрешения оно учитывается при очередном запуске программы.

В Unity 2017 можно настроить параметры запуска программы так, чтобы можно было выбирать различное разрешение, сохраняя соотношение сторон:



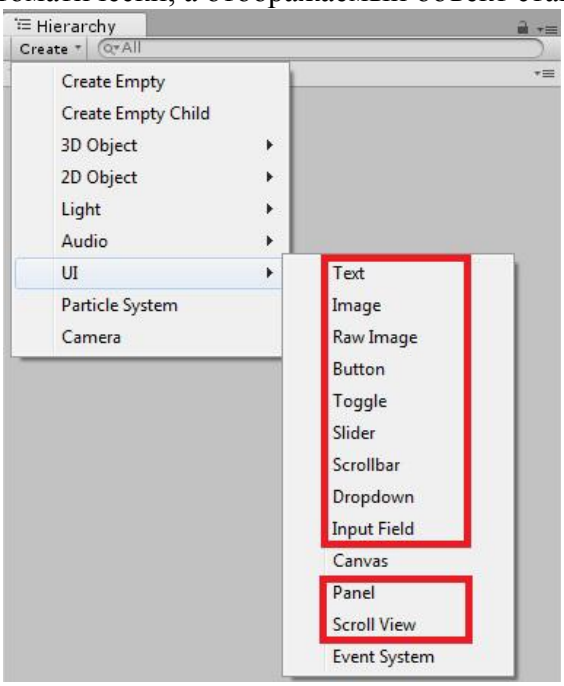


В версии 2019.1 опция Display Resolution Dialog объявлена устаревшей, а в версии 2019.3 она удалена из настроек.

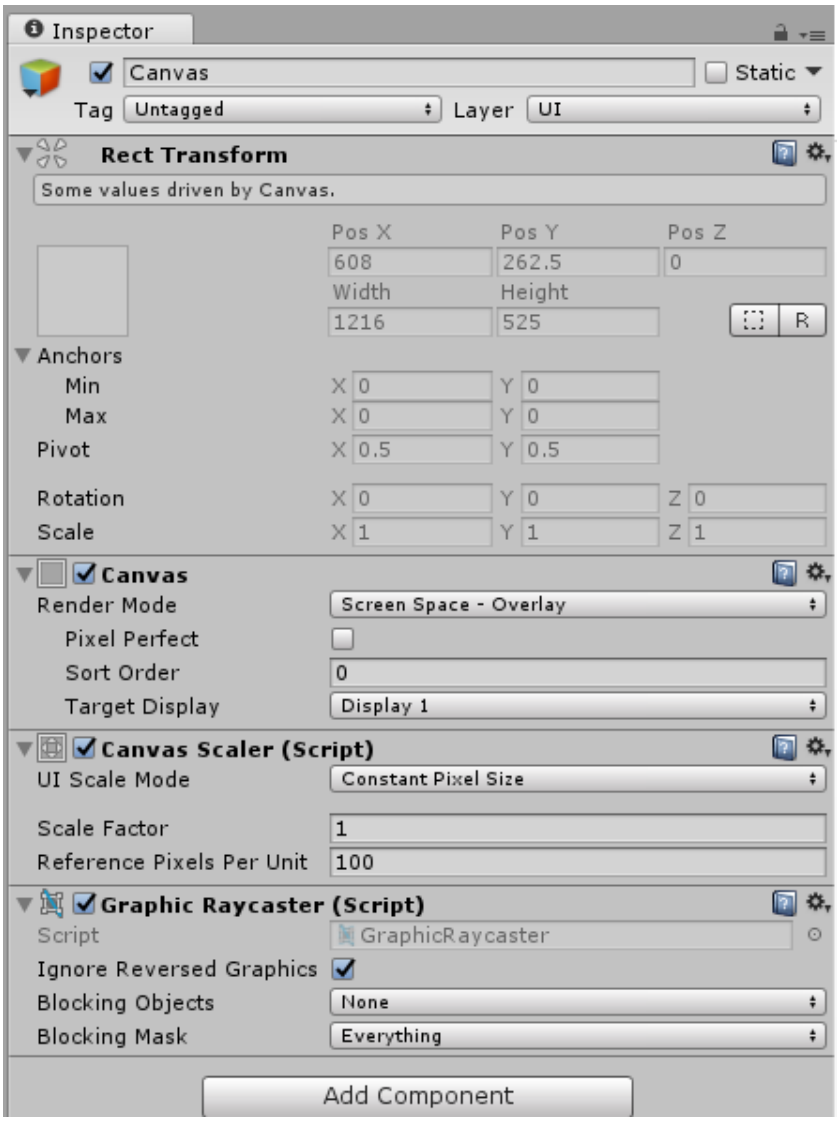


Объект Canvas

Любой отображаемый UI-объект (в красной рамке) должен находиться на объекте Canvas (полотно, канва), чтобы он мог быть изображен на сцене. При попытке добавления отображаемого UI-объекта на сцену, не содержащую Canvas, объект Canvas (а также объект EventSystem) будет добавлен к сцене автоматически, а отображаемый объект станет дочерним объектом объекта Canvas.

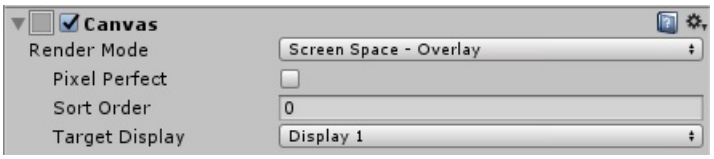


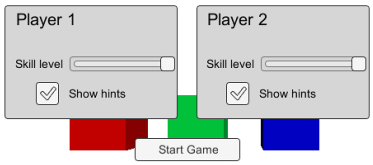
Компоненты объекта Canvas



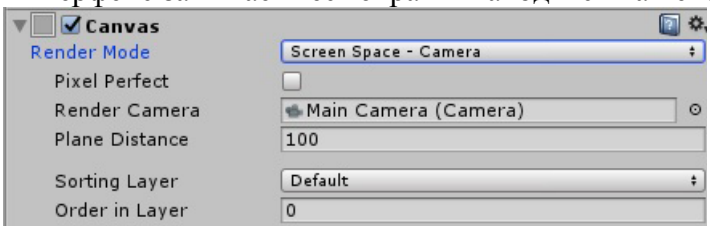
Варианты отрисовки канвы (Canvas Render Mode)

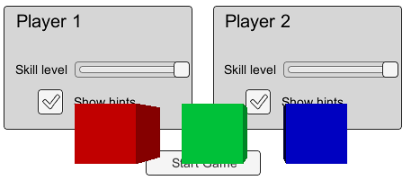
Интерфейс занимает весь экран, не зависит от камеры и рисуется поверх всех игровых объектов:



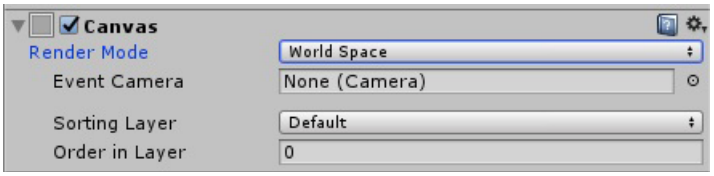


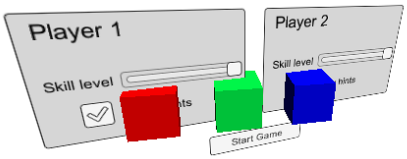
Интерфейс занимает весь экран и находится на некотором расстоянии от выбранной камеры:





Интерфейс — плоский объект сцены (вариант реализации *диегетического* интерфейса):

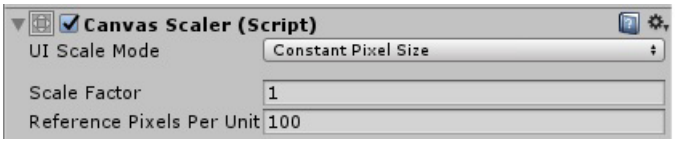


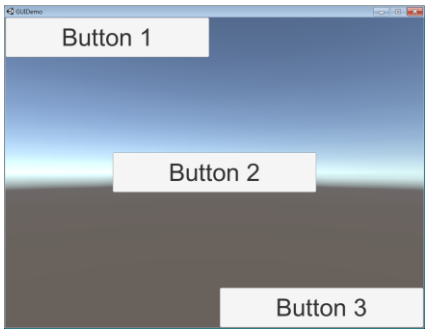


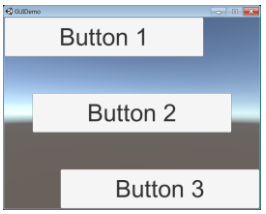
В этом режиме указывается Event Camera (не Render Camera), поскольку с точки зрения отрисовки канва не отличается от других игровых объектов. Event Camera отвечает за прием событий от UI-объектов и связана с объектом EventSystem.

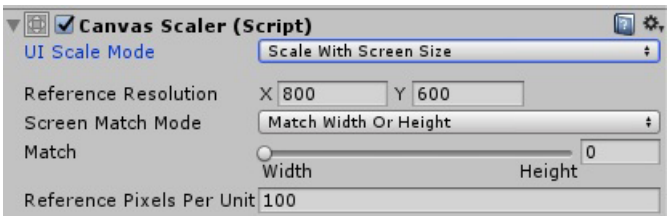
Компонент Canvas Scaler

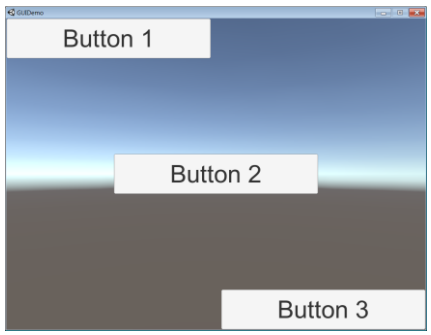
Определяет, как меняется вид UI-элементов в зависимости от разрешения (в примерах — 1024x768 и 640x480).

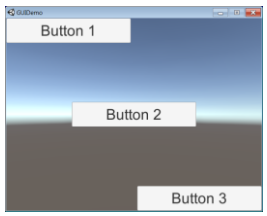


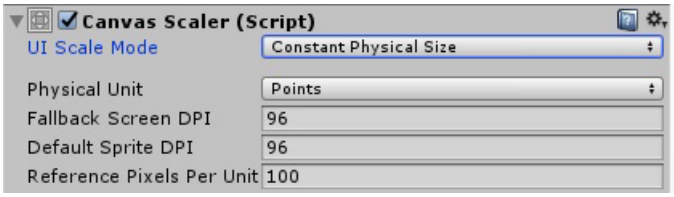


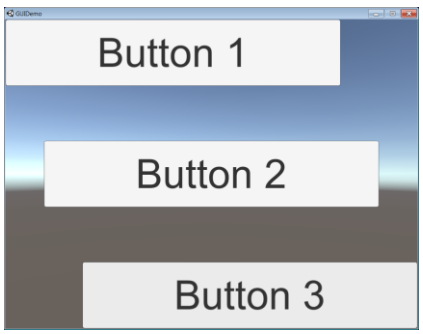


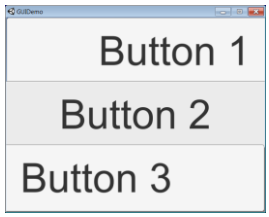








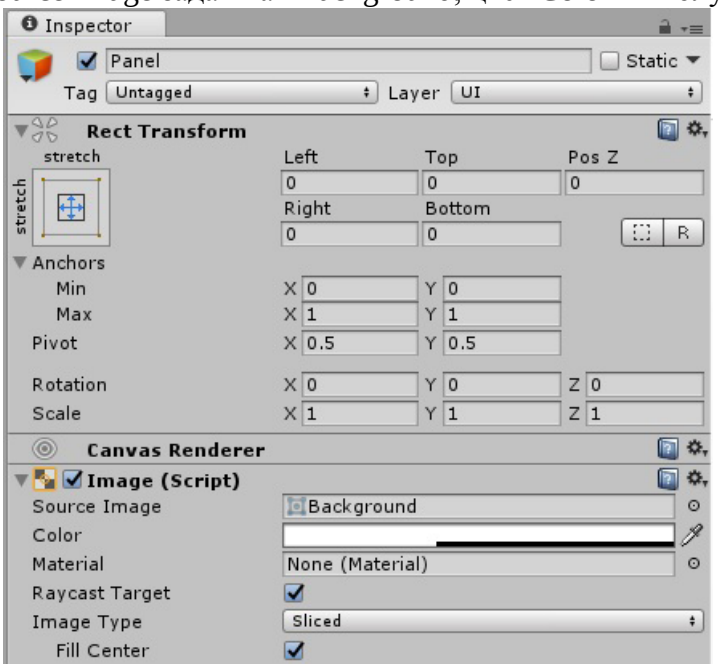




Вариант **World** — это единственный вариант, доступный для третьего режима отображения канвы.

Объект Panel

Назначение панели — содержать другие UI-объекты. На самом деле *компонента* Panel нет, панель — это игровой объект, снабженный компонентом Image с несколько измененными свойствами по умолчанию (Source Image задан как Background, цвет Color — полупрозрачный).

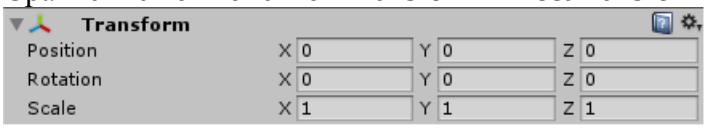


Компонент Rect Transform

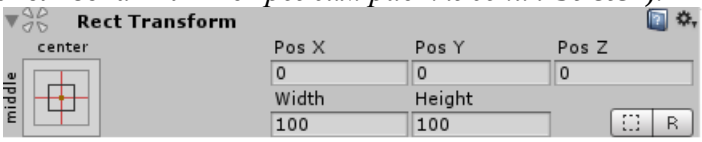
UI-объекты имеют данный компонент вместо компонента Transform (связанного с трехмерными и двумерными игровыми объектами). Заметим, что класс RectTransform является потомком класса Transform.

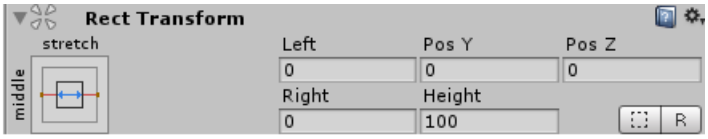
Для манипулирования двумерными объектами (в том числе UI-объектами) предназначен инструмент Rect Tool:

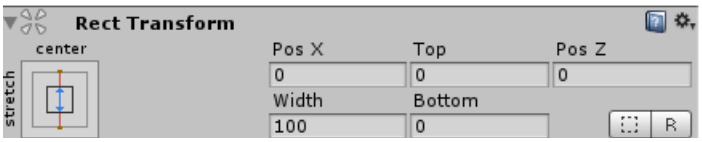
Сравнение компонентов Transform и RectTransform

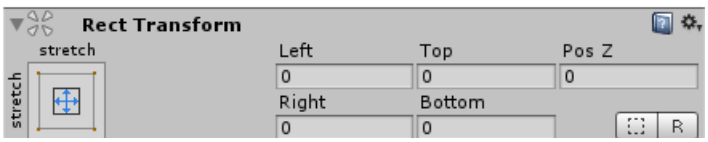


Особенности отображения числовых данные в зависимости от режима *настроек привязки* Anchor Preset (использован или нет *режим растяжения* Stretch):



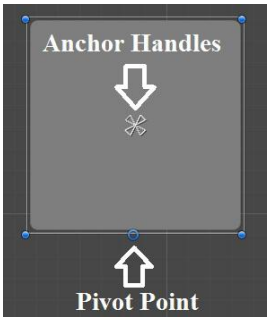


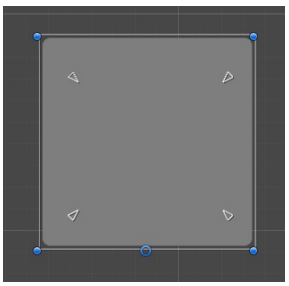
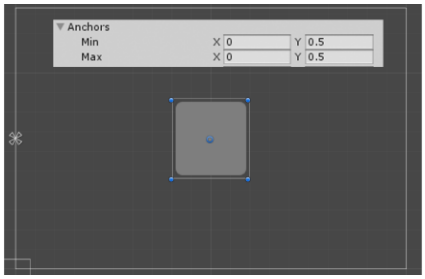


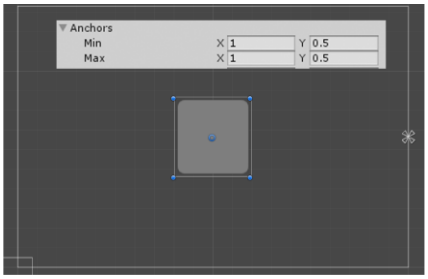


Значения (PosX, PosY, PosZ) определяют *опорную точку* (Pivot Point), значения Left, Right, Top, Bottom определяют отступы от соответствующих границ в режиме растяжения по данному измерению.

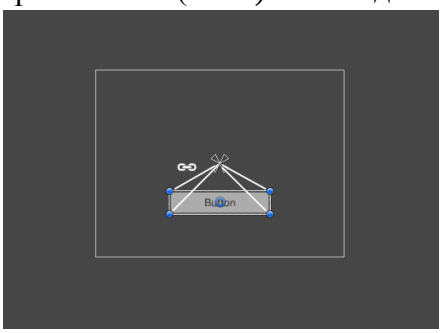
Опорная точка (Pivot Point) и позиции привязки (Anchors)

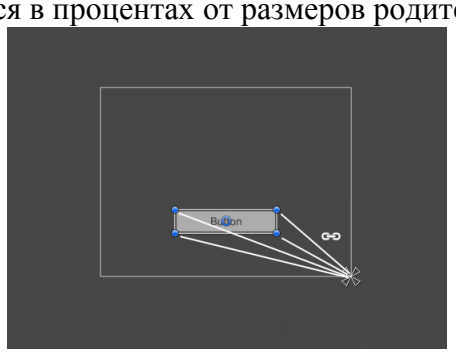


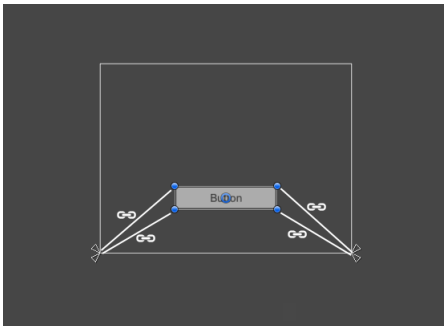



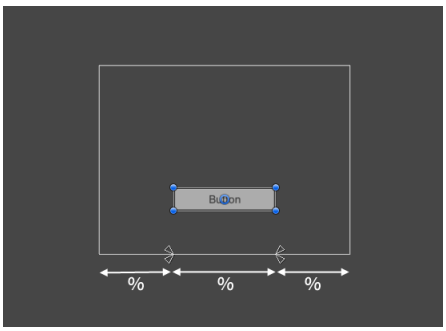


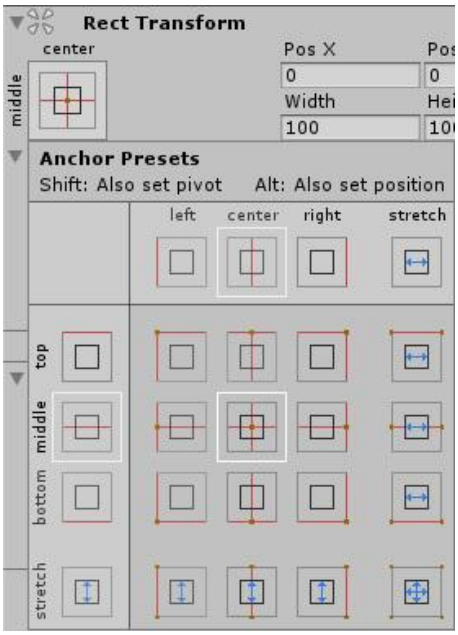
Позиции привязки всегда расположены в вершинах прямоугольника (возможно, вырожденного). Эти позиции задаются значениями Anchors Min и Max в процентах от размеров родительского объекта. Позиция опорной точки (Pivot) тоже задается в процентах от размеров родительского объекта.





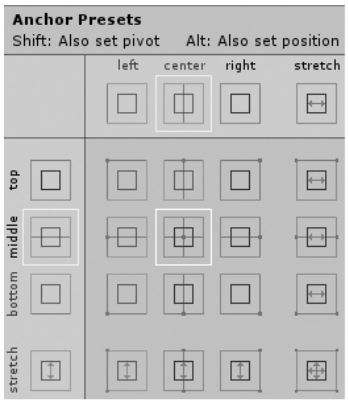


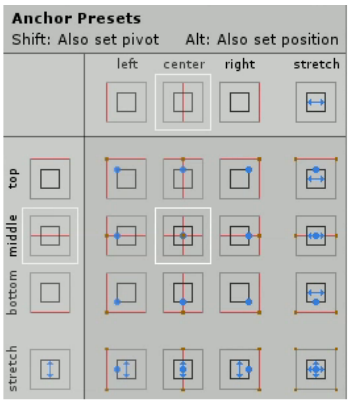


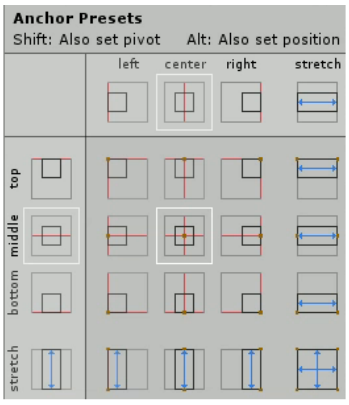


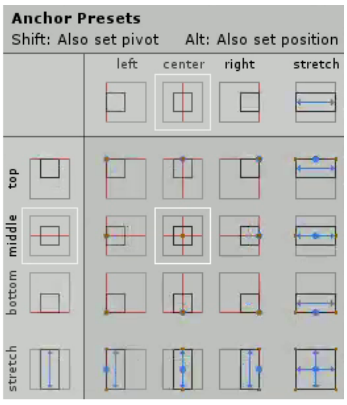
Настройки привязки (Anchor Presets)

Важная возможность: если при задании настроек привязки держать нажатыми клавиши Shift и/или Alt, то кроме позиций привязки можно сразу настроить позицию опорной точки и/или положение объекта.









Shift Alt Shift+Alt

Компонент Canvas Group

Add Component | Layout | Canvas Group

Позволяет настраивать ряд важных свойств сразу для игрового объекта и всех его дочерних объектов.

Alpha — настройка прозрачности (1 — полная непрозрачность);

Interactable — возможность реагировать на события;

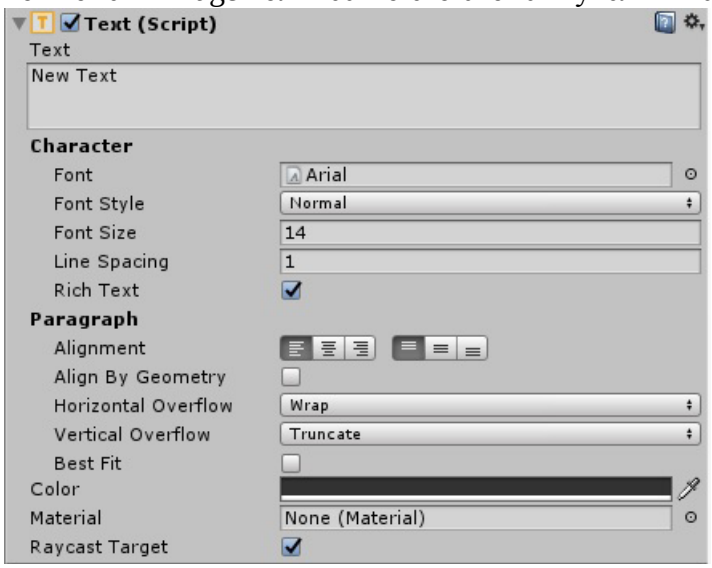
Blocks Raycasts — лучи не доходят до объектов, расположенных ниже (и, соответственно, эти объекты не могут реагировать на события);

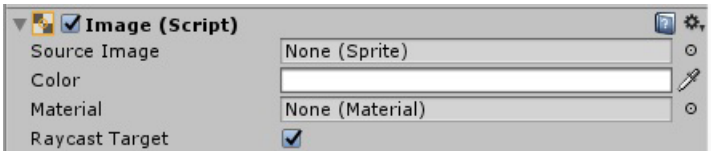
Ignore Parent Groups — если установлено, то настройки компонента Canvas Group родителя не учитываются.

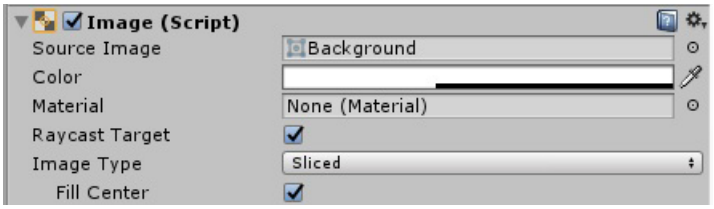
Объекты Text и Image

Create | UI | Text или Image

Это UI-объекты, к которым присоединены компоненты Text и Image. При добавления изображения в компонент Image количество его свойств увеличится (ср. с компонентом Image для объекта Panel).



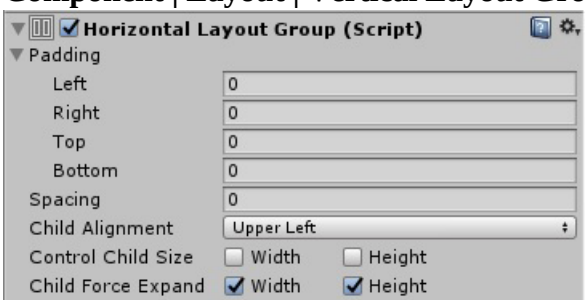


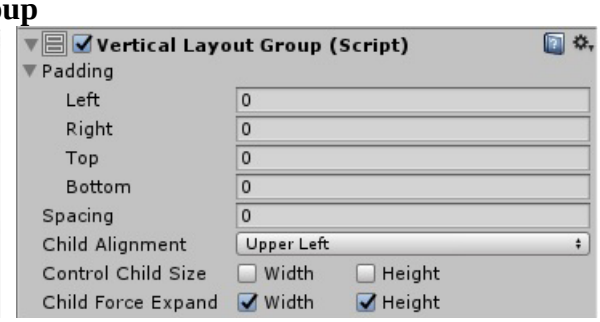


Средства автоматической компоновки (Layout)

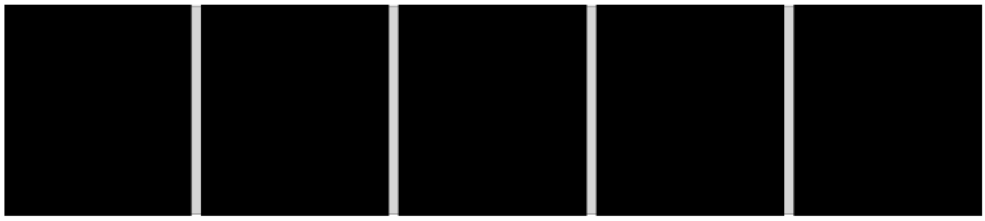
Компоненты Horizontal Layout Group и Vertical Layout Group

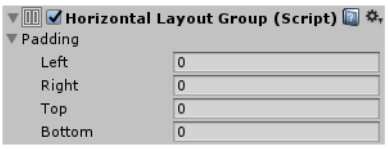
Component | Layout | Horizontal Layout Group
Component | Layout | Vertical Layout Group

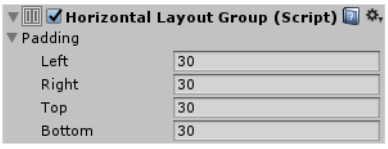


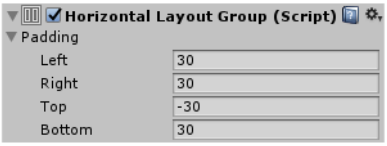


Padding (в этих примерах установлены свойства Control Child Size и Child Force Expand):

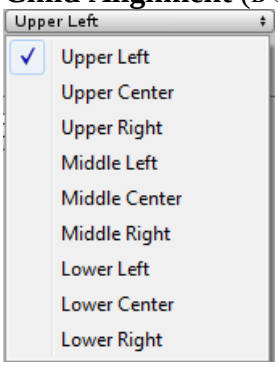


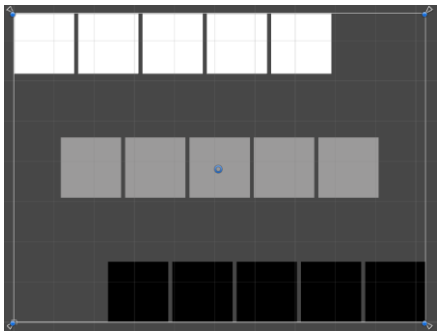






Spacing — размер промежутка между дочерними объектами.
Child Alignment (в этих примерах свойства Control Child Size и Child Force Expand не установлены):

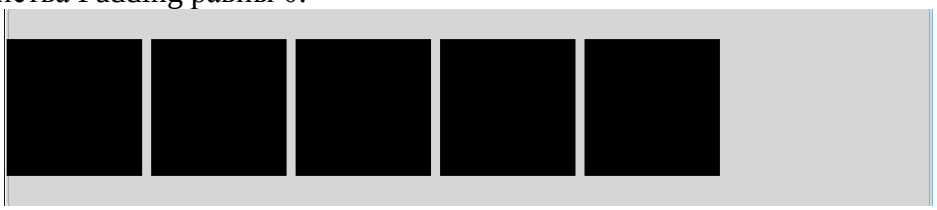




Control Child Size — если установлено, то компоновщик может изменять соответствующий размер дочернего объекта. Если при этом не установлено свойство Child Force Expand, то дочерние объекты не будут видны, если для них не задан компонент Layout Element с установленными значениями Preferred Width/Height).

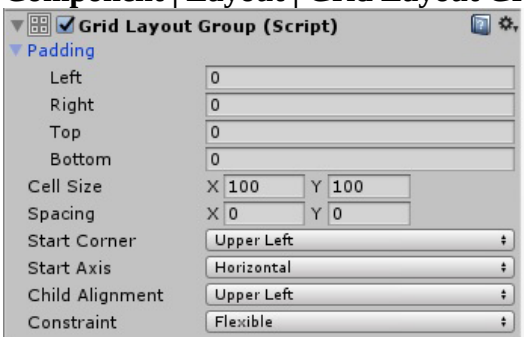
Child Force Expand — если установлено, то дочерние объекты могут захватывать доступное пространство по указанному измерению. Если при этом *не установлено* свойство Control Child Size, то объекты не изменяют свой размер, но смещаются в нужном направлении, причем к ним добавляется пустой промежуток нужного размера (при этом свойство Spacing игнорируется). Если свойство Control Child Size *установлено*, то размеры объектов изменяются так, чтобы они захватывали все доступное пространство (при этом свойство Spacing учитывается).

В следующих примерах свойство Child Alignment было установлено в Middle Left, Spacing = 10, все свойства Padding равны 0.

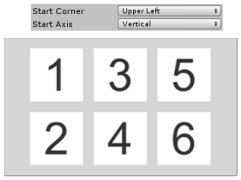

(свойства не установлены)

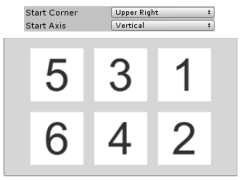
Компонент Grid Layout Group

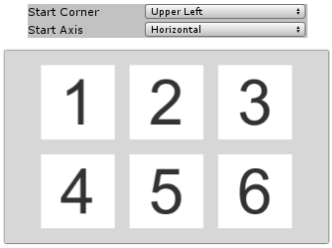
Component | Layout | Grid Layout Group

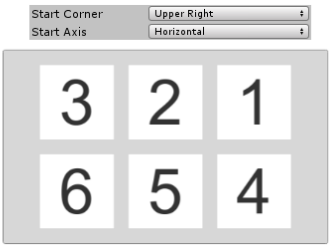


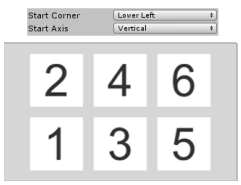
Start Corner и Start Axis

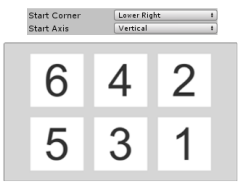


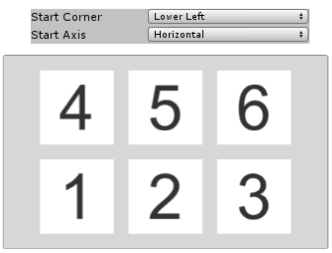


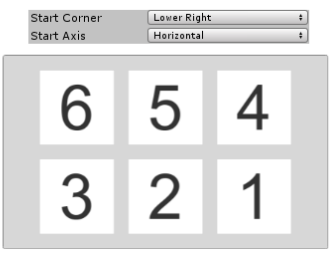




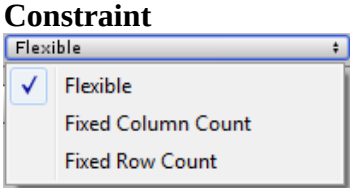








Constraint

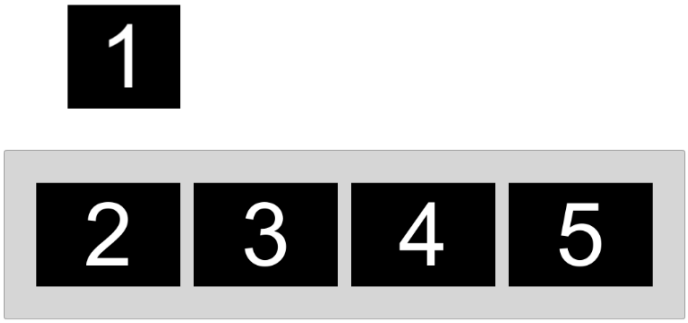


Компонент Layout Element

Component | Layout | Layout Element

Позволяет задать для объекта различные варианты размеров, используемые при автоматической компоновке.

Ignore Layout (установлено для объекта 1; игнорируется как позиция, так и масштаб):



Для Grid Layout Group настройки свойств Width и Height *не учитываются.*

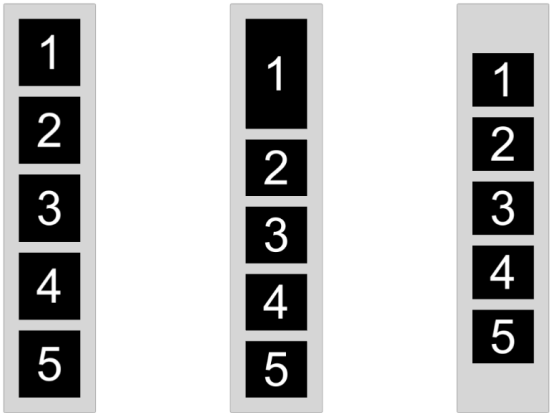
Min Width и Min Height (установлены для объекта 1):

Preferred Width и Preferred Height (задают максимальные размеры объекта, если для родительского компоновщика установлены Control Child Size и *не* установлены Child Force Expand)

Слева: для компоновщика установлены Control Child Size и Child Force Expand, для дочерних объектов *не установлены* Preferred Width и Preferred Height.

В центре: для компоновщика установлены Control Child Size и Child Force Expand, для объекта 1 *установлено* Preferred Height = 100. В результате объект 1 на 100 единиц выше, чем остальные (то есть Preferred Height в данном случае это не вся высота, а *дополнительная* высота объекта по сравнению с остальными).

Справа: для компоновщика установлены Control Child Size и Child Force Expand Width, но *не установлено* Child Force Expand Height, для всех дочерних объектов *установлено* Preferred Height = 100. В результате высота всех объектов равна 100. **Примечание**. Если задать Preferred Height = 200, то вариант справа совпадет с вариантом слева, то есть требуемая максимальная высота достигнута не будет, поскольку для этого пришлось бы выйти за границу родительского объекта.

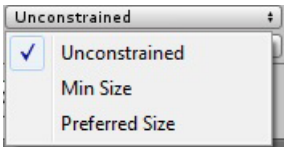


Flexible Width и Flexible Height (задаются в процентах, определяют *относительные* размеры дочерних объектов). Значения, меньшие 1 (100%), принимаются во внимание только если *не установлено* свойство Child Force Expand родительского компоновщика; свойство Control Child Size не учитывается).

Пример (в обоих случаях значения Flexible Width равны соответственно 0, 0.5, 0.75, 1 и 1.5):
сверху — Child Force Expand Width установлено, Control Child Size Width какое угодно,
снизу — Child Force Expand Width *не* установлено, Control Child Size Width какое угодно.

Компонент Content Size Fitter

Component | Layout | Content Size Fitter



Задается для *родительского* объекта, снабженного групповым компоновщиком, и подстраивает размер родительского объекта под размеры дочерних. После данной подстройки *нельзя* вернуться к прежним размерам, просто выбрав вариант Unconstrained (надо не только выбрать вариант Unconstrained, но и явно установить нужные размеры).

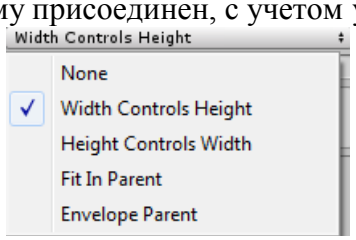
В случае компоновщиков Horizontal Layout Group и Vertical Layout Group родительский объект подстраивается под значения Min Size или Preferred Size своих дочерних объектов.

В случае компоновщика Grid Layout Group учитывается только его свойство Cell Size, наряду с Padding и Spacing (в этом случае режимы Min Size или Preferred Size работают одинаково).

Компонент Aspect Ratio Fitter

Component | Layout | Aspect Ratio Fitter

Настраивает размер объекта, к которому присоединен, с учетом указанного соотношения сторон.



Width Controls Height — высота подстраивается под указанную ширину,

Height Controls Width — ширина подстраивается под указанную высоту,

Fit in Parent — объект масштабируется с учетом соотношения сторон к размерам родительского объекта, оставаясь в его границах;

Envelope Parent — объект масштабируется с учетом соотношения сторон к размерам родительского объекта, полностью занимая родительский компонент по одному измерению и, возможно, выходя за его границы по другому измерению.

Компонент Aspect Ration Fitter нельзя применять к объекту, если его родительский объект содержит групповой компоновщик.