

Лекция 4. Основы компьютерной графики. User Interface

Вопросы перед лекцией к ДЗ

- Pivot можно менять только в среде моделирования. В Unity это поменять нельзя.
- **Gizmo Center** - это геометрический центр
- **Gizmo Pivot** - геометрическая точка, которая устанавливается в среде моделирования, вокруг которой удобно вращать
- Через код вращение происходит вокруг **Gizmo Pivot**

Основы компьютерной графики. Продолжение

Ответы на вопросы к прошлой лекции

- **Frame Debugger** - показывает как отрисовывает каждый кадр
- **Render Texture** - нужно релизить ручками, т.к. могут быть утечки
- **onPostRenderer** - отображает до появления кадра на экран

Шейдеры и пост-эффекты

- **v2f vert (appdata v)** - отвечает за вершинный шейдер
- **stencil buffer** - отрисовка определенных фрагментов, которые равны каким-то параметрам

User Interface

Canvas

- **Canvas** - компонент, который навешивается на GameObject
- Порядок отрисовки идет от первого к последнему среди детей **canvas**
- Режимы отображения
 - **ScreenSpace - Overlay** - 2D отображение на мониторе без перспективы, т.е. текст всегда будет 2D
 - **ScreenSpace - Camera** - возможность просмотра перспективы детей, например, посмотреть текст под углом. *3D - меню*
 - **World Space** - рендерятся в реальном мире. Есть возможность перемещения детей **canvas**
 - 3D элементы не рендерятся на **canvas**
- Плохое разрешение на **text**, потому что на **canvas** маленькое разрешение пикселей на unit
- При работе с 2D интерфейсом лучше использовать кнопку переключения
- **meta file** - настройки импорта, в котором хранятся также айдишники для, например, текстуры материала
- При импорте/экспорте объектов нужно обращать внимание на флаг - подтягивать зависимости, чтобы не было сломанных ассетов
- **RectTool** - позволяет удобно манипулировать **RectTransform** - hotkey T
 - Можно менять **pivot**

- Hotkey **Alt+Shift** при трансформации **pivot** дочернего объекта в **canvas** - изменить **gizmo pivot** и применить
- Якоря
 - Удобное позиционирование детей **canvas**
- **Text**
 - Переполнение
 - **wrapped** - текст не может выходить за **Rect Transform**
 - **overflow** - текст не может выходить за **Rect Transform**
- **Image**
 - По умолчанию нет **sprite**
 - **Sprite**
 - **simple** - просто отображаем картинку
 - **slice** - возможность растягивать стороны, без искажения углов, которые мы задаем
 - **tiled** - клонирование спрайта на весь **image**
 - **filled** - заливка картинки в заданном направлении, с заданной точки
- **RawImage**
 - Принимает на вход **texture**, а не **sprite**
- **Mask**
 - ограничивает дочерний элемент, родительским. Дочерний элемент будет показан на тех местах родительского элемента, где есть родительский
- **Outline**
 - обводка для разграничения элементов от накладывания друг на друга. В большом количестве использовать не рекомендовано. Большой прирост в затратах производительности
- **Raycast Target** - может ли элемент воспринимать интерфейсные клики и тому подобное. Т.е. является ли он декоративный
- Progress bar, как и полоски здоровья, лучше делать через два **image**. **Slider** - все же используется для ввода, т.к. на нем много каллбеков изменения состояния
 - Изменять состояние через полоски лучше через **anchor** из кода