

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук
Департамент программной инженерии

СОГЛАСОВАНО
Доцент департамента
программной инженерии,
кандидат технических наук

УТВЕРЖДАЮ
Академический руководитель
образовательной программы
«Программная инженерия»
профессор департамента программной
инженерии, канд. техн. наук



П.А. Манахов
«18» Сентября 2021 г.

В.В. Шилов
«__» _____ 2021 г.

UNITY-ПЛАГИН ДЛЯ ПРОТОТИПИРОВАНИЯ ЖЕСТОВЫХ КЛАВИАТУР В XR

Пояснительная записка

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.04.01-01 81 01-1-ЛУ

Име. № подл	Подп. и дата	Взам.	Име. № дубл.	Подп. и дата
RU.17701729.04.01-01 81 01-1				

Исполнитель
студент группы БПИ184



/ Е.Г. Байдаров /
«18» Сентября 2021 г.

Москва 2021

УТВЕРЖДЕН
RU.17701729.04.01-01 81 01-1-ЛУ

**РАЗРАБОТКА ПРОТОТИПА УСТРОЙСТВА ВВОДА И ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ ДЛЯ ВВОДА ТЕКСТА В РАСШИРЕННОЙ РЕАЛЬНОСТИ**

Пояснительная записка

RU.17701729.04.01-01 81 01-1

Листов 22

<i>Инв. № подл</i>	<i>Подп. и дата</i>	<i>Взам. инв.</i>	<i>Инв. № дубл.</i>	<i>Подп. и</i>
RU.17701729.04.01-01 81 01-1				

Москва 2021

АННОТАЦИЯ

В данном программном документе приведена пояснительная записка к приложению «Unity-плагин для прототипирования жестовых клавиатур в XR»

В разделе «Введение» указано наименование программы, краткое наименование программы и документы, на основании которых ведется разработка.

В разделе «Назначение и область применения» указано функциональное назначение программы, эксплуатационное назначение программы и краткая характеристика области применения программы.

В разделе «Технические характеристики» содержатся следующие подразделы:

- постановка задачи на разработку программы;
- описание алгоритма и функционирования программы с обоснованием выбора схемы алгоритма решения задачи и возможные взаимодействия программы с другими программами;
- описание и обоснование выбора метода организации входных и выходных данных;
- описание и обоснование выбора состава технических и программных средств.

В разделе «Ожидаемые технико-экономические показатели» указана предполагаемая потребность и экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами

Настоящий документ разработан в соответствии с требованиями:

- 1) ГОСТ 19.101-77 Виды программ и программных документов [1];
- 2) ГОСТ 19.102-77 Стадии разработки [2];
- 3) ГОСТ 19.103-77 Обозначения программ и программных документов [3];
- 4) ГОСТ 19.104-78 Основные надписи [4];
- 5) ГОСТ 19.105-78 Общие требования к программным документам [5];
- 6) ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом [6];
- 7) ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению [7].

Изменения к Пояснительной записке оформляются согласно ГОСТ 19.603-78 [8], ГОСТ 19.604-78 [9].

Перед прочтением данного документа рекомендуется ознакомиться с терминологией, приведенной в Приложении 1 настоящей пояснительной записки.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01 —01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

1.	ВВЕДЕНИЕ	5
1.1.	Наименование программы	5
1.2.	Документы, на основании которых ведется разработка	5
2.	НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ	6
2.1.	Назначение программы	6
2.1.1.	Функциональное назначение	6
2.1.2.	Эксплуатационное назначение	6
2.2.	Краткая характеристика области применения	6
3.	ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ	7
3.1.	Постановка задачи на разработку программы	7
3.2.	Описание алгоритма и функционирования программы	7
3.2.1.	Архитектура программной системы	7
3.2.2.	Обоснование выбора способа решения задачи	7
3.3.	Описание и обоснование выбора метода организации входных и выходных данных	8
3.3.1.	Описание метода организации входных и выходных данных	8
3.4.	Описание и обоснование выбора состава технических и программных средств	8
3.4.1.	Состав технических и программных средств	8
3.4.2.	Обоснование выбора технических и программных средств	8
4.	ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ	9
4.1.	Предполагаемая потребность	9
4.2.	Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами	9
5.	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	10

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01 — 01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

1. ВВЕДЕНИЕ

1.1. Наименование программы

Наименование программы – «Unity-плагин для прототипирования жестовых клавиатур в XR»

Краткое наименование программы – «Unity XR Keyboard».

1.2. Документы, на основании которых ведется разработка

Основанием для разработки является учебный план подготовки бакалавров по направлению 09.03.04 "Программная инженерия" и утвержденная академическим руководителем тема курсового проекта.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01 —01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

2.1. Назначение программы

2.1.1. Функциональное назначение

Функциональным назначением программы является использование возможностей Windows Touch Keyboard внутри Unity.

2.1.2. Эксплуатационное назначение

Разработанное приложение является Unity-плагином, который будет использоваться для упрощения создания клавиатур для жестового ввода в Unity.

2.2. Краткая характеристика области применения

Расширенная реальность и исследования способов ввода текста.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01 —01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

3.1. Постановка задачи на разработку программы

Постановка задачи на разработку программы изложена в документах «Unity-Плагин для прототипирования жестовых клавиатур в XR» Техническое задание.

3.2. Описание алгоритма и функционирования программы

3.2.1. Архитектура программной системы

Система реализована с использованием DesktopDuplicationAPI для отрисовки клавиатуры и библиотеки UWP для инъекции ввода мыши. Также дополнительно используется WPF виджет с полем ввода, события от которого проброшены в Unity. Отрисованная с помощью DesktopDuplicationApi клавиатура, реализует интерфейсы системы событий пользовательского ввода в Unity3D. Реализация интерфейсов основана на инъекциях с помощью библиотеки UWP для управления курсором мыши. Полем ввода является WPF виджет в который осуществляется ввод текста вызовом методов интерфейсов для обработки событий системы событий Unity. Полученный данные пишутся в MemoryMappedFile и читаются из него на стороне Unity.

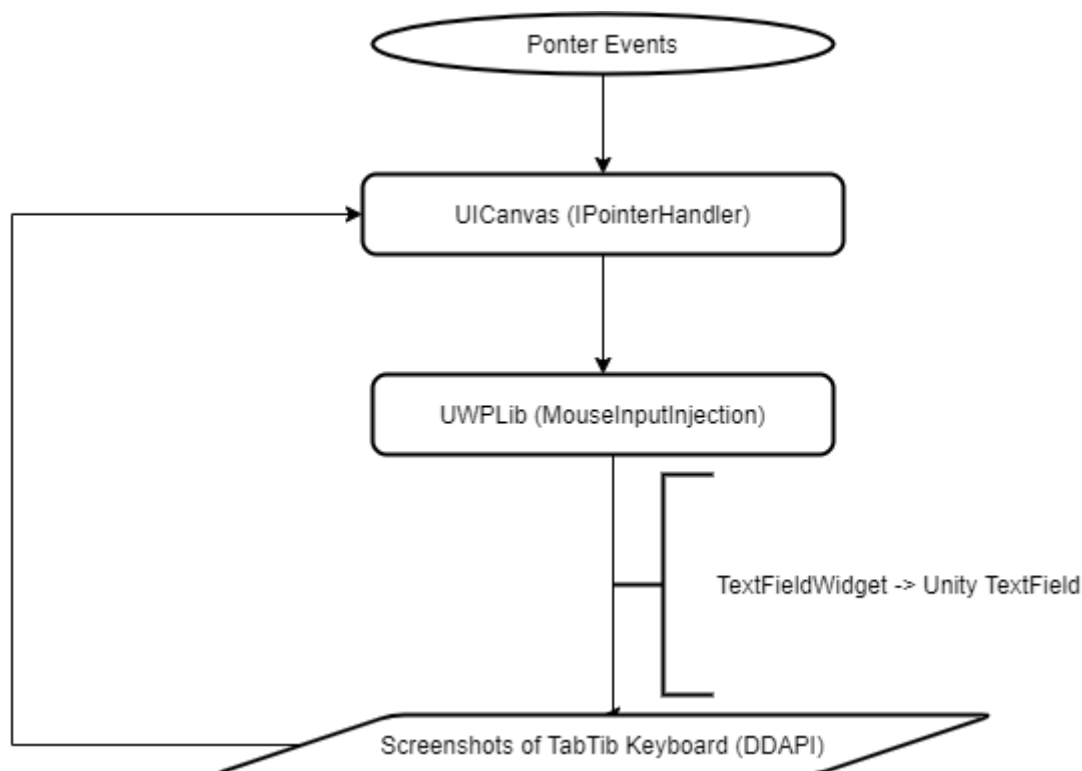


Рисунок 1 Архитектура системы плагина

3.2.2. Обоснование выбора способа решения задачи

DesktopDuplicationApi: интерфейс для прямого доступа к изображению рабочего стола, позволяет с минимальной потерей производительности получить 2DTexture объект из фрагмента рабочего стола.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01 —01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Инъекции ввода мыши: позволяют правильно работать с курсором рабочего стола для доступа и отправки нажатий к клавиатуре Windows.

Виджет с полем ввода WPF поддерживает предикативный ввод в отличие от стандартных полей ввода Unity.

MemoryMappedFile: наиболее простой и быстрый способ для локальной межпроцессовой коммуникации.

Стандартные события Unity: обеспечение независимости XR фреймворка использованного в прототипе.

3.3. Описание и обоснование выбора метода организации входных и выходных данных

3.3.1. Описание метода организации входных и выходных данных

Входными данными являются значения атрибутов события системы событий Unity.

Выходные данные формируются системой Windows и представляются пользователю средствами плагина в виде захваченного изображения клавиатуры и введенного текста.

3.4. Описание и обоснование выбора состава технических и программных средств

3.4.1. Состав технических и программных средств

- Intel Core i5-4590/AMD FX 8350 аналогичная или более новая модель
- NVIDIA GeForce GTX 1060, AMD Radeon RX 480 аналогичная или более новая модель
- 4 ГБ ОЗУ или более
- HDMI 1.4, DisplayPort 1.2 или более новая модель
- 1x USB 2.0 или более новая модель
- Windows 10 (OS build: 19042.867)

3.4.2. Обоснование выбора технических и программных средств

Технические и программные средства выбраны основываясь на системных требованиях Unity [10] и рекомендуемых системных требованиях предполагаемой VR платформы для тестирования HTC VIVE [1].

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01 —01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

4.1. Предполагаемая потребность

- Упрощение создания прототипов жестового ввода текста для AR и VR,
- Использование существующей виртуальной клавиатуры (Windows Touch Keyboard) для реализации "тяжелых" функций вроде предикативной системы и системы коррекции опечаток

4.2. Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами

В рамках данной работы подсчет экономической эффективности не предусмотрен

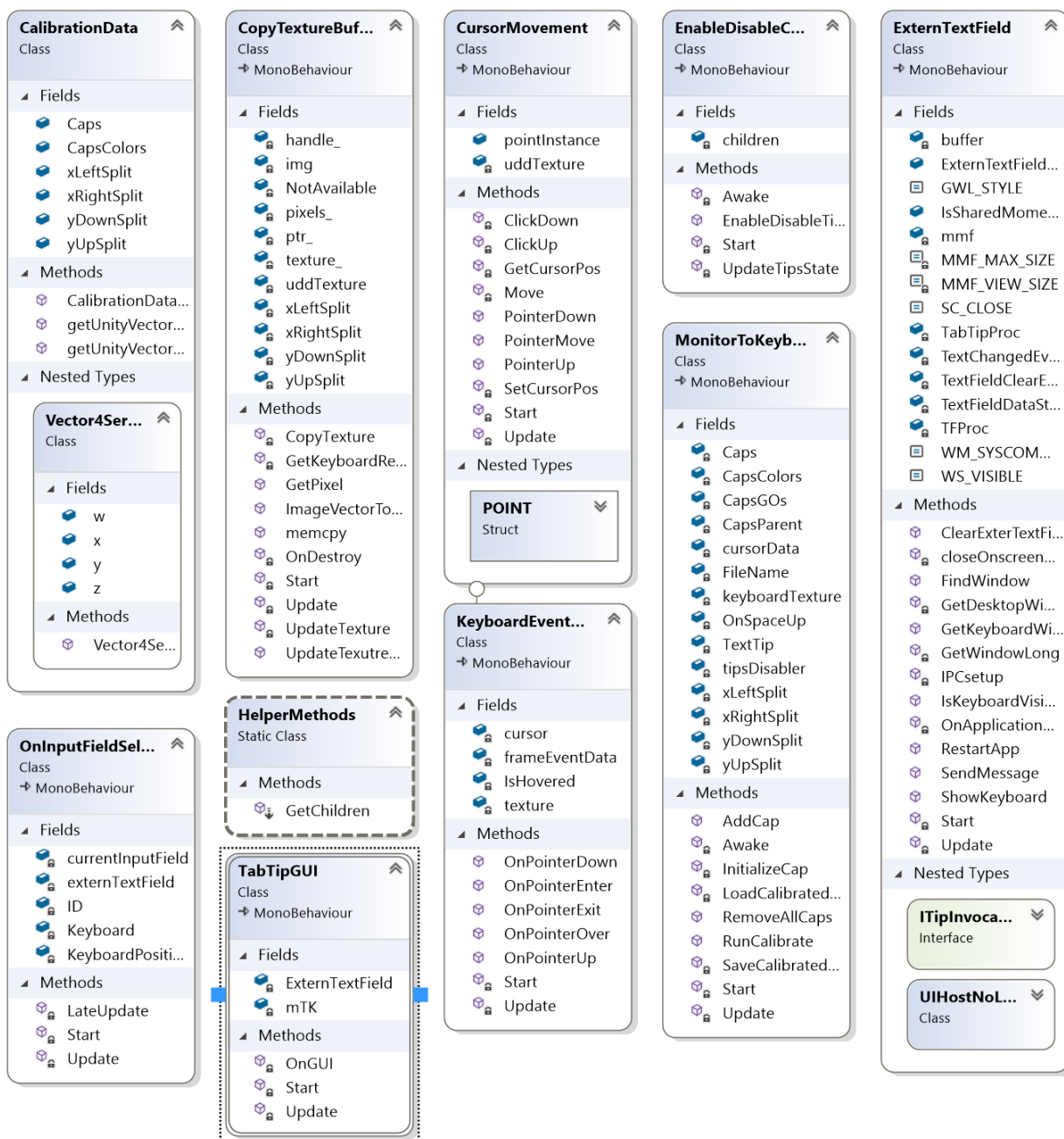
Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01 —01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

5. СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 19.101-77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.104-78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
5. ГОСТ 19.105-78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7. ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8. ГОСТ 19.603-78 Общие правила внесения изменений. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
9. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
10. System requirements for Unity 2019.3 [Электронный ресурс]: документация / Unity. – Режим доступа : <https://docs.unity3d.com/Manual/system-requirements.html>
11. System requirements for VIVE [Электронный ресурс]: поддержка / Vive. – Режим доступа :https://www.vive.com/us/support/wireless-adapter/category_howto/vive-wireless-system-requirements.html

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01 —01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ПРИЛОЖЕНИЕ 1 ДИАГРАММА КЛАССОВ ПЛАГИНА



Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01 —01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ПРИЛОЖЕНИЕ 2

КРАТКОЕ ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ ПЛАГИНА

Таблица 1.1 Класс	Назначение
CalibrationData.cs	Сериализуемый класс данных калибровки
CopyTextureBuffer.cs	Создание 2DTexture
CursorMovement.cs	Управление движениями курсора и получение данных о его позиции на рабочем столе
EnableDisableCalibrationTips.cs	Включение выключение подсказок при калибровке
ExternTextField.cs	Внешнее поле ввода
OnInputFieldSelected.cs	Управление отображением клавиатуры при выборе поля ввода в сцене Unity
HelperMethods.cs	Вспомогательные методы плагина
KeyboardEventHendler.cs	Обработчик событий нажатия на Canvas с клавиатурой
MonitorToKeyboard.cs	Выделение участка клавиатуры из изображение рабочего стола
ExternTextField.cs	Управление внешним полем ввода из TextFieldWidget.exe
TabTipGUI.cs	Отображения кнопок управления плагином на экране монитора

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01 —01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ПРИЛОЖЕНИЕ 2

ПОДРОБНОЕ ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ ПЛАГИНА

Классы

Класс CalibrationData

Класс модель для хранения данных о размерах

Классы

- class **Vector4Serializable**
Сериализуемый 4-х мерный вектор

Открытые члены

- List< Vector4 > **getUnityVectors** ()
- CalibrationData** (int yUpSplit, int xRightSplit, int yDownSplit, int xLeftSplit, List< Vector4 > caps, int[] yColor, int[] xColor)

Открытые атрибуты

- int **yUpSplit**
- int **yDownSplit**
- int **xLeftSplit**
- int **xRightSplit**
- Vector4Serializable[] Caps**
- int[] **xColor**
- int[] **yColor**

Подробное описание

Класс модель для хранения данных о размерах

Класс CopyTextureBuffer

Копирование буфера изображения рабочего стола в Unity Texture2d GameObject. Копирование происходит каждый кадр. Отображается заданный участок рабочего стола

Открытые члены

- static IntPtr **memcpy** (IntPtr dest, IntPtr src, int count)
native с mem copy function
- void **UpdateTexutreWithBorders** (int yUpSplit, int yDownSplit, int xLeftSplit, int xRightSplit)
Обновить текстуру с учетом указанных границ
- void **AddCap** (Vector4 cap, Vector2Int colorCoords)
добавляет "прикрывашку" по координатам на экране и координатам пикселя с цветом
- Vector3 **ImageVectorToDesktopPos** (Vector3 ImageLocalPosition)

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01 —01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Перевод координат на Image объекте в координаты на рабочем столе

- void **LoadAllCaps** (List< Vector4 > caps, int[] yColor, int[] xColor)
load all caps into scene

Подробное описание

Копирование буфера изображения рабочего стола в Unity Texture2d GameObject. Копирование происходит каждый кадр. Отображается заданный участок рабочего стола

Методы

void CopyTextureBuffer.AddCap (Vector4 cap, Vector2Int colorCoords)

добавляет "прикрывашку" по координатам на экране и координатам пикселя с цветом

Аргументы

cap	координаты границы
colorCoords	координаты пикселя с цветом

Vector3 CopyTextureBuffer.ImageVectorToDesktopPos (Vector3 ImageLocalPosition)

Перевод координат на Image объекте в координаты на рабочем столе

Аргументы

ImageLocalPosition	Координаты луча на объекте изображения клавиатуры в сцене
--------------------	---

Возвращает

Координаты на рабочем столе

void CopyTextureBuffer.LoadAllCaps (List< Vector4 > caps, int[] yColor, int[] xColor)

load all caps into scene

Аргументы

caps	caps borders
yColor	y coords of caps colors
xColor	x coords of caps colors

static IntPtr CopyTextureBuffer.memcpy (IntPtr dest, IntPtr src, int count)

native с mem copy function

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01 —01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Аргументы

dest	Destination pointer
src	Source pointer
count	buffer length

Возвращает

void CopyTextureBuffer.UpdateTexutreWithBorders (int yUpSplit, int yDownSplit, int xLeftSplit, int xRightSplit)

Обновить текстуру с учетом указанных границ

Аргументы

yUpSplit	верхняя координата по оси y
yDownSplit	нижняя координата по оси y
xLeftSplit	левая координата по оси x
xRightSplit	правая координата по оси x

Класс CursorMovement

Управление событиями и позицией указателя

Классы

- struct **POINT**
Структура данных для точки указателя в WinApi

Открытые члены

- static bool **GetCursorPos** (out **POINT** lpPoint)
Получить текущую позицию курсора
- static bool **SetCursorPos** (int x, int y)
Устанавливает позицию курсора на координаты на рабочем столе. Левый верхний угол (0, 0). Правый нижний угол (displayWidth, displayHeight)
- void **PointerDown** (int x, int y)
Обработчик событий PointerDown
- void **PointerEnter** (Vector2Int point)
Обработчик событий PointerEnter
- void **PointerUp** ()
Обработчик событий PointerUp
- void **PointerMove** (Vector2Int mv)
Обработчик событий PointerMove

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01 —01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- void **UpdateRealCursor** ()
Обновление данных о положении курсора

Открытые атрибуты

- **POINT** pointInstance
- Vector2Int **RealCursorXY**
- bool **IsPointerDown**

Статические открытые данные

- static object **_lock** = new object()

Свойства

- Vector2Int **CanvasCursorXY** [getset]

Подробное описание

Управление событиями и позицией указателя

Методы

static bool CursorMovement.GetCursorPos (out POINT lpPoint)

Получить текущую позицию курсора

Аргументы

lpPoint	защищенный указатель на Point структуру
---------	---

Возвращает

true если успешно, false если не удалось взять данные о позиции курсора

void CursorMovement.PointerDown (int x, int y)

Обработчик событий PointerDown

Аргументы

x	x координата
y	y координата

void CursorMovement.PointerEnter (Vector2Int point)

Обработчик событий PointerEnter

Аргументы

point	Точка входа
-------	-------------

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01 —01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

void CursorMovement.PointerMove (Vector2Int mv)

Обработчик событий PointerMove

Аргументы

mv	Вектор сдвига
----	---------------

static bool CursorMovement.SetCursorPos (int x, int y)

Устанавливает позицию курсора на координаты на рабочем столе. Левый верхний угол (0, 0). Правый нижний угол (displayWidth, displayHeight)

Аргументы

x	x координата
y	y координата

Возвращает

true если успешно, false если не удалось установить позицию курсора<

Объявления и описания членов класса находятся в файле:

- /Users/ebaydarov/UnityTabTip/Assets/TabTipKeyboard/Scripts/CursorMovement.cs

Класс EnableDisableCalibrationTips

Управление видимостью подсказок калибровки на клавиатуре

Открытые члены

- void **EnableDisableTips** (bool IsEnabled)
Запуск отложенного бновления видимости подсказок

Подробное описание

Управление видимостью подсказок калибровки на клавиатуре

Методы

void EnableDisableCalibrationTips.EnableDisableTips (bool IsEnabled)

Запуск отложенного бновления видимости подсказок

Аргументы

IsEnabled	
-----------	--

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01 —01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Класс ExternTextField

Управление виджетом поля ввода

Открытые члены

- void **SharedMemorySetup** ()
Настройка разделяемой памяти для передачи между виджетом и полем ввода
- void **RestartApp** ()
Перезапуск процесса виджета
- void **ShowKeyboard** ()
Отобразить клавиатуру Windows TabTip лунищфкв
- static IntPtr **FindWindow** (String sClassName, String sAppName)
Поиск окна по имени класса и имени приложения
- static int **SendMessage** (int hWnd, uint Msg, int wParam, int lParam)
Отправка команды через COM интерфейс

Открытые статические члены

- static bool **IsKeyboardVisible** ()
Свойство видимости клавиатуры
- static IntPtr **GetKeyboardWindowHandle** ()
Хендлер клавиатуры

Открытые атрибуты

- string **ExternTextFieldData** = ""

Статические открытые данные

- static bool **IsSharedMemoryReachable** = true
- const UInt32 **WS_VISIBLE** = 0x94000000
[http://msdn.microsoft.com/en-gb/library/windows/desktop/ms632600\(v=vs.85\).aspx](http://msdn.microsoft.com/en-gb/library/windows/desktop/ms632600(v=vs.85).aspx).
- const int **GWL_STYLE** = -16
Specifies we wish to retrieve window styles.
- const int **WM_SYSCOMMAND** = 0x0112
- const int **SC_CLOSE** = 0xF060

Подробное описание

Управление виджетом поля ввода

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01 —01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Методы

static IntPtr ExternTextField.FindWindow (String sClassName, String sAppName)

Поиск окна по имени класса и имени приложения

Аргументы

sClassName	Имя класса
sAppName	Имя приложения

Возвращает

Хендлер окна

static IntPtr ExternTextField.GetKeyboardWindowHandle () [static]

Хендлер клавиатуры

Возвращает

Указатель(Хендлер) окна клавиатуры

static bool ExternTextField.IsKeyboardVisible () [static]

Свойство видимости клавиатуры

Возвращает

true если видима.

static int ExternTextField.SendMessage (int hWnd, uint Msg, int wParam, int lParam)

Отправка команды через COM интерфейс

Аргументы

hWnd	Хендлер окна
Msg	сообщение
wParam	ширина
lParam	длина

Возвращает

Статус отправки

Класс KeyboardEventHandler

Обработчик событий Unity UI

Открытые члены

- void **OnPointerDown** (PointerEventData eventData)
Обработчик события PointerDown

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01 —01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- void **OnPointerOver** (PointerEventData eventData)
Обработчик события PointerOver
- void **OnPointerEnter** (PointerEventData eventData)
Обработчик события PointerEnter
- void **OnPointerExit** (PointerEventData eventData)
Обработчик события PointerExit
- void **OnPointerUp** (PointerEventData eventData)
Обработчик события PointerUp

Статические открытые данные

- const int **MagicCoeff** = 10

Подробное описание

Обработчик событий Unity UI

Методы

void KeyboardEventHandler.OnPointerDown (PointerEventData eventData)

Обработчик события PointerDown

Аргументы

eventData	Данные указателя
-----------	------------------

void KeyboardEventHandler.OnPointerEnter (PointerEventData eventData)

Обработчик события PointerEnter

Аргументы

eventData	Данные указателя
-----------	------------------

void KeyboardEventHandler.OnPointerExit (PointerEventData eventData)

Обработчик события PointerExit

Аргументы

eventData	Данные указателя
-----------	------------------

void KeyboardEventHandler.OnPointerOver (PointerEventData eventData)

Обработчик события PointerOver

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01 —01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Аргументы

eventData	Данные указателя
-----------	------------------

void KeyboardEventHandler.OnPointerUp (PointerEventData eventData)

Обработчик события PointerUp

Аргументы

eventData	Данные указателя
-----------	------------------

Класс MonitorToKeyboard

Транслирования части изображения рабочего стола с клавиатурой в юнити

Открытые члены

- void **RunCalibrate** ()
Запуск скрипта калибровки размеров клавиатуры
- void **AddCap** ()
3Впуск скрипта добавления "прикрывашки"
- void **LoadCaps** ()
Загрузка сохраненных "прикрывашек" в сцену

Подробное описание

Транслирования части изображения рабочего стола с клавиатурой в юнити

Структура CursorMovement.POINT

Структура данных для точки указателя в WinApi

Открытые члены

- **POINT** (int x, int y)

Открытые статические члены

- static implicit operator **System.Drawing.Point** (**POINT** p)
- static implicit operator **POINT** (System.Drawing.Point p)

Открытые атрибуты

- int X
- int Y

Подробное описание

Структура данных для точки указателя в WinApi

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01 —01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Класс **TorsoReferencedContent**

Плавная относительная привязка объектов в пространстве

Открытые члены

- virtual void **SwitchEnabled** ()
включение/выключение скрипта

Защищенные члены

- Проверка конфигурации и вычисление сдвига virtual void **Start** ()
Проверка конфигурации и вычисление сдвига
- virtual void **Update** ()
Плавное смещение объекта

Защищенные данные

- new Transform **camera**
- float **distanceFromCamera** = 2.5f
- float **pitch** = 0f
- Vector3 **offset**

Статические защищенные данные

- static readonly float **POSITION_LERP_SPEED** = 5f

Подробное описание

Плавная относительная привязка объектов в пространстве

Класс **CalibrationData.Vector4Serializable**

Сериализуемый 4-х мерный вектор

Открытые члены

- **Vector4Serializable** (Vector4 cap)

Открытые атрибуты

- int **x**
- int **y**
- int **z**
- int **w**

Подробное описание

Сериализуемый 4-х мерный вектор

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01 —01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

[illegible]