МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**Липецкий Государственный Технический Университет**

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

Лабораторная работа №2

по Проектированию пользовательских интерфейсов  
“Текстовый редактор”

Студент \_\_\_\_\_\_\_\_\_\_\_\_ Станиславчук С. М.

(подпись, дата)

Группа АС-21-1

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_ Назаркин О. A.

(подпись, дата)

Липецк 2023 г.

**2. Задание**

Реализовать текстовый редактор и его графический интерфейс. Текстовый редактор нужен с расширенными функциями:   
0) Неизменный шрифт

1) Для элементов текста должна быть возможность делать их жирными, курсивными, подчеркнутыми.

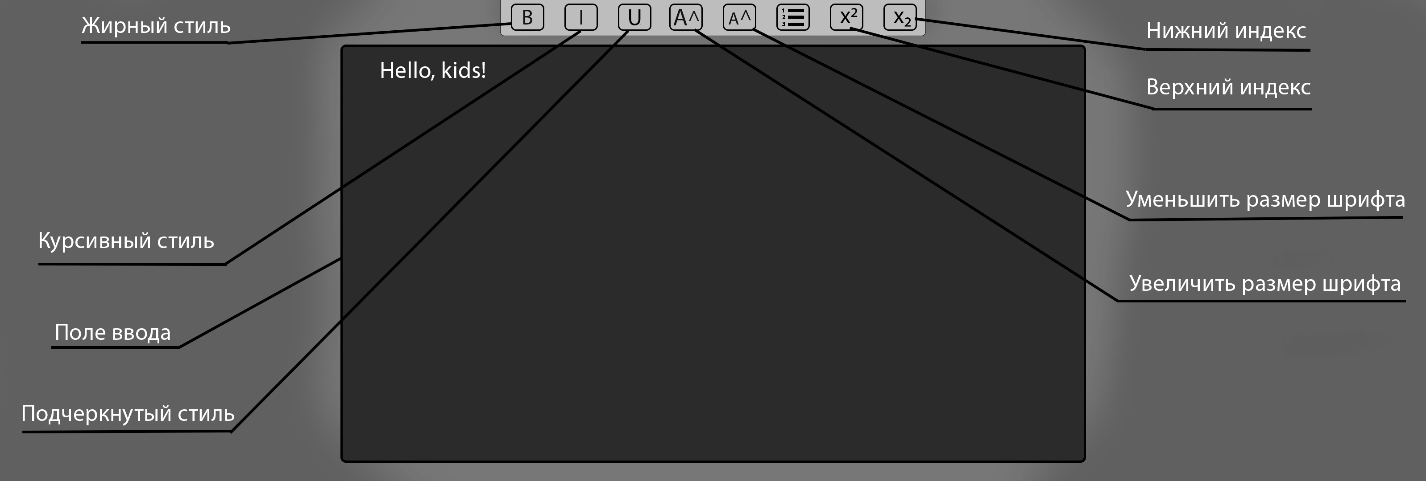
2) Поддержка списков

3) Поддержка верхнего и нижнего индекса x2

4) Подсветка слов в соответствии с регулярными выражениями, настраиваемый список ( RegExp ) (не выполнено)

**3. Интерфейс программы**

Интерфейс калькулятора представляет собой панель ввода и панель вышеописанных функций.



**4. Пример программного кода**

Скрипт, реализующий один из стилей текста – жирный.

using UnityEngine;

using TMPro;

public class FontStyles : MonoBehaviour {

public TMP\_InputField textInputField;

public void SetBold()

if (textInputField != null && textInputField.text != null)

{

int start = textInputField.selectionAnchorPosition;

int end = textInputField.selectionFocusPosition;

if (start == end) // if no text is selected, assume bold toggle for current cursor position

{

if (!textInputField.textInfo.characterInfo[start].style & FontStyles.Bold)

{

textInputField.text = textInputField.text.Insert(start, "<b>");

textInputField.text = textInputField.text.Insert(start + 3, "</b>");

}

else

{

textInputField.text = textInputField.text.Remove(start, 3);

textInputField.text = textInputField.text.Remove(start + 3, 4);

}

textInputField.caretPosition = start + 3;

}

else // if text is selected, apply bold to entire selection

{

string newText = textInputField

text.Substring(0, start) + "<b>" + textInputField.text.Substring(start, end - start) + "</b>" + textInputField.text.Substring(end);

textInputField.text = newText;

textInputField.caretPosition = end + 7; // move caret to end of selection with closing bold tag included

textInputField.selectionAnchorPosition = start; // reset selection to start

textInputField.selectionFocusPosition = end + 7; // reset selection to end with closing bold tag included

}

}

Скрипт, реализующий нижний и верхний индексы

using UnityEngine;

using TMPro;

public class TextFormatting : MonoBehaviour

{

// Ссылка на InputField

public TMP\_InputField inputField;

// Преобразовать выделенный текст в верхний индекс

public void ToSuperscript()

{

// Получить выделенный текст

string selectedText = inputField.text.Substring(inputField.selectionAnchorPosition, inputField.selectionFocusPosition - inputField.selectionAnchorPosition);

// Создать тег <sup> и добавить выделенный текст

string superscript = "<sup>" + selectedText + "</sup>";

// Заменить выделенный текст на тег <sup>

inputField.text = inputField.text.Remove(inputField.selectionAnchorPosition, inputField.selectionFocusPosition - inputField.selectionAnchorPosition);

inputField.text = inputField.text.Insert(inputField.selectionAnchorPosition, superscript);

// Сбросить выделение и установить курсор в конец выделенного текста

inputField.selectionAnchorPosition = inputField.selectionFocusPosition = inputField.selectionAnchorPosition + superscript.Length;

}

// Преобразовать выделенный текст в нижний индекс

public void ToSubscript()

{

// Получить выделенный текст

string selectedText = inputField.text.Substring(inputField.selectionAnchorPosition, inputField.selectionFocusPosition - inputField.selectionAnchorPosition);

// Создать тег <sub> и добавить выделенный текст

string subscript = "<sub>" + selectedText + "</sub>";

// Заменить выделенный текст на тег <sub>

inputField.text = inputField.text.Remove(inputField.selectionAnchorPosition, inputField.selectionFocusPosition - inputField.selectionAnchorPosition);

inputField.text = inputField.text.Insert(inputField.selectionAnchorPosition, subscript);

// Сбросить выделение и установить курсор в конец выделенного текста

inputField.selectionAnchorPosition = inputField.selectionFocusPosition = inputField.selectionAnchorPosition + subscript.Length;

}

}

**5. Используемые средства.**

C#, Unity, библиотека TextMeshPro, изображения с сети.

**6. Вывод**

В ходе выполненной работы реализовал текстовый редактор и его графический интерфейс.

Субъективная оценка графического интерфейса ULVEEU по пятибалльной шкале: 2, 3, 3, 2, 3, 2.