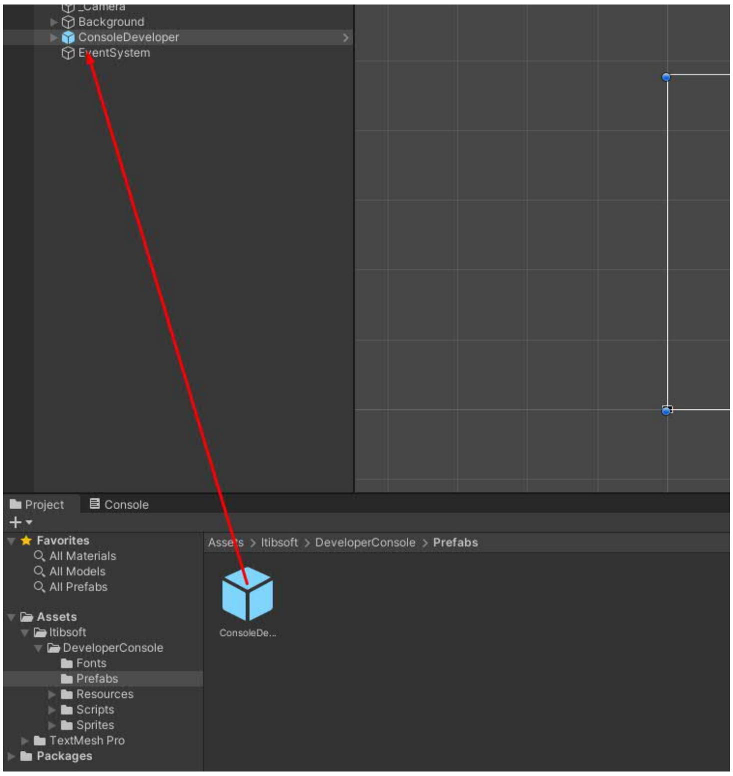
Documentation: Console Developer v0.0.7 [Free]

- A stripped-down version. Most likely, there will not be those implementations that are currently in development.

How to use:

1. First you need to drag the prefab of the console itself (ConsoleDeveloper.prefab) onto the main scene. It is located: Assets\Itibsoft\DeveloperConsole\Prefabs
You also need to check that you have an "EventSystem" on the stage. Without it, adding to the console will not occur. It is created in the standard way. RMB in an empty space of the hierarchy - UI – EventSystem



1. Having turned on the Play Mode (Just turn on the game), you need to press the "~" button to open the console itself. Then you can use it.
2. By clicking on the command entry bar, we can enter all any available commands in the console. To familiarize yourself with them, you can write the command /help
3. The command history buffer works in such a way that if you have already entered a command, the console adds it to the history. And if in the next call you need the same command or an earlier entered one, then you can easily go through the history of the commands entered with the arrows on the keyboard. The up arrow displays the commands that were previously entered. And the back arrow displays commands that are more recent. Works on the principle LIFO (Last input – first out).

Principle of operation:

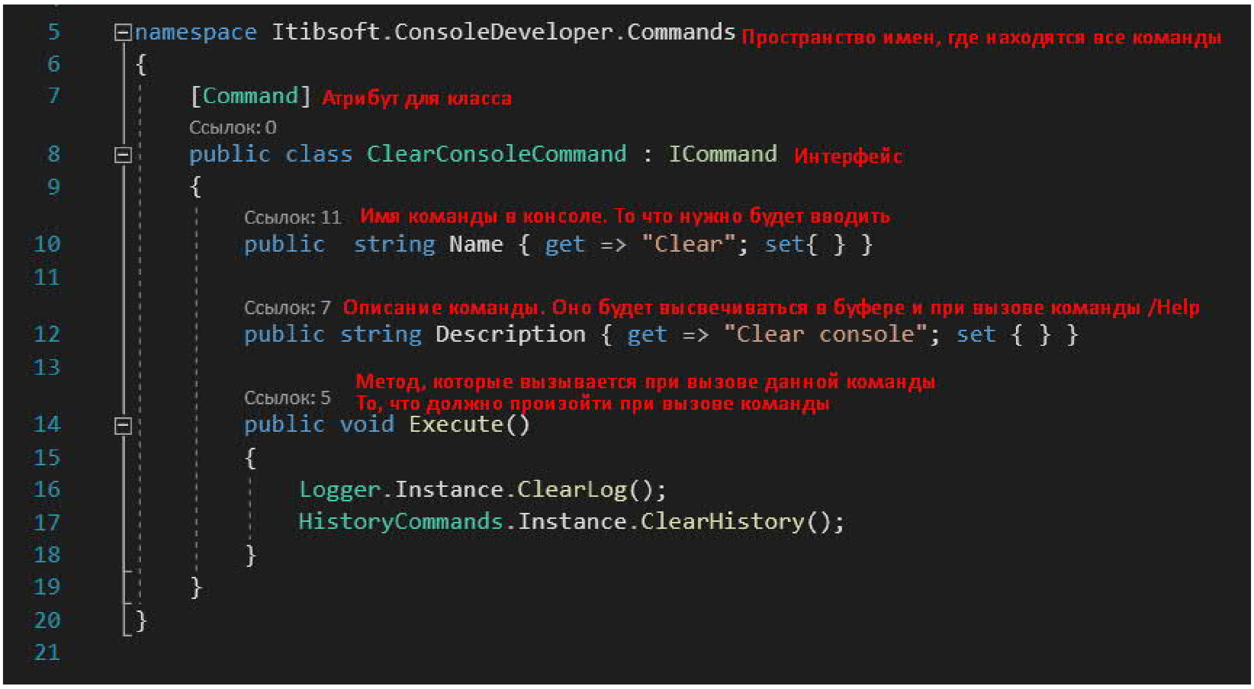
1. How to add new commands:
   a. Creating a new class:

      1. It needs to be given a name: NameCommand. This is not important, but it will just be more clear what this class is for.

      2. After creating the class, you need to place it namespace: Itibsoft.ConsoleDeveloper.Commands

      3. The class needs to sign the attribute [Command]. This is necessary in order for this command to be added to the list of active console commands. If this attribute is not present, the console will not see the given command.

      4. In this class, you need to implement the interface ICommand. This interface contains two properties Name(The name of the command, which will need to be entered in the console) and Description (Description of this command. It appears when the command is invoked Help). The interface also has a method Execute(); It does what should happen when you call the command from the console. There, write your implementation of what should happen. Everything that only comes to your mind can be realized there.



```
namespace Itibsoft.ConsoleDeveloper.Commands Пространство имен, где находятся все команды
{
    [Command] Атрибут для класса
    Ссылок: 0
    public class ClearConsoleCommand : ICommand Интерфейс
    {
        Ссылок: 11  Имя команды в консоле. То что нужно будет вводить
        public  string Name { get => "Clear"; set{ } }

        Ссылок: 7 Описание команды. Оно будет высвечиваться в буфере и при вызове команды /Help
        public string Description { get => "Clear console"; set { } }

                   Метод, которые вызывается при вызове данной команды
        Ссылок: 5  То, что должно произойти при вызове команды
        public void Execute()
        {
            Logger.Instance.ClearLog();
            HistoryCommands.Instance.ClearHistory();
        }
    }
}
```