**Преграда**

1. Создаётся объект со спрайтом преграды
2. К нему добавляется Box Collider 2D

**Взаимодействие с рычагом, кнопкой, нажимной плитой и т.д.**

Создаётся 3 объекта со спрайтами рычага, кнопки и двери им добавляется свойство Box Collider 2D, Animator и скрипт. В Box Collider 2D поставить галочку напротив isTrigger. Для каждого объекта создаётся по 2 анимации включённого/выключенного состояния. В Animator создаётся булевый параметр, отвечающий за переключение между анимациями.

Скрипт рычага

Создаётся переменная Anim(1) ей присваивается аниматор(4).

Далее создаются две булевые переменные: OnOff и inArea(2).

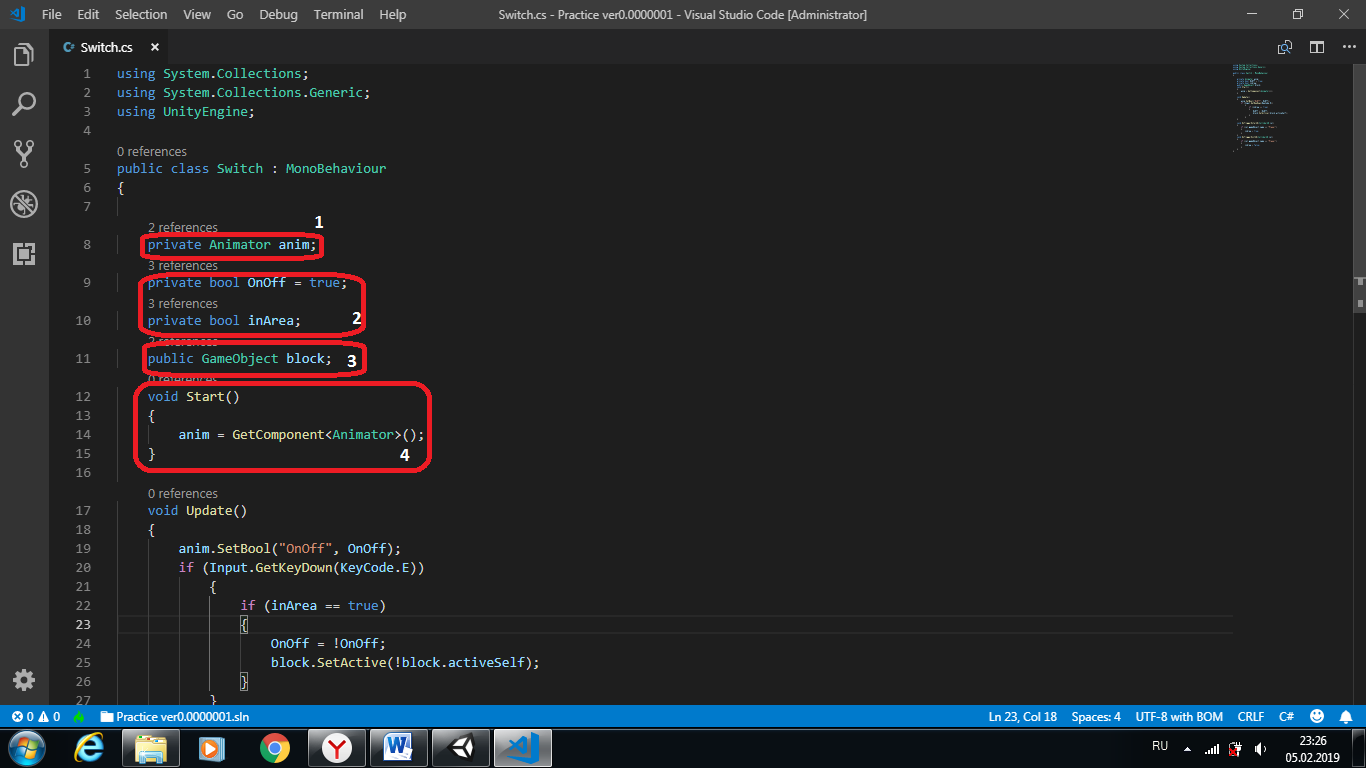
OnOff отвечает за смену анимации, inArea за нахождение персонажа в пределах колайдера .

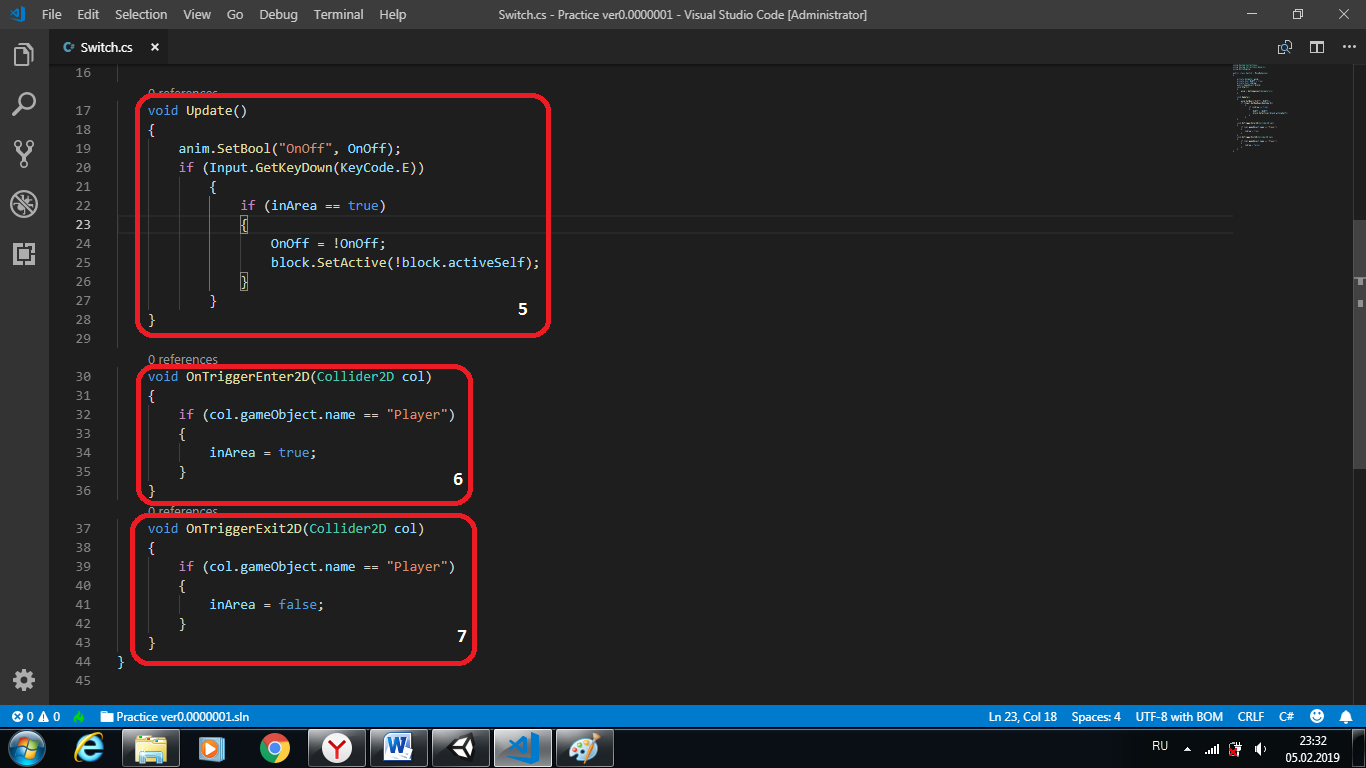
Метод OnTriggerEnter2D(6) – срабатывает при соприкосновении с колайдером рычага и колайдера другого объекта. Переменная inArea становится true если с рычагом соприкасается колайдер персонажа.

Метод OnTriggerExit2D(7) – срабатывает, если коллайдеры рычага и другого объекта перестают соприкасаться. Переменная inArea становится false если с колайдером рычага перестаёт соприкасаться колайдер персонажа.

Метод Update - вызывается один раз за кадр. Вызываем параметр OnOff из аниматора и присваиваем его переменной OnOff. Анимация меняется если нажата кнопка E и inArea(т.е персонаж находится в непосредственной близости с рычагом) имеет значение true.

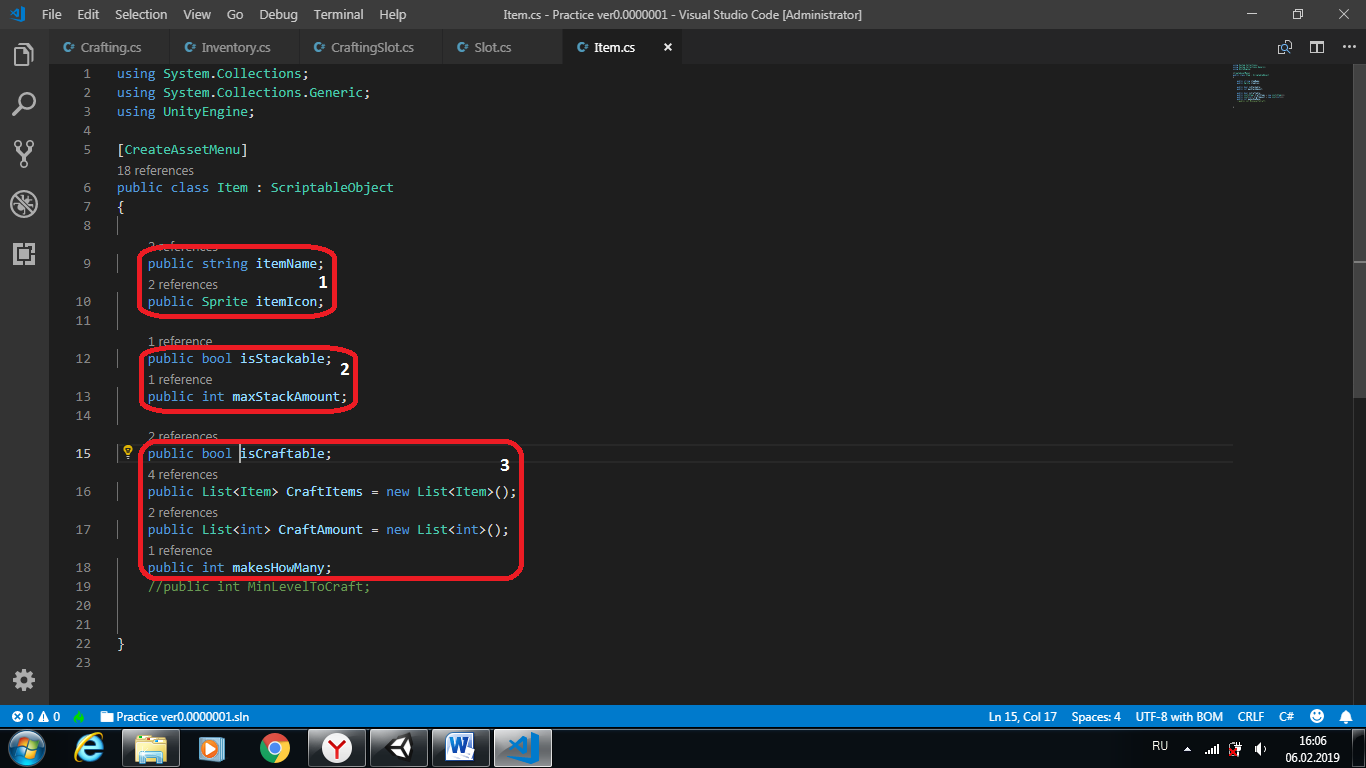
Скрипты кнопки и двери работают аналогичным образом.





**Система крафта**

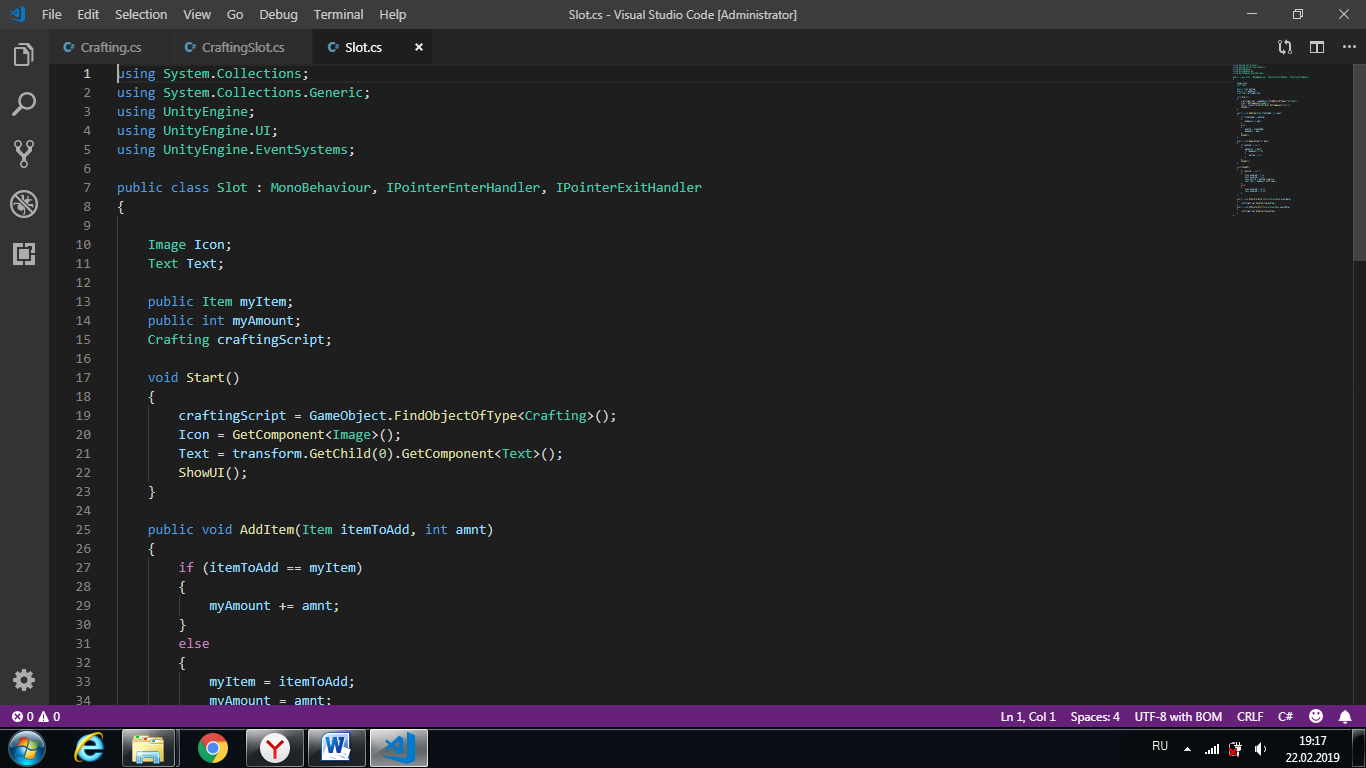
Скрипт Item

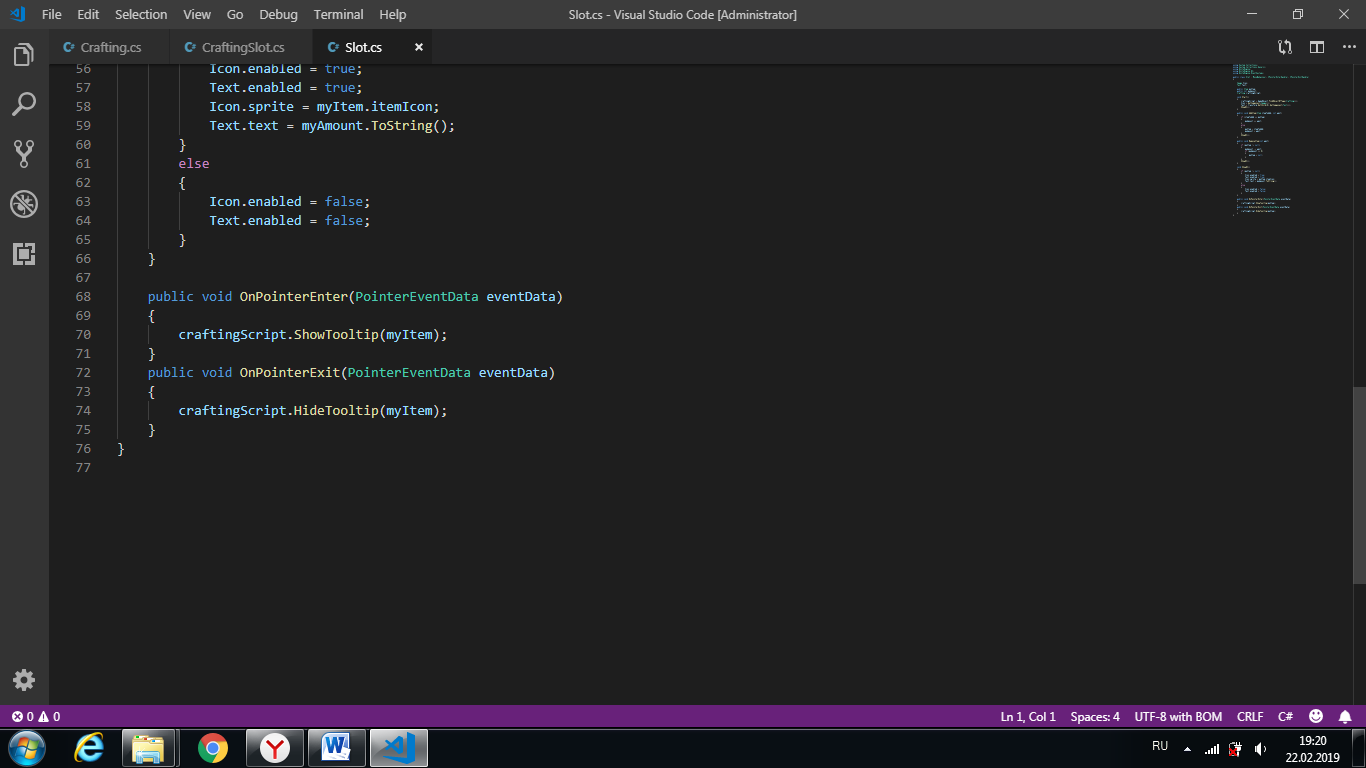


В этом скрипте создаются изменяемые параметры для будущих предметов, такие как имя и спрайт предмета(1), можно ли поместить в один слот больше 1 предмета и если да то сколько(2), можно ли создать этот предмет, какие и сколько других предметов нужно чтобы создать этот предмет и сколько предметов будет создано(3).

Скрипт Slot

Этот скрипт находится в каждом слоте инвентаря. В классе созданы 2 интерфейса на события входа (IPointerEnterHandler), выхода (IPointerExitHandler) курсора мыши которые воспроизводят 2 метода ShowTooltip и HideTooltip, из скрипта Crafting, соответственно.



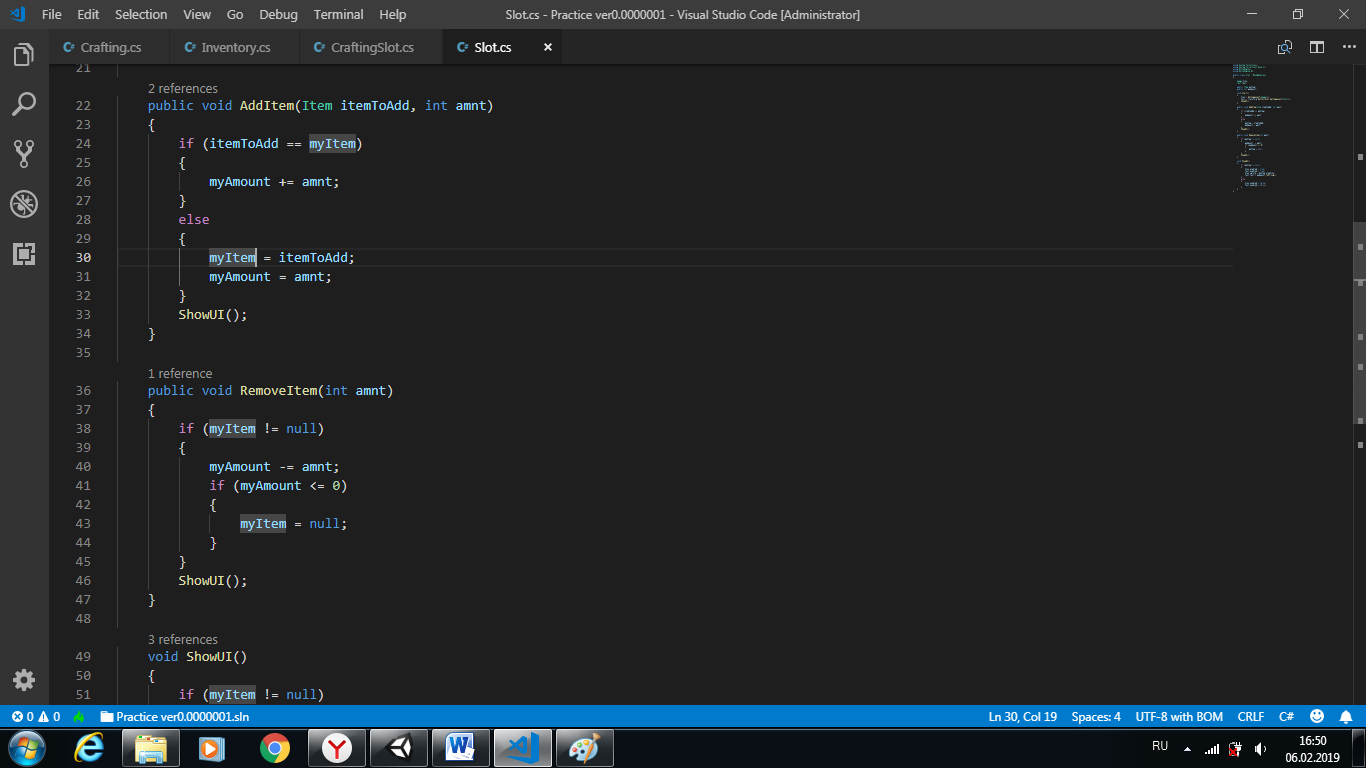


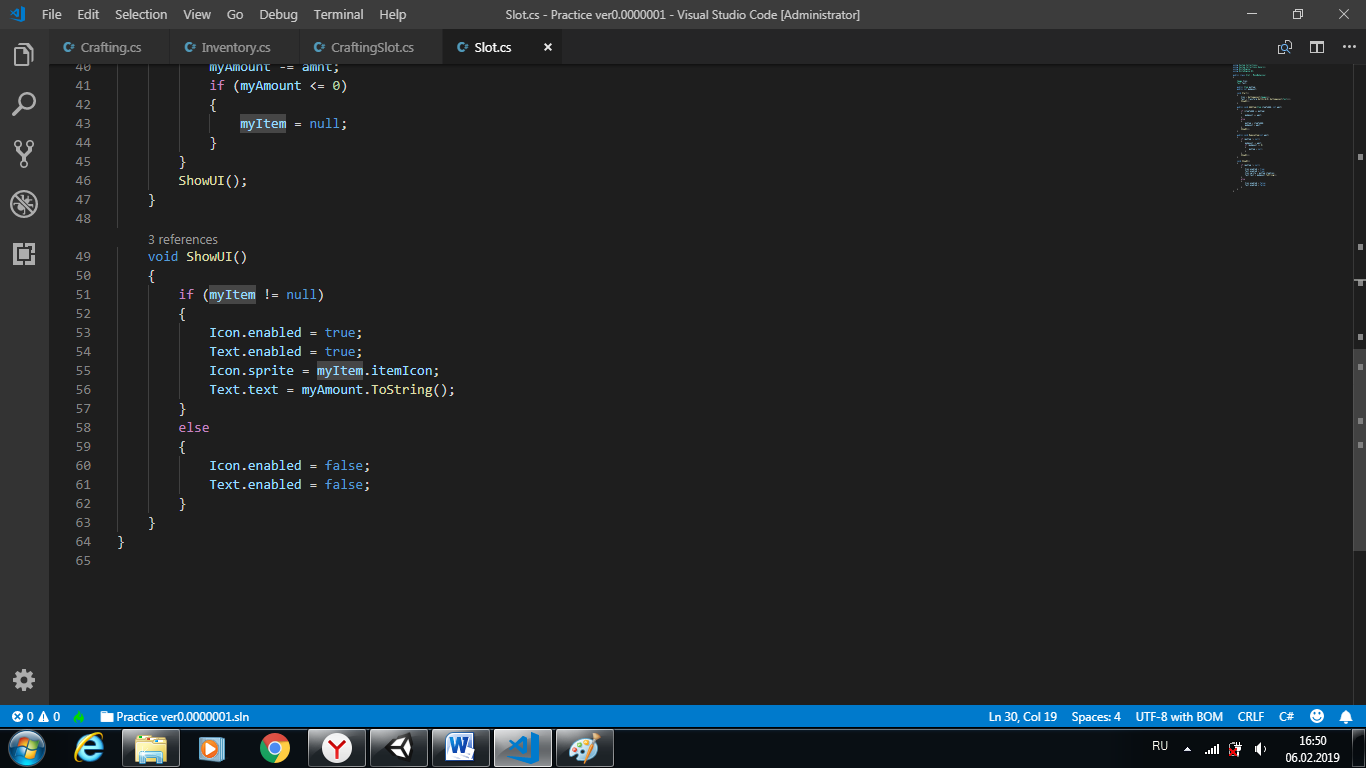
Также там находятся 3 метода отвечающие за добавление в слот, убирание из слота предмета, показ UI.

Метод ShowUI –показывает спрайт предмета и его количество в слоте, если предмет присутствует и не показывает ничего, если слот пустой.

Метод AddItem – добавляет предмет в слот и если в слот добавляется идентичный предмет, то количество этого предмета в слоте увеличивается.

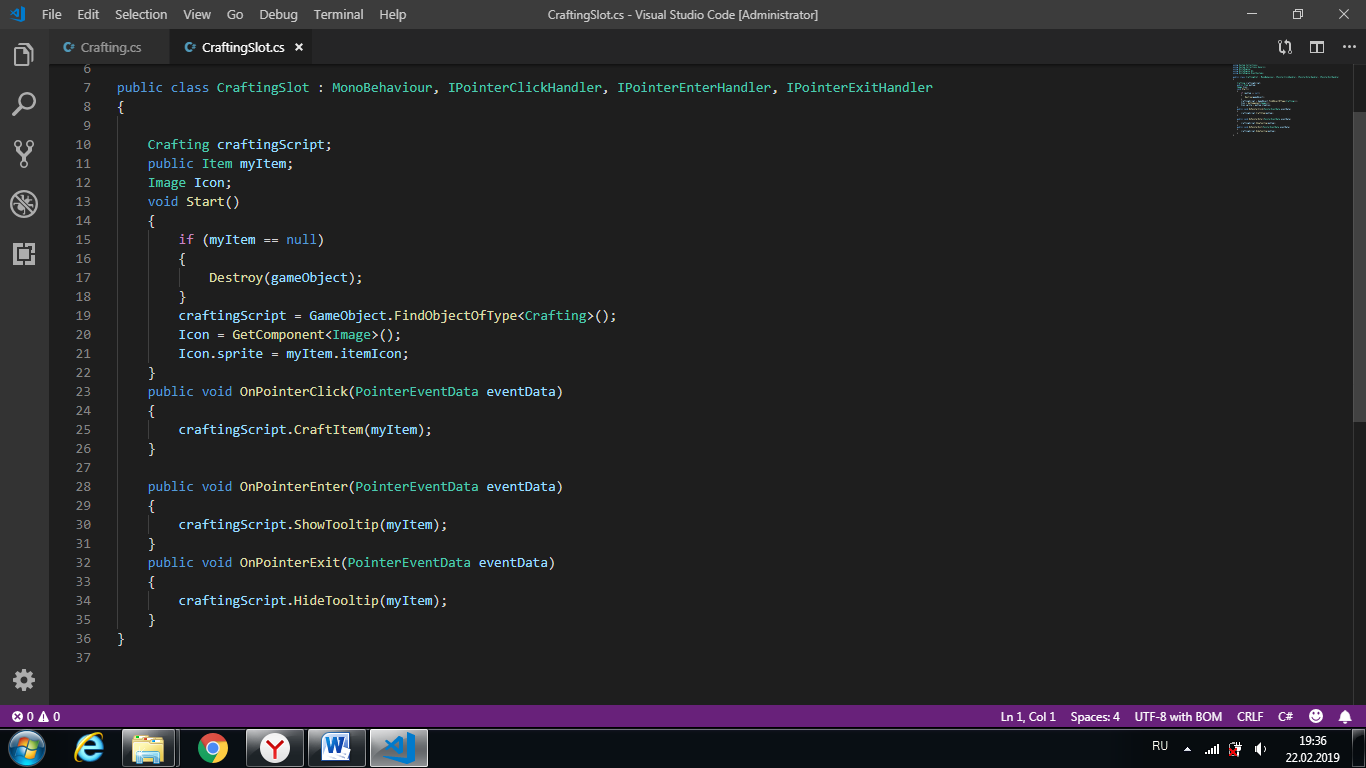
Метод RemoveItem - Очищает слот или уменьшает количество предметов в 1 слоте.





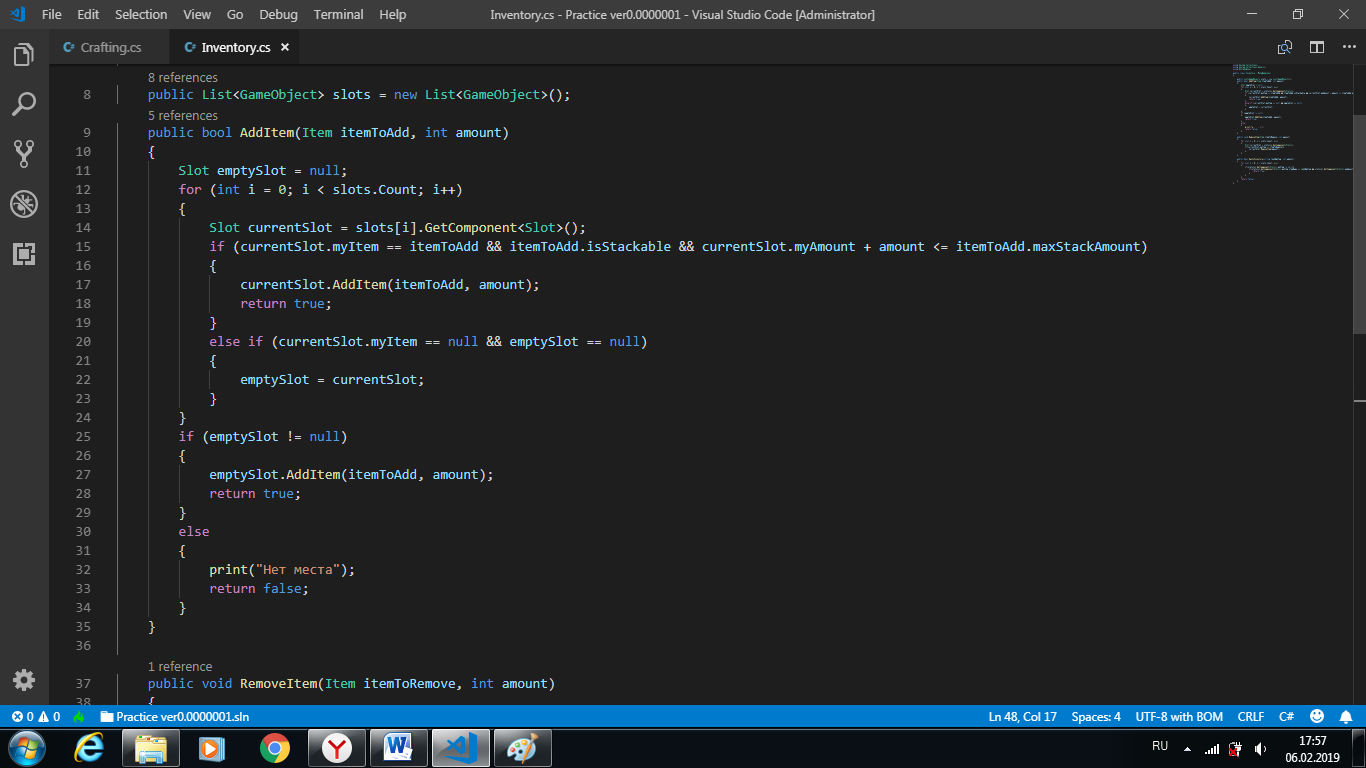
Скрипт CraftingSlot

Находится в слотах на панели крафта. В классе созданы 3 интерфейса на события входа (IPointerEnterHandler), выхода (IPointerExitHandler) и клика(IPointerClickHandler) курсора мыши. События входа и выхода курсора мыши вызывают аналогичные методы что и в скрите Slot. Событие клика вызывает метод CraftItem из скрипта Crafting.



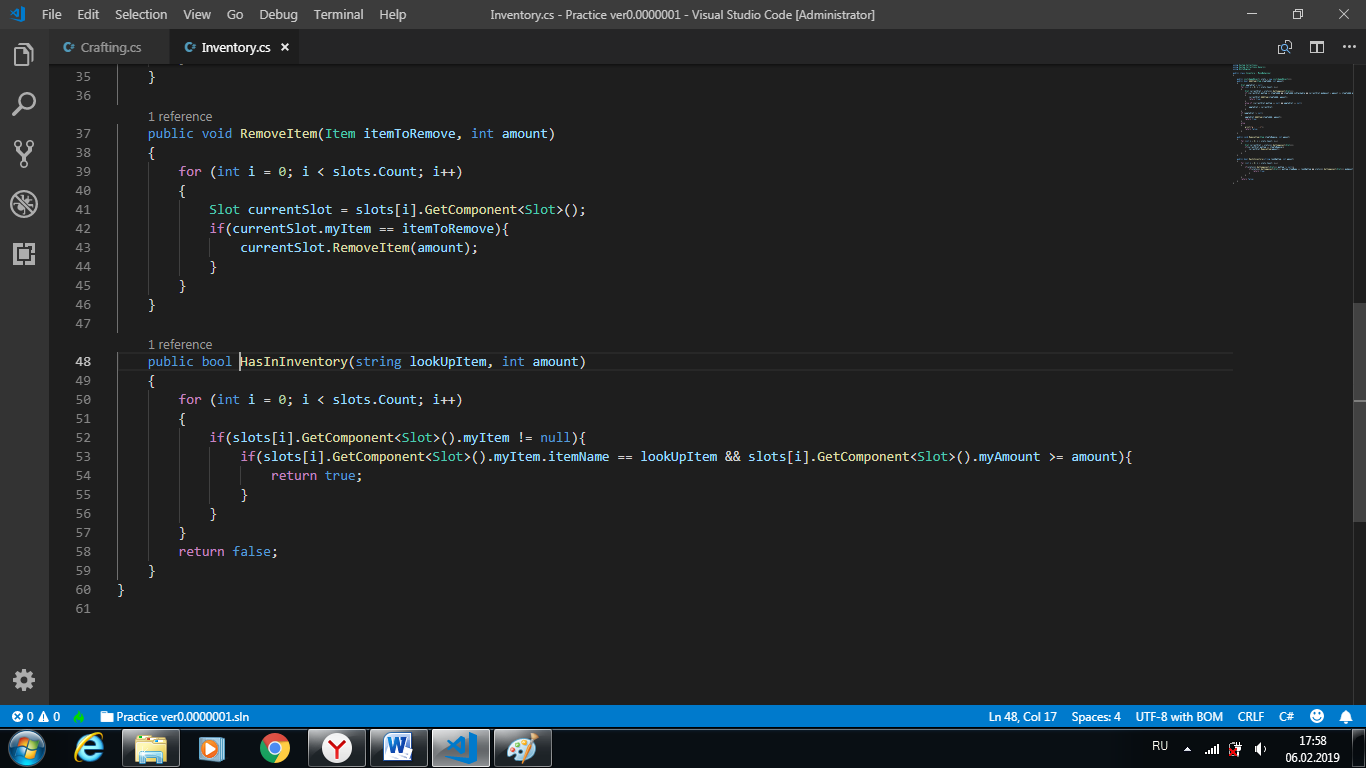
Скрипт Inventory

Метод AddItem – перед тем как добавить предмет в инвентарь делает проверку по всем слотам. Если слот пустой добавляет предмет в слот посредством метода AddItem из скрипта Slot или если слот занят, а предмет, который хотят добавить идентичен тому, что в слоте то делает вторую проверку, не привышает ли слот максимальное допустимое количество предметов и если нет то добавляет предмет в слот посредством метода AddItem из скрипта Slot и возвращает true, а если все слоты заполнены то возвращает false и прописывает в консоли «Нет места».



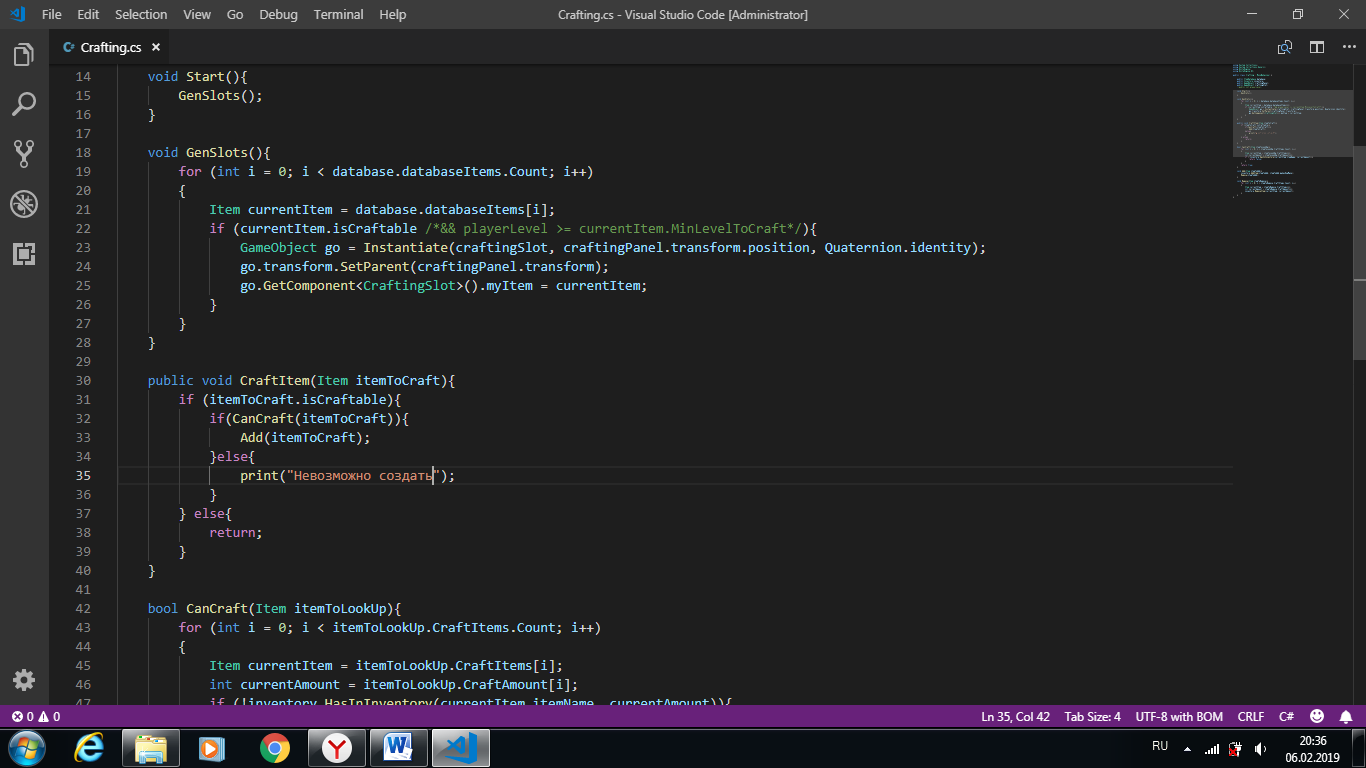
Метод RemoveItem – делает проверку по всем слотам и если какой-то слот удовлетворяет проверку то выполняет метод RemoveItem из скрипта Slot.

Метод HasInInventory – проверяет находится ли нужный нам предмет в инвентаре, если да то возвращает true, если нет то false.

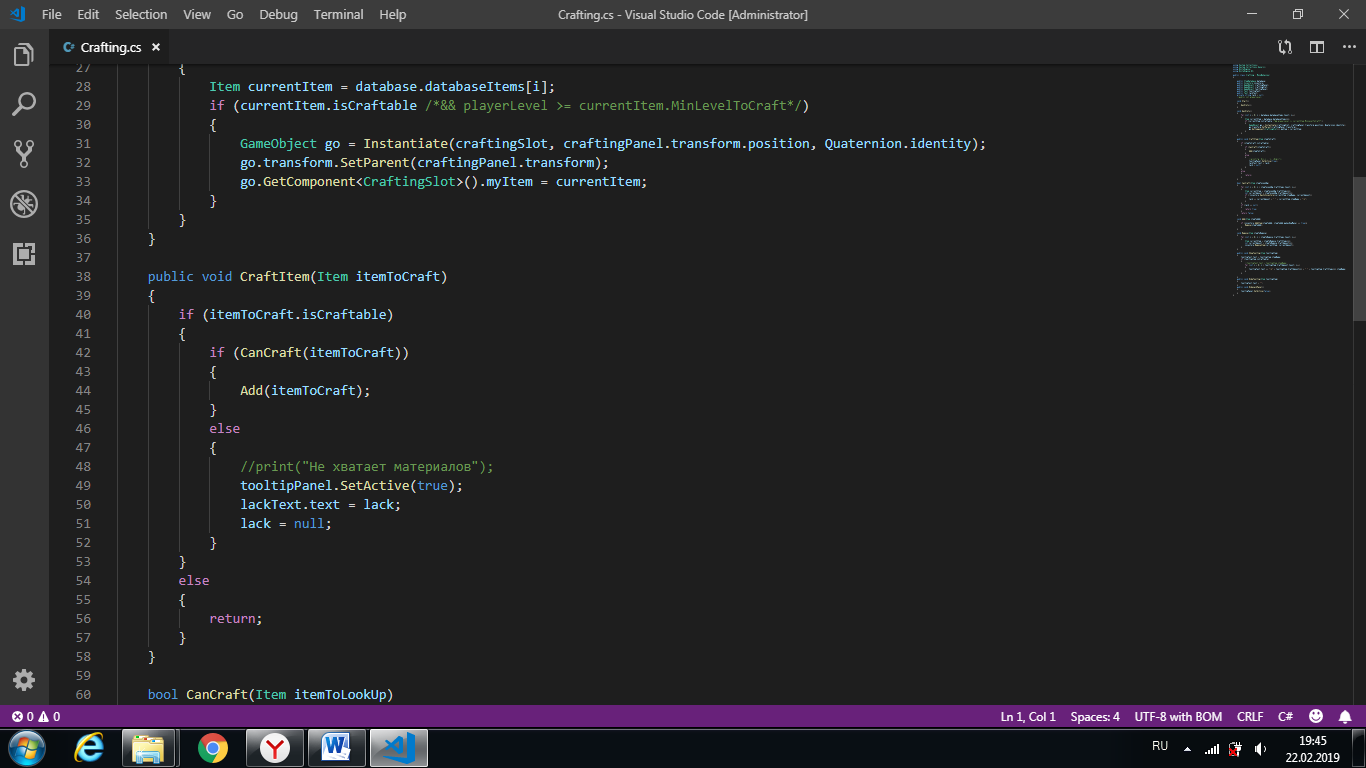


Скрипт Crafting

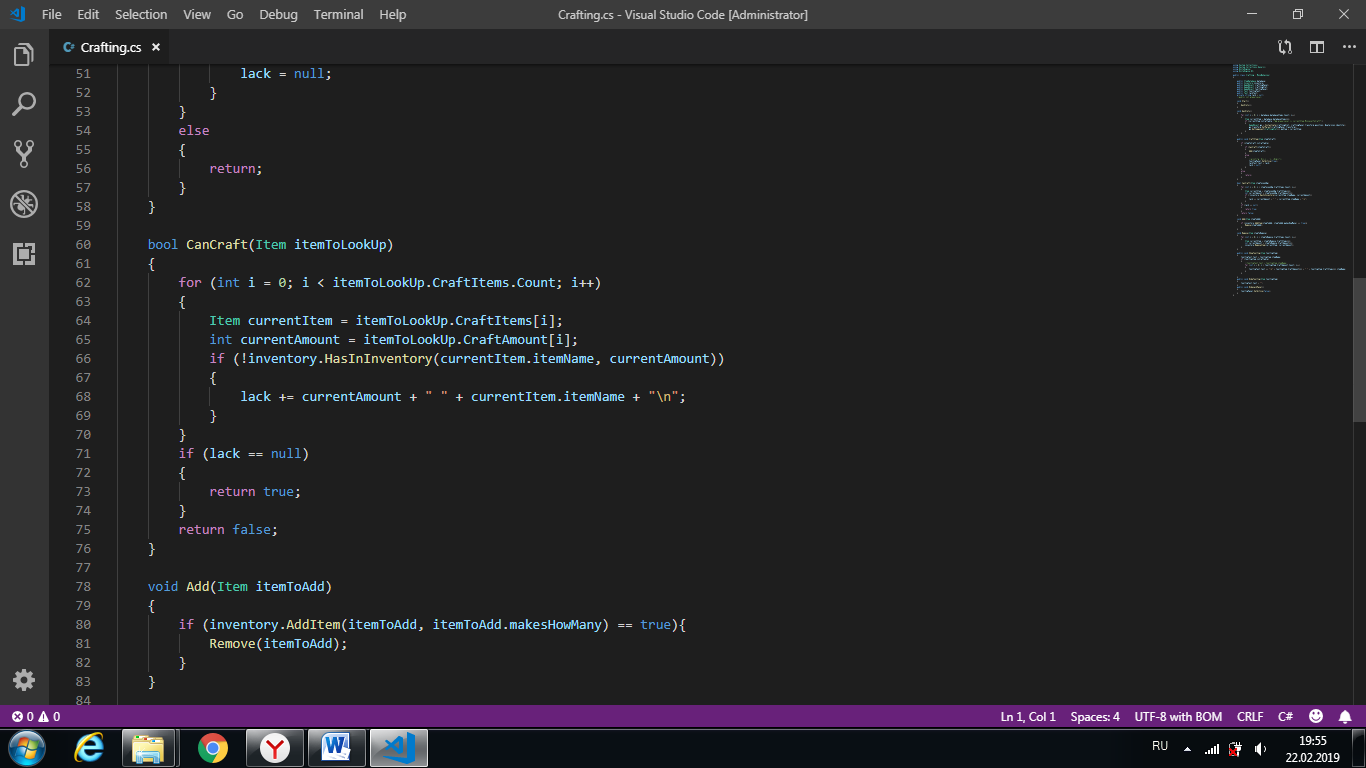
При старте выполняется метод GenSlots – он заполняет панель крафта слотами всех предметов которые можно создать.



Метод Craftitem – проверяет предмет на возможность его создания, после делает вторую проверку на то достаточно ли ресурсов для создания и сколько предметов не достаёт(если такие есть) путём метода CanCraft(описанного ниже) при положительном результате выполняет метод Add(описанный ниже), при отрицательном делает активной панель подсказкок в которую записываются не достающие материалы находящиеся в переменной lack.

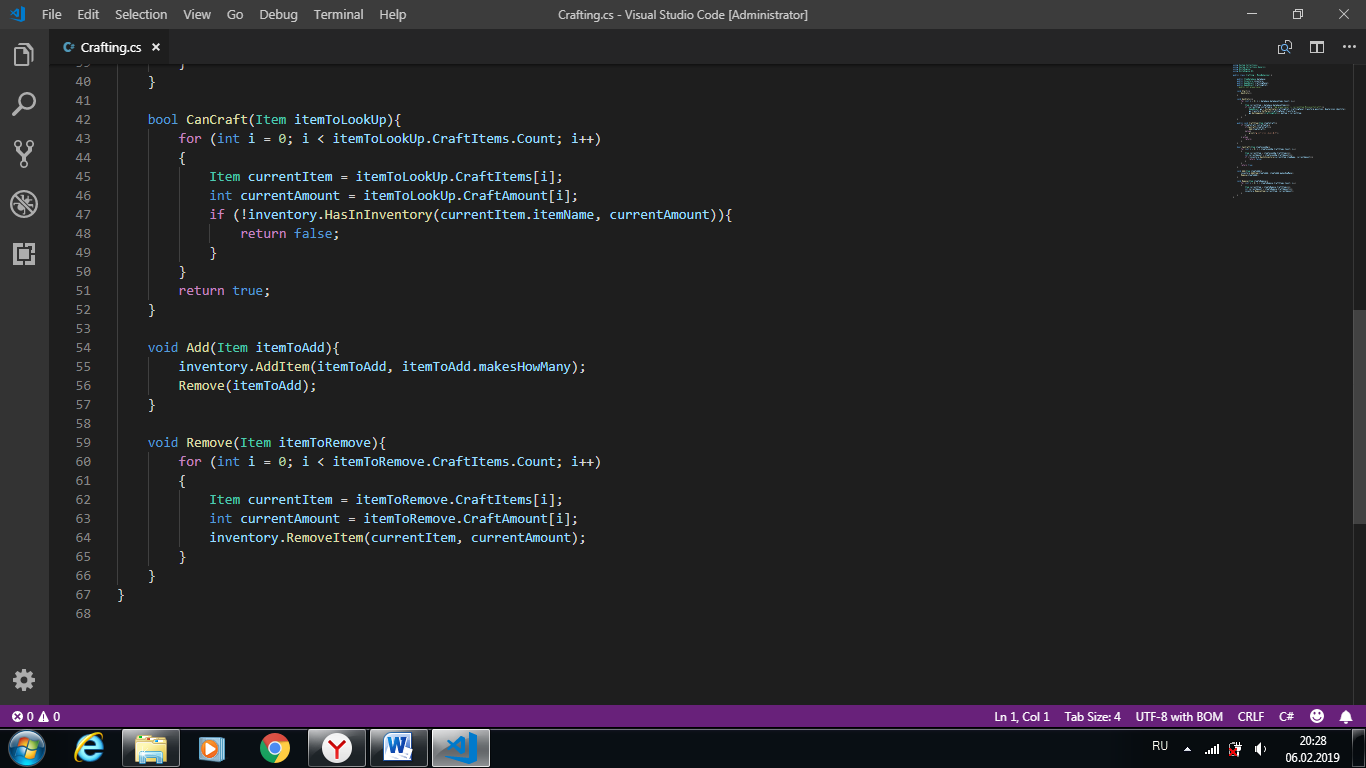


Метод CanCraft – Для каждого предмета нужного для крафта делает проверку на наличие его в инвентаре и записывает в переменную lack недостаток. Выдаёт true если недостатка нет и false если есть.



Метод Add – добавляет в инвентарь созданный предмет с помощью метода AddItem из скрипта Inventory и выполняет метод Remove(описанный ниже).

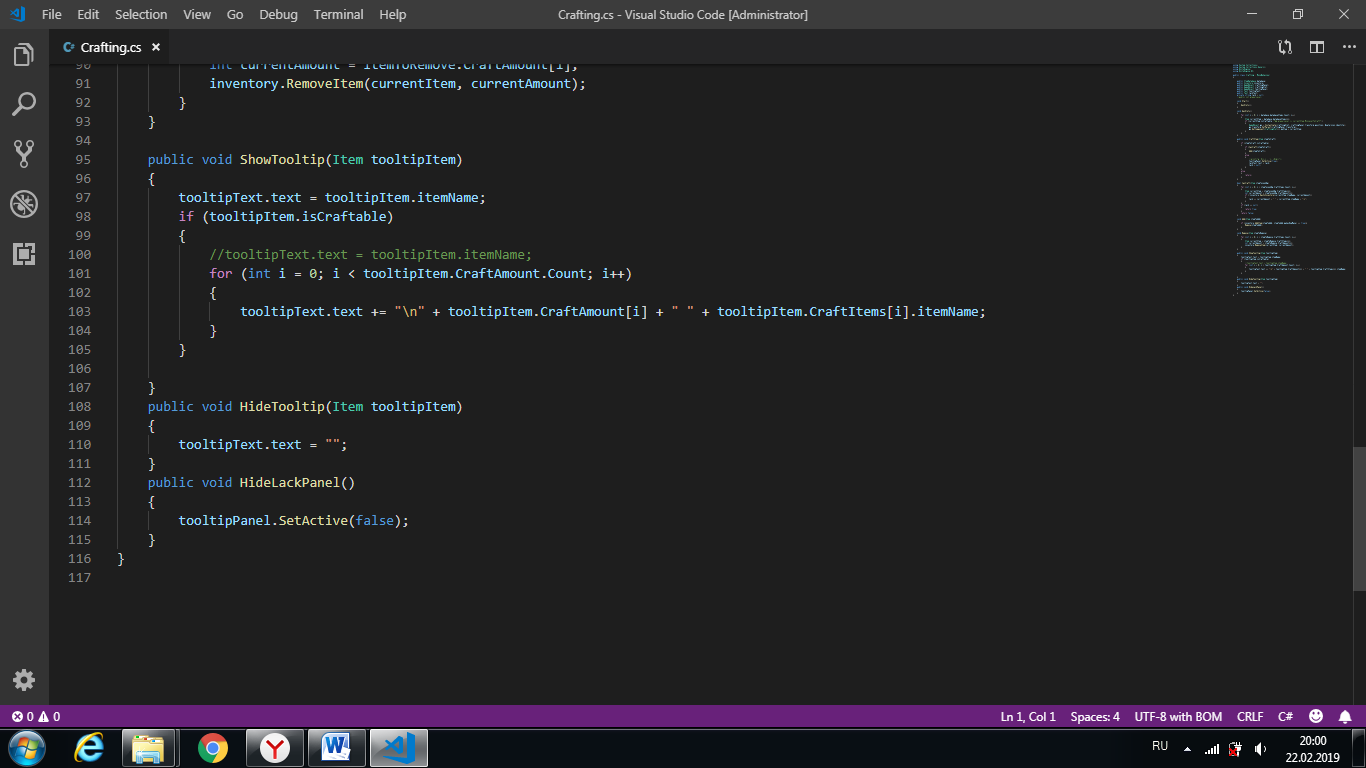
Метод Remove - убирает из инвентаря требуемые для создания предмета ингридиенты используя метод RemoveItem из скрипта Inventory.



Метод ShowTooltip – Показывает на панели подсказок имя предмета и нужные ингредиенты для его создания, если он является создаваемым.

Метод HideTooltip – Очищает панель подсказок.

Метод HideLackPanel – вызывается при нажатие на кнопку Ok на панели недостатка появляющуюся при недостатке предметов для крафта .



Система диалогов

Все диалоги хранятся в XML файлах.

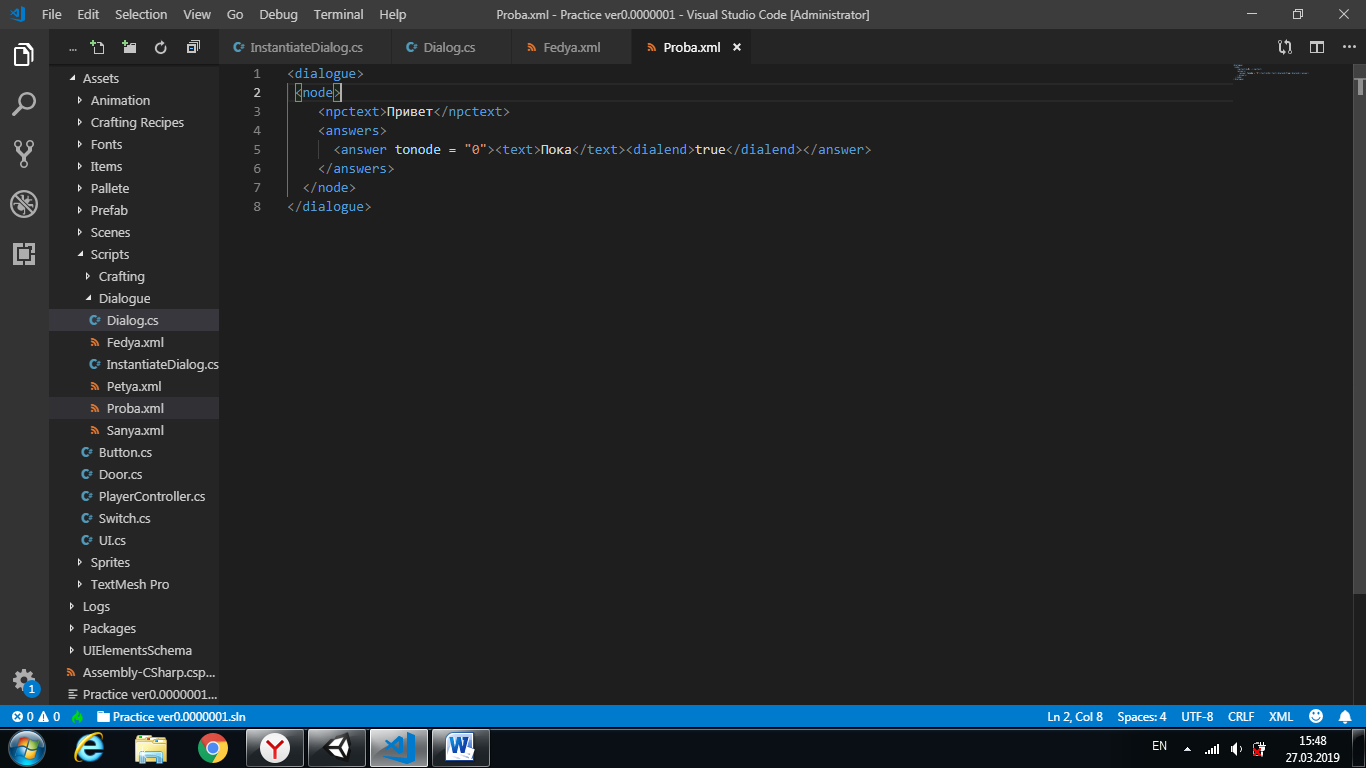


Рисунок 1

XML файл (рис.1) внутри представляет собой корневой путь «dialogue» в котором находятся ноды(«node»). нод - состоит из фразы NPC(«npctext») и вариантов ответа главного героя(«answer»), которые в свою очередь могут поменять текущий нод или закончить диалог.

Скрипт Dialog

На рисунке 2 показан конструкт который производит сериализацию и десериализацию(перевод из XML в C# и обратно). TextAsset в данном случае это XML файл.

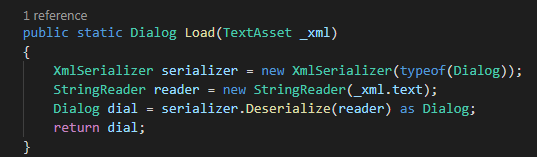


Рисунок 2

На рисунках 3,4,5 показано как происходит связывание элементов, атрибутов из XML файла с классами, массивами, переменными из C#.

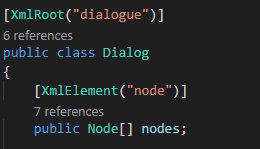


Рисунок 3

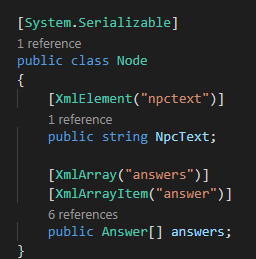


Рисунок 4

У ответов персонажа есть атрибуты и елементы, отвечающие за некоторые функции.

«tonode» - меняет текущий нод.

«text» - в нём записывается текст.

«dialend» - обозначает конец диалога.

«questvalue» - обозначает текущий статус квеста.

«needquestvalue» - обозначает нужный статус квеста.

«questname» - название квеста.

«rewardgold» - золота получено.

«needgold» - золота нужно.

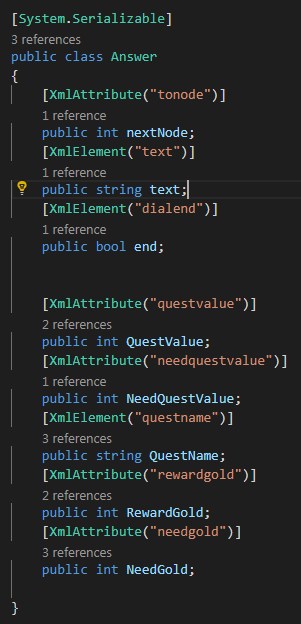


Рисунок 5

Скрипт InstantiateDialog

Это скрипт, который прикрепляется к NPC.

Переменная ShowDialogue отвечает за показ диалогового окна.

При старте в переменную dialog переносится вся информация из XML файла.

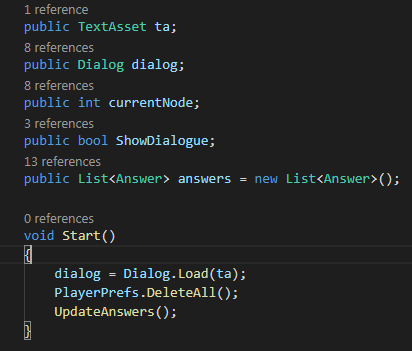


Рисунок 6

Для визуализации диалогового окна используется GUI. В Box будет помещено всё диалоговое окно. В Label речь NPC. Под Label генерируются Button равные количеству ответов игрока, а также присваивается текущий нод(«currentNode») и вызывается метод UpdateAnswers описанный ниже.

Во время генерации кнопок происходят проверки:

1. Если квест начат(«questvalue» > 0) то связывается имя квеста и его текущий статус.
2. Закончить диалог если присутствует атрибут «dialend».
3. Добавить золото в кошелёк персонажа, если присутствует атрибут «rewardgold».
4. Отнять золото из кошелька, если присутствует атрибут «needgold».

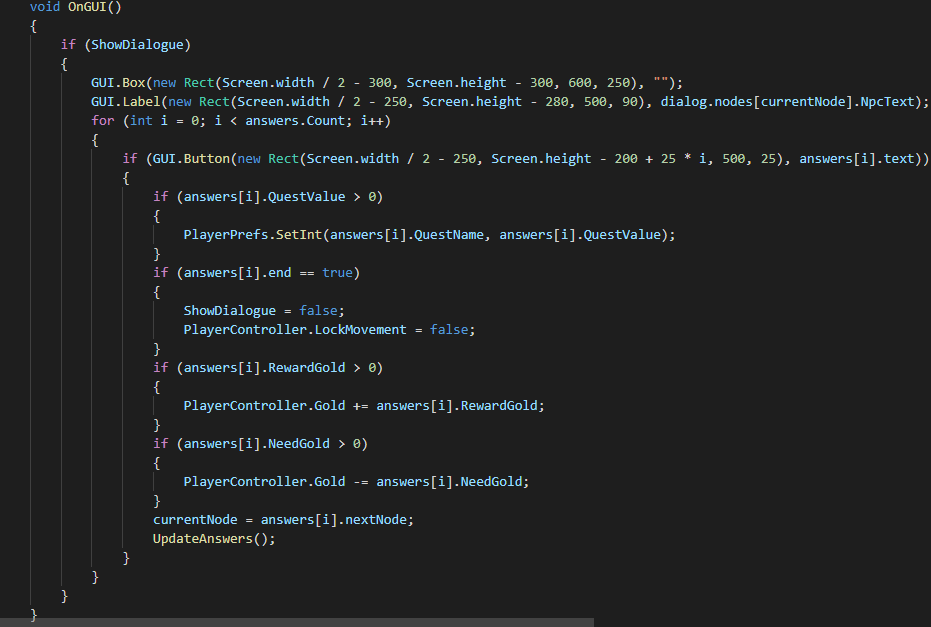


Рисунок 7

Метод UpdateAnswers

Проверяет все условия на появление скрытых ответов персонажа, при положительном результате обновляет диалоговое окно.

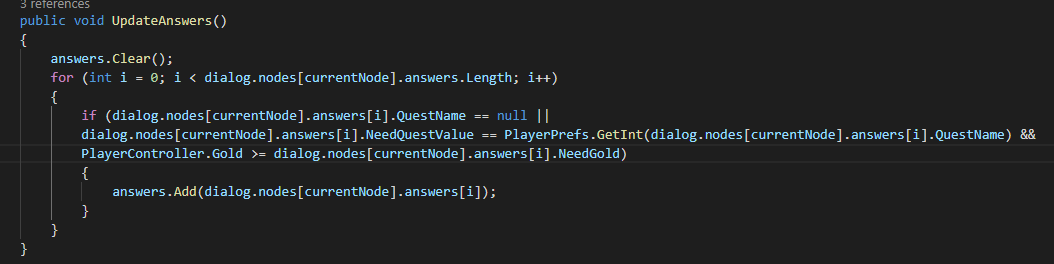


Рисунок 8

Классификация предметов

В данном случае предметы классифицируются по имени(itemName), Помещается ли в один слот больше одного предмета(isStackable), Возможно ли создание этого предмета(isCraftable).

