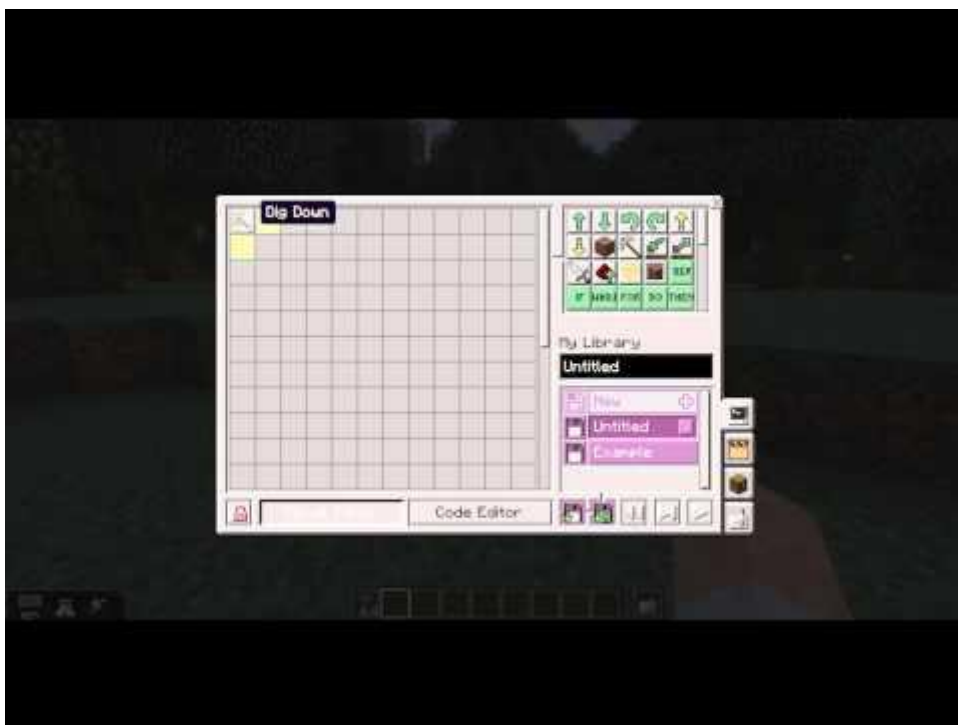


Программирование с помощью ComputerCraftEdu

ComputerCraftEdu делает простым программирование. Существует несколько способов создания кода: перетаскивание, щелчок по сетки, ввод с клавиатуры и редактор кода. Черепахи - мощные существа, которые могут делать большинство вещей, которые могут делать игроки, эта статья будет охватывать основные команды, доступные в визуальном редакторе.

[Видео](#)

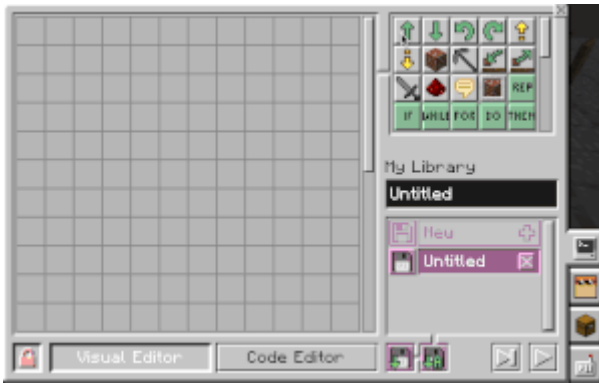


Параметры ввода

Существует несколько способов создания авторских программ в ComputerCraftEdu, каждый из которых ближе к написанию фактического кода, чем последний:

Перетаскивание

Перетаскивание команд из панели инструментов - это самый простой способ записи программ. Просто нажмите на плитку, которую вы хотите добавить, а затем, где вы хотите поместить ее в сетку. Панель инструментов справа включает в себя все возможные команды в визуальном редакторе.



Щелчок сетки

Вместо перетаскивания плиток вы можете щелкнуть по выделенным желтым и красным квадратам на сетке. Это вызывает выбор команд, которые могут использоваться на конкретном квадрате. Это также работает при использовании циклов и условий: нажатие на красный квадрат приводит к возможным вариантам.

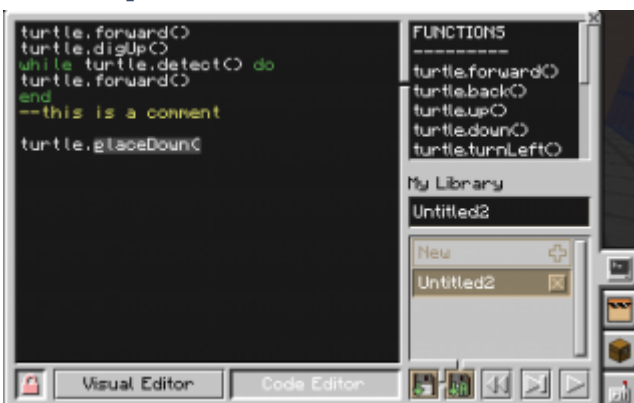
Ввод клавиатуры

Вы можете использовать клавиатуру для навигации по сетке. Клавиши со стрелками перемещают выделенный квадрат вокруг сетки. Вы можете нажать TAB на желтом и красном квадратах, чтобы открыть выбор возможных команд. Используйте клавиши со стрелками для навигации по параметрам и выберите с помощью TAB.

Другие команды клавиатуры:

- ENTER,- добавить новую строку перед выделенным квадратом
- SPACE, - добавить новое пространство перед выделенным квадратом
- BACKSPACE, -удалить пробел (и возможные команды) перед выделенным квадратом

Редактор кода



Редактор кода и автозаполнение

Редактор кода позволяет вам давать команды черепах в реальном Lua-скрипте. Чтобы помочь вам в этом, редактор имеет функцию автозаполнения, которая предложит возможные команды на основе того, что вы начали писать.

- Чтобы принять предложение, нажмите TAB
- Для прокрутки между опциями используйте клавиши UP и DOWN
- Продолжайте печатать, игнорируя предложения

Пространство в правом верхнем углу экрана отображает список общих команд черепахи. Проверьте [регулярные ComputerCraft-вики](#) для получения дополнительной информации!

Команды черепахи

Основные команды

Команды перемещения

Команды перемещения перемещают и поворачивают вашу черепаху в указанном направлении. Они используют черепаху как точку отсчета. В визуальном редакторе они могут быть сложены, чтобы черепаха повторяла команду несколько раз (до 64).



Иконы команд перемещения

Соответствующие Lua-команды

- `turtle.moveForward()`
- `turtle.moveBack()`
- `turtle.moveUp()`
- `turtle.moveDown()`
- `turtle.turnLeft()`
- `turtle.turnRight()`

Копание и строительство



Копирование и создание значков в визуальном редакторе

Если у вашей черепахи есть инструмент, он может копать и строить. Черепахи могут прорыть почти любой материал в игре (исключая несколько [блоков MinecraftEdu](#)).

Когда вы используете команду `digging` в визуальном редакторе, вам нужно будет выбрать направление копания (вперед, вверх или вниз). Это можно изменить позже, щелкнув правой кнопкой мыши команду или выбрав команду вверх и поместив ее в одну и ту же плитку. Блоки, которые собирает черепаха, появятся на вкладке инвентаря.

Соответствующие Lua-команды

- `turtle.dig()`
- `turtle.digUp()`
- `turtle.digDown()`

Черепахи также могут размещать блоки. Им нужны материалы в инвентаре для создания чего-либо. По умолчанию черепаха будет использовать предметы из первого слота инвентаря. Это можно изменить с помощью команды «выбрать слот». Когда вы используете команду `place` в визуальном редакторе, вам нужно будет выбрать направление размещения (вперед, вверх или вниз). Это можно изменить позже, щелкнув правой кнопкой мыши команду или выбрав команду вверх и поместив ее в одну и ту же плитку.



Черепахи копают и строят

Соответствующие Lua-команды

- `turtle.place()`
- `turtle.placeUp()`
- `turtle.placeDown()`

Расширенные команды

Поднимите и опустите

Черепашки могут забрать предметы и бросить их. Когда вы выберете любую команду в визуальном редакторе, вам нужно будет выбрать направление отбрасывания или подбора (вперед, вверх или вниз). Когда вы сообщите своей черепахе опустить предмет, он выведет элемент из активного слота инвентаря (по умолчанию первый).

Примечание. Вы также можете использовать эти команды для приема или размещения предметов в сундуке и из него!

Соответствующие Lua-команды

- `turtle.place()`
- `turtle.placeUp()`
- `turtle.placeDown()`

Атака



Значок атаки в визуальном редакторе

Если у вашей черепахи есть инструмент, он может атаковать. Когда вы используете команду атаки в визуальном редакторе, вам нужно будет выбрать направление атаки (вперед, вверх или вниз).

Соответствующие Lua-команды

- `turtle.attack()`
- `turtle.attackUp()`
- `turtle.attackDown()`

Выбор слота



Выбор номера слота в визуальном редакторе

Многие команды и условия, такие как «место» и «сравнение», используют слоты инвентаря. Они всегда используют активный слот (по умолчанию первый слот, обозначенный серой рамкой вокруг него). Вы можете изменить слот, используя

команду «Выбрать слот». Когда вы выбираете команду в визуальном редакторе, вам предлагается указать, какой слот вы хотите активировать. Это можно сделать, выбрав число или переменную или некоторые другие средства.

Соответствующие Lua-команды

- `turtle.select(number)`

Установите redstone

Черепашки могут взаимодействовать с redstone. Они могут излучать сигнал красного цвета, и когда вы выбираете команду в визуальном редакторе, вам нужно указать, будет ли черепаха включать сигнал (true) или off (false).



Черепашка активирует красный камень

Соответствующие Lua-команды

- `turtleedu.setRedstone(true/false)`
- `turtleedu.setRedstoneUp(true/false)`
- `turtleedu.setRedstoneDown(true/false)`

Комментарий

Как следует из названия, команда «Комментарий» добавляет комментарии к вашей программе. Они облегчают другим людям понимание программ, которые вы пишете. Они не влияют на то, как программа работает в любом случае.

Комментарии представлены в Lua as

- `-- "Your comment here"`

Циклы и if-statement



Циклы позволяют вам повторять команды на основе условия. Условия могут быть любыми от переменных до осмотра окружения черепах. Чтобы помочь вам использовать их, в визуальном редакторе в ComputerCraftEdu есть встроенный синтаксис. Когда вы перетаскиваете зеленый элемент синтаксиса в сетку, красные поля предлагают то, что вам нужно добавить, чтобы закончить цикл.

Повторение



Пример цикла повторения

Повторение - это самый простой цикл в игре: кроме нескольких элементов синтаксиса, ему нужно только количество повторений для работы. Поместите команды, которые вы хотите повторить на линии (линиях) между «do» и «end».

Повторить цикл в коде Lua:

- `for _n = 1 , 4 do insert commands here end`

В то время как

While-loop повторяет команды до тех пор, пока выполняется условие. Вы также можете добавить несколько альтернативных условий. Поместите команды, которые вы хотите повторить на линии (линиях) между «do» и «end».

Пока цикл в коде Lua:

```
while condition do
```

```
insert command here
```

```
end
```

Для

For-loop - более сложная версия «repeat». Создав переменную в качестве счетчика и установив предел, программа повторяет команды, каждый раз рассчитывая на ограничение.

Для цикла в коде Lua:

```
for x=1,10 do
```

```
insert command here
```

```
end
```

Если

If-statement передает программу: если условие выполнено, программа выполняет определенные команды, если программа не переходит после «конца». Операторы if могут также иметь альтернативные команды для выполнения, если условия не выполняются: «else» и «else-if» добавить дополнительные опции в программу.

If-statement в коде Lua:

```
if condition then
```

```
insert command here
```

```
else if
```

```
insert command here
```

```
else
```

```
insert command here
```

```
end
```

условия

Условия могут использоваться в сочетании с другими командами. Они определяют, что-то делается или нет.

детектировать

Обнаружить проверки, есть ли блок перед, над или под черепахой. Он не обнаруживает монстров, животных или жидкостей.

сравнить

Сравните проверки, если блок перед или над черепахой совпадает с тем, который находится в слоте активного инвентаря.



Черепаха, проверяющая блоки для золотой руды

Осмотреть блок / элемент

Проверяйте блокировку, если блок перед или над черепахой совпадает с блоком, выбранным вами в качестве игрока. Вы можете ввести имя любого блока, который вы хотите. Это работает для элементов, также с условием проверки предмета.

Обнаружение красного камня

Черепаха проверяет сигнал красного цвета в блоке перед, над или под ним.

Переменные и прочие

Переменные, числа, логические операторы и математические операторы составляют остальные доступные команды. Они обычно используются в сочетании с условиями и циклами.

Переменные и числа



Настройка переменной в визуальном редакторе

Вы можете использовать переменные как условия. Они могут использоваться как счетчики (например, проверка расстояния, количество раз).

Прежде чем использовать переменную, вам нужно создать ее (по умолчанию в Repeat- и For-Loops в визуальном редакторе это делается). Чтобы создать переменную, вам нужно указать ее базовое значение. Обычно это делается с помощью числа.

Все переменные являются локальными, то есть они могут быть доступны только в рамках программы, в которой они появляются.

Логические утверждения

Вы можете добавить несколько, альтернативных или отрицательных условий с помощью логических операторов. **AND**, **OR** И **NOT** могут использоваться между условиями и доступны через предложения.

Математические операторы

Для сравнения можно использовать математические операторы: вы можете проверить, больше ли количество элементов в слоте больше числа.

Доступные операторы

- equals (=)
- does not equal (≠)
- is greater than (>)
- is greater than or equal to (≥)
- is less than (<)
- is less than or equal to (≤)

Существуют также математические функции, которые можно использовать с переменными. Доступны основные математические функции (сложение, вычитание, умножение и деление).

Булевы операторы



Булевы переменные могут быть либо истинными, либо ложными. Попробуйте проверить, есть ли перед черепахой блок с командой Detect. Это утверждение является либо истинным, либо ложным. Вы можете использовать логические значения, сравнивая их с условием.