

Завдання №1. Перевірка уведених даних.

Користувачеві пропонується ввести числові значення за допомогою `input()` в програмі Python, які присвоюються відповідній координаті гравця Minecraft. Проаналізувати різницю між командами `postToChat()` та `print()`. Перевірити введені дані.

Приклад вікна із завданням зображений на рис.2.1.а та рис.2.1.б

Забезпечити:

1. Перевірка уведеного значення Y.

```
Введіть значення X = 1000000
Введіть значення Y = 100
Введіть значення Z = 20000
Гравця телепортовано на координати 1000000 100 20000
```

Рис. 2.1.а. Приклад виконання завдання



Рис. 2.1.б. Приклад виконання завдання в Minecraft

Наведемо приклад завершеної програми, яку повинен зробити користувач після виконання завдання №1:

```
import mcpi.minecraft as minecraft
mc = minecraft.Minecraft.create()
x = int(input("Введіть значення X = "))
y = int(input("Введіть значення Y = "))
while y<0:
    print("Невірне значення Y")
    y = int(input("Введіть значення Y = "))
z = int(input("Введіть значення Z = "))
mc.player.setPos(x,y,z)
mc.postToChat("Teleported to "+str(x)+" "+str(y)+" "+str(z))
print("Гравця телепортовано на координати ",x,y,z)
```

Завдання №2. Знайти помилку у коді

Дано фрагмент кода (рис 2.2), в якому потрібно знайти помилку, яка заважає роботі програми. Наприкінці потрібно отримати результат – повідомлення в консолі «Все працює». Код зображений на рис.2.2.

Забезпечити:

Оцінювання правильності виконаного завдання таким чином, щоб при старті правильного коду виводилося повідомлення в консоль Python з текстом «Все працює!».

Рекомендації:

Використати перевірку типу змінної за допомогою `type()`.

```
import mcpi.minecraft as minecraft
mc = minecraft.Minecraft.create()

x = 100
y = 200
z = "30"

mc.pleyp.setPos(str(x),z,c)
print("Все працює!")
```

Рис. 2.2. Початкова форма завдання

Завдання №3. Змінити позицію гравця на випадкове значення

Створити програму, яка змінює позицію гравця на випадкове значення 3 рази з інтервалом 10 секунд. Приклад виконання завдання зображений на рис.2.3..

Рекомендації:

Для виконання завдання доцільно скористатися:

Модуль random, в якій знаходиться функція randint(a,b);

Модуль time, в якій знаходиться функція sleep(m).

Завдання №4. Вивести координати гравця

У наданому коді (рис. 2.4) потрібно додати команди для виведення координат гравця, отриманих за допомогою `getTilePos()`.

Забезпечити:

Оцінювання правильності виконаного завдання таким чином, щоб при запуску правильного коду виводилося 3 повідомлення в грі з координатами гравця.

Рекомендації:

Скористатися командами команд `getTilePos()` та `postToChat()`;

Використовувати змінну `pos` для створення списку координат гравця.

```
import mcpi.minecraft as minecraft
mc = minecraft.Minecraft.create()

pos= mc.player.getTilePos()
x = pos.x
y = pos.y
z = pos.z
```

Рис. 2.4. Приклад коду до завдання 4

```
import mcpi.minecraft as minecraft
from random import *
from time import *
mc = minecraft.Minecraft.create()

x = randint(-100000,100000)
y = randint(70,200)
z = randint(-100000,100000)
mc.player.setPos(x,y,z)
sleep(10)
x = randint(-100000,100000)
y = randint(70,200)
z = randint(-100000,100000)
mc.player.setPos(x,y,z)
sleep(10)
x = randint(-100000,100000)
y = randint(70,200)
z = randint(-100000,100000)
mc.player.setPos(x,y,z)
sleep(10)
```

Рис. 2.3. Приклад програми до завдання 3

Наведемо приклад програми, яку отримає користувач після реалізації

завдання 4:

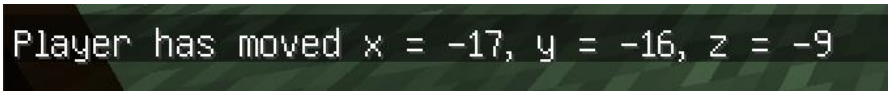
```
import mcpi.minecraft as minecraft
mc = minecraft.Minecraft.create()
pos= mc.player.getTilePos()
x = pos.x
y = pos.y
z = pos.z
mc.postToChat(x)
mc.postToChat(y)
mc.postToChat(z)
```

Завдання №5. Відстежити відстань, яку проходить гравець за 20 секунд

Створити програму, за допомогою якої можна відстежити відстань, яку проходить гравець за 20 секунд вільної гри в Minecraft. Приклад виконаного завдання зображений на рис. 2.5..

Виконати завдання таким чином:

1. Створити змінну pos1 з координатами гравця при запуску програми.
2. Записати окремо кожен координату в змінну(X1,Y1,Z1).
3. Використати модуль time з функцією sleep().
4. Створити змінну pos2 з координатами гравця.
5. Записати окремо кожен координату в змінну(X2,Y2,Z2).
6. Створити нові змінні – різниці координат, від координати з pos1 відняти координату з pos2. Приклад : $distX=x2-x1$;
7. Вивести в гру кожен різницю координат.



```
Player has moved x = -17, y = -16, z = -9
```

Рис. 2.5. Приклад виконаного завдання 5 в Minecraft

Завдання №6 Змінити координати гравця

Користувачеві пропонується ввести, яку координату потрібно змінити та значення, на яке змінити. Це значення присвоюється відповідній координаті гравця Minecraft.

Рекомендації:

Висвітлення зміни координати гравця відносно свого місцезнаходження та різниці здійснюється за допомогою команд `getTilePos()` та `getPos()`.

Приклад коду зі завданням:

```
import mcpi.minecraft as minecraft
mc = minecraft.Minecraft.create()
pos= mc.player.getPos()
x = pos.x
y = pos.y
z = pos.z
a = input()
if a=='x':
    mc.player.setPos(x+int(input()),y,z)
elif a=='y':
    mc.player.setPos(x,y+int(input()),z)
else:
    mc.player.setPos(x,y,z+int(input()))
```

Завдання №7. Створення нового блоку

Користувачеві пропонується ввести ідентифікатор блоку в програмі Python. Після цього, в світі Minecraft, поруч з гравцем, створиться блок згідно цього ідентифікатору. Переглянути параметри команди Приклад виконаного завдання зображений на рис. 2.6..

Рекомендації:

Використати список ID блоків.



Рис. 2.6. Приклад виконаного завдання 7 в Minecraft

Завдання №8. Створення бар'єру

Створити програму, яка створює навколо гравця бар'єр висотою 1 блок з дерева. Приклад виконаного завдання зображений на рис. 2.7.

Виконати завдання таким чином:

1. Створити змінну pos з координатами гравця.
2. Записати окремо кожен координату в змінну (X,Y,Z).
3. Додати 8 команд setBlock(x,y,z,5,0), де x,y,z – відповідно будуть змінюватись згідно позиції поруч з гравцем.

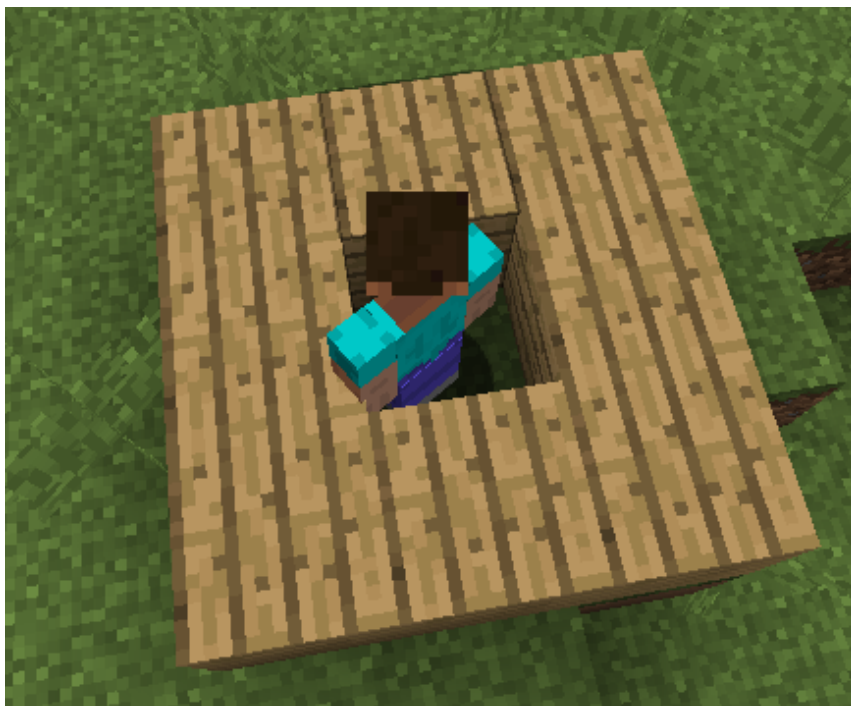


Рис. 2.7. Приклад виконаного завдання 8 в Minecraft

Завдання №9. Побудова вежі з каменя

Створити програму, яка створює поруч з гравцем вежу з каменя висотою 10 блоків. Приклад виконаного завдання зображений на рис. 2.8.

Виконати завдання таким чином:

1. Створити змінну pos з координатами гравця.
2. Визначити цикл for, що виконує 10 ітерацій
3. Додати в цикл команду створення нового блока, в якій до координати Y слід додати змінну циклу



Рис. 2.8.6. Приклад виконаного завдання 9 в Minecraft

Приклад коду зі завданням:

```
import mcpi.minecraft as minecraft
mc = minecraft.Minecraft.create()
pos= mc.player.getPos()
for i in range(10):
    mc.setBlock(pos.x,pos.y+i,pos.z,1,0)
```

Завдання №10. Побудова вежі з випадкових блоків

Створити програму, яка створює поруч з гравцем вежу висотою 30 випадкових блоків. Приклад виконаного завдання зображений на рис. 2.9.

Рекомендації:

Для виконання завдання доцільно скористатися циклом for.

Додаткове завдання:

Додати у цикл for команду sleep(1). Відстежити зміну значення змінних виконання програми в грі;

Змінити умову коду, додати можливість побудові вежі одного кольору. Запрограмувати вибір способу побудови користувачем.



Рис. 2.9. Приклад виконаного завдання 10 в Minecraft

Завдання №11 Побудова куба

Користувачеві пропонується ввести розмір куба та ідентифікатор блока в программі Python. Ці значення використовуються в команді `setBlocks()`. Висвітлення використання команди `setBlocks()` та команди `map()`. Приклад вікна із завданням зображений на рис. 2.10.



Рис 2.10. Приклад виконаного завдання 11 в Minecraft

Завдання №12 Додавання умови в код

Дано код, який працює - з'єднання з Minecraft присутнє, в чаті гри друкується повідомлення «Code completed». Але більше нічого не з'являється. Створіть програму, в якій буде використовуватись перевірка на те, що гравець в повітрі чи на землі.

Рекомендації:

1. Використати команду `minecraft.player.getPos()` для отримання поточної позиції гравця.
2. Використати команду `minecraft.getBlock()` для отримання ідентифікатора блоку в якому знаходиться гравець.
3. Використати команду `minecraft.postToChat()` для друку повідомлення в чат гри.

Завдання №13 Створення квадрату

В бібліотеці `mcpi` є чудова команда `setBlocks()`, яка будує будь-який паралелепіпед від двох точок. Але за допомогою цієї команди неможливо створити пустий квадрат, в якого буде видно лише його сторони. Створіть програму, за допомогою якої можна позбавитись цієї проблеми.

Виконати завдання таким чином:

1. Відкрийте шаблон файлу «з'єднання з грою Minecraft».
2. Створіть змінну `pos` з координатами гравця при старті програми (використайте команду `minecraft.player.getPos()` для отримання значень).
3. Побудуйте квадрат за допомогою команди `minecraft.setBlocks()`. Використайте змінну `pos` як першу точку, в другій змініть лише пару `XY` або `YZ` – для досягнення бажаного результату. Можна використати будь який матеріал.
4. Побудуйте ще один квадрат, але тепер з іншими точками. Зробіть зрушення по осям квадрату (значення в команді `minecraft.setBlock()` які змінюються користувачем) на один блок, в першій точці в плюс, в другій в мінус. В якості матеріалу обов'язково виберіть повітря.
5. Перевірте результат, у вас повинно вийти квадрат, в якого видно лише сторони. Якщо зовсім зник квадрат – поміняйте місцями віднімання та додавання в другому квадраті з повітря.

2.3. Розробка ігрових ситуацій для школярів в середовищі Minecraft

Завдання №1 «Неймовірні пригоди»

Стів трохи занудьгував. Він не може знайти жодного цікавого містечка по всьому світу Minecraft. Допоможіть йому. Напишіть код, в якому потрібно телепортувати Стіва в цікаві місця кожну хвилину. Координати цих точок спочатку потрібно вам самим відшукати (рис. 2.11.), тому зупинимось на п'яти цікавих місцевостях. Програму можна зробити більш хаотичною, якщо використовувати бібліотеку random замість заданих координат.



Рис. 2.11. Приклад цікавої локації до завдання «Неймовірні пригоди»

Завдання №2 «Супер сила»

Всі мріють стати супергероєм, Алекс також не виняток. Але все що вона може робити, це створити випадково пожежу в своєму лісі. Давайте зробимо їй сюрприз. Створіть програму, яка підкине її на 100 блоків вгору, по 10 блоків кожну секунду. Після цього програма поставить під нею блок скла, щоб вона змогла побачити всі простори під собою (рис. 2.12.). Нехай відчує себе справжнім супергероєм!



Рис. 2.12. Приклад фінальної стадії завдання «Супер сила»

Завдання №3 «Блоки за секунду»

Всі знають швидкість вагонетки, коня, свині але ніхто не знає швидкість Стіва в Minecraft! Виправте ситуацію, напишіть код, який буде виводити результат підрахунків - скільки блоків пройшов Стів по прямій за одну хвилину. Результат зображений на рис. 2.13.

Приклад виконаного завдання №3:

```
from mcpi.minecraft import Minecraft
import time
mc = Minecraft.create()
pos = mc.player.getPos()
time.sleep(60)
newPos = mc.player.getPos()
difference = abs(pos - newPos)
mc.postToChat("Speed - " + difference)
```



Рис. 2.13. Приклад виконання завдання «Блоки за секунду»

Завдання №4 «Безпека понад усе»

Інколи трапляються ситуації, коли дещо пішло не по плану. Наприклад, у процесі шахтарства, Стів провалився в лаву, або під час подорожі, Алекс забула, що не вміє плавати та почала тонути в воді. Треба цього уникнути! Напишіть код, який буде завжди перевіряти, де знаходиться гравець. Якщо він впав на один блок в лаву, то підніміть його на два блоки верх, та поставте під ним блок, бажано непалаючий. Якщо ж гравець занурився в воду на три блоки, то підніміть його на чотири блоки, і також розмістіть, як в ситуації з лавою, під ним блок, можна будь-який. Результат зображений на рис. 2.14.

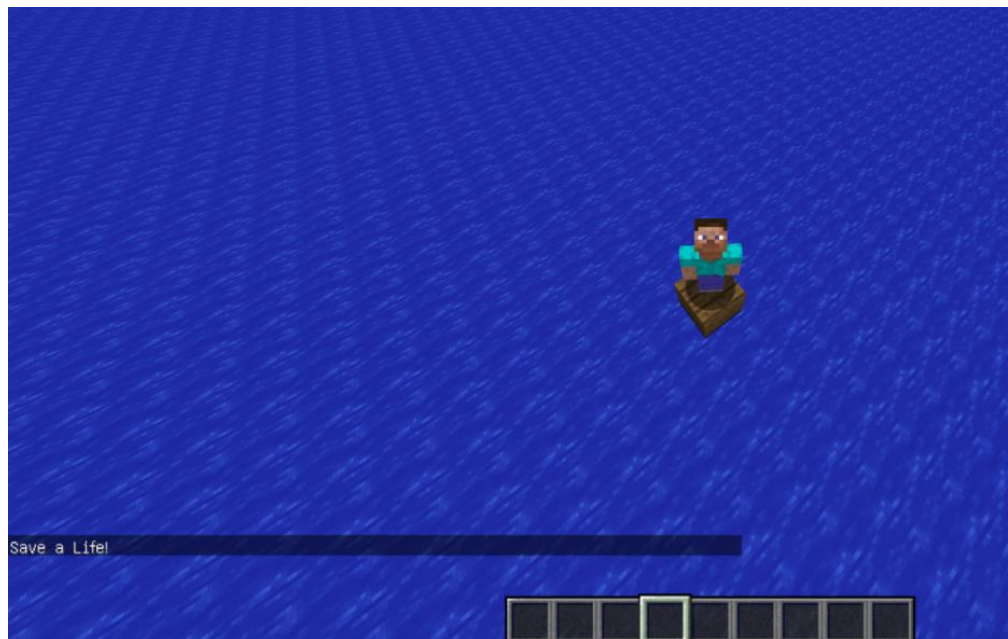


Рис. 2.14. Приклад виконаного завдання «Безпека понад усе»

Завдання №5 «Копач»

Стів втомився шукати золоту руду для своєї нової кирки. Щоб зберегти час від шахтарства, напишіть код, який дізнається, чи є в землі під гравцем будь-яка руда(залізо, золото, алмаз). Програма отримує поточні координати гравця і по одному перебирає блоки під ногами, перевіряючи, чи не є вони рудою. Якщо знайдеться задовільний блок або блоки, повідомте Стіва де він знаходиться. Результат зображений на рис. 2.15.

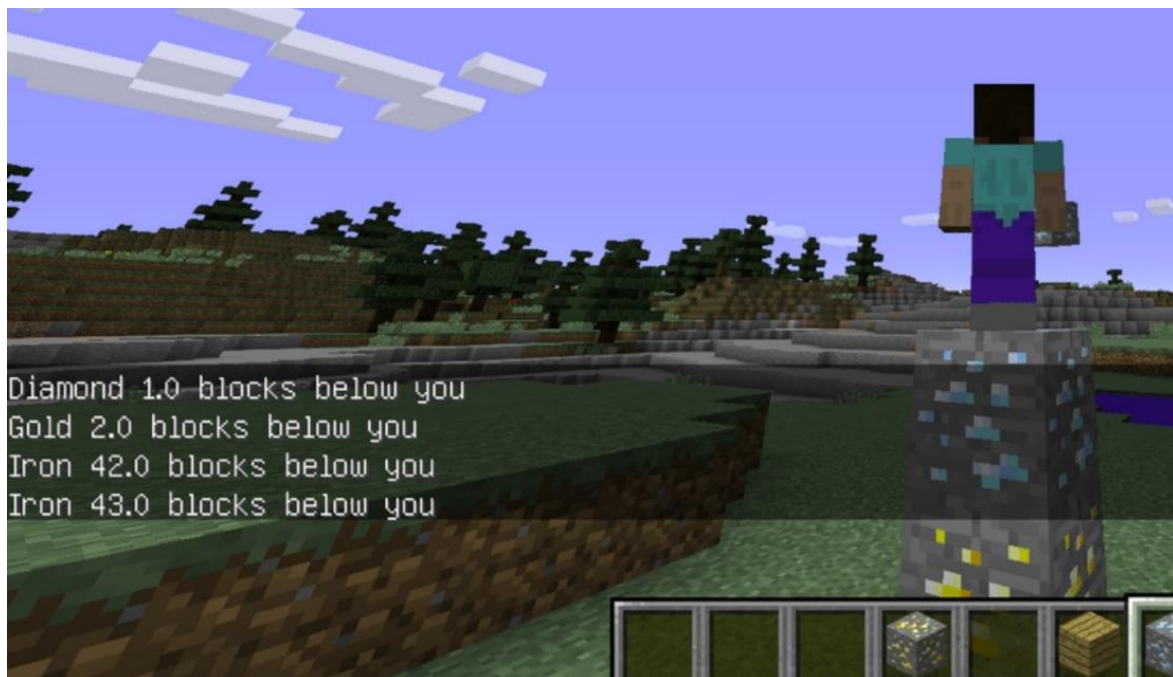


Рис. 2.15. Приклад виконаного завдання «Копач»

Завдання №6 «Шахтар-експерс»

Всі гравці люблять мандрувати по світу Minecraft, незалежно де, в болотах, в шахтах, в глибинах океану. Але є одна прикра річ, відстань, інколи доводиться проходити значний шлях. Спробуймо це подолати! Створіть код, який буде будувати автоматичну колію для вагонеток. Так як, всі світи дуже привабливі і Стів хоче всюди побувати, вам потрібно поміркувати як зробити колію придатну для підземних та підводних пригод. Приклад виконання зображено на рис. 2.16.

Фрагмент виконаного завдання №6:

```
def railLong(n):
    p = mc.player.getPos()
    for i in range(n):
        mc.setBlock(p.x,p.y ,p.z+i,idRoad)#дорога
        mc.setBlock(p.x,p.y+1 ,p.z+i,idRail) #колія
        mc.setBlock(p.x,p.y+2 ,p.z+i,idAir)#повітря
        mc.setBlock(p.x,p.y+3 ,p.z+i,idAir)#повітря
        mc.setBlock(p.x,p.y+4 ,p.z+i,idAir)#повітря
        mc.setBlock(p.x,p.y+5 ,p.z+i,idAir)#повітря
        mc.setBlock(p.x,p.y+6 ,p.z+i,idBlock)#верхній люк
        mc.setBlock(p.x-1,p.y ,p.z+i,idLamp)#стіна 1
```

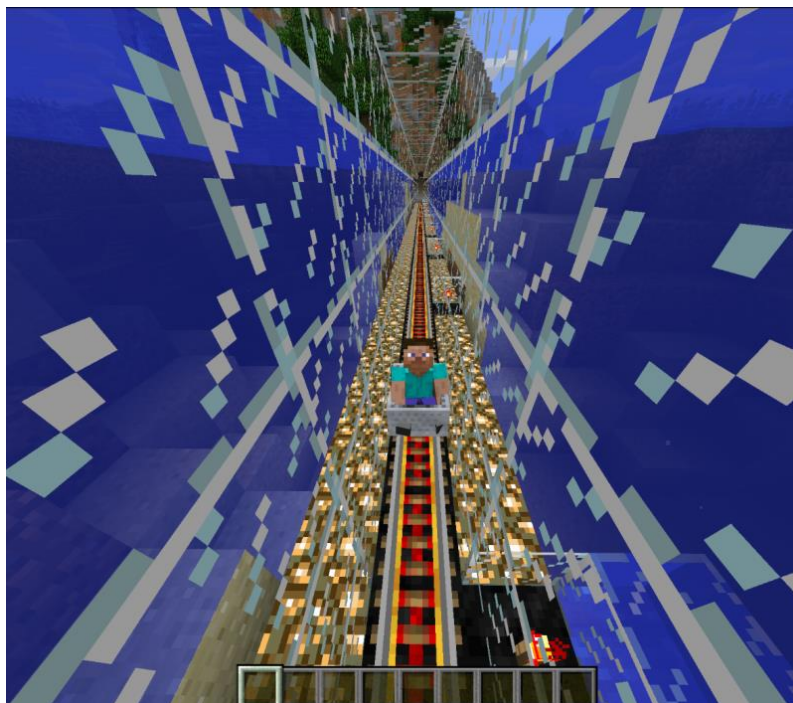


Рис. 2.16. Приклад виконаного завдання «Шахтар-експерс»

Завдання №7 «Неприступна фортеця»

Незабаром настане темрява! Стів та Алекс загубили свою домівку, а блоків для побудови нового будинку в них зовсім немає. Починають з'являтися зомбі, і без нашої допомоги вони не впораються. Створіть код, який побудує захисну оселю для Стіва та Алекс. Параметри оселі: будинок має бути розміром 5 x 5 x 5 блоків, матеріал стіни – обсидіан, мати вхід з дверима та принаймні 1 віконце для того, щоб дізнатись чи є небезпека зовні. Приклад виконання зображено на рис. 2.17.

Фрагмент виконаного завдання №7:

```
from mcpi.minecraft import Minecraft
mc = Minecraft.create()
p=mc.player.getPos()
x=p.x
y=p.y
z=p.z
mc.setBlocks(x, y, z, x+5,y+5,z+5,5)
mc.setBlocks(x+1,y+1,z+1,x+4, y+4, z+4, 0)
mc.setBlocks(x, y+5,z, x+5,y+5,z+5,20)
mc.setBlocks(x, y, z, x+5,y, z+5,20)
mc.setBlock(x+2,y,z,1)
mc.setBlock(x+2,y+1,z,64,1)
mc.setBlock(x+2,y+2,z,64,8)
```



Рис.2.17. Приклад виконаного завдання «Неприступна фортеця»

Завдання №8 «Фобія темряви»

«Фобія темряви» Після останнього візиту до лікара Стів точно знає, що в нього ніктофобія (страх перед тьмою). Та тепер усіма силами намагається уникати темряву будь-де, навіть в своїй домівці! Допоможіть, будь-ласка, Стіву з цією кепською ситуацією! Створіть код, який завжди буде будувати на землі попереду Стіва факел. Якщо так трапиться, що Стів виявиться в повітрі – побудуйте замість факелу блок лампи. Ні темряві!