****

**Lekcja IV**

**Moby i przedmioty**

# **Cel lekcji**

Celem zajęć będzie poznanie mechanizmów odpowiadających za pojawianie się różnego rodzaju mobów oraz przedmiotów w grze z wykorzystaniem pętli.

**Agenda**

1. Pytania wstępne i rozmowa
2. Pętla for
3. Steve Archer
4. Zombie run
5. Pajęcze siedlisko

# **Pytania wstępne**

1. Co to są funkcje?
2. Jak utworzyć własną funkcję?
3. Jak budować z użyciem czatu?

**IDLE**

**Pętla for:**

Kolejny element programowania jaki poznamy to pętle. Pętla jest to taki element języka, który pozwala powtarzać kilkukrotnie pewne fragmenty skryptów. Na dzisiejszych zajęciach poznamy jedną z pętli, które występują w języku Python – ***for***. Wyobraźmy sobie, że ktoś poprosił nas o napisanie skryptu, który wyświetli liczby od 0 do 100. Teoretycznie moglibyśmy napisać sto jeden razy funkcje ***print()***, ale nie jest to raczej wygodny sposób. Możemy do tego wykorzystać wspomnianą pętlę ***for***:

for i in range(101):

    print(i)

Opowiadamy, że ***range()*** jest to funkcja która zwraca kolejne liczby z podanego zakresu, domyślnie zaczynając od zera i w każdym wykonaniu pętli do zmiennej pętli **i** przypisywana jest ta liczba. Liczba podawana jako górna część zakresu nie jest do niego uwzględniona.

Do funkcji ***range()*** możemy przekazać dwa parametry (oddzielone przecinkiem), wtedy określamy zakres liczb od do:

for i in range(5, 10):

    print(i) # wynik: 5, 6, 7, 8, 9

Zwrócimy uwagę, że podany zakres 5 – 10, wylistuje liczby od 5 do 9.

Istnieje też wersja funkcji ***range()*** z trzema parametrami, wtedy trzeci parametr określa krok (czyli o ile ma być zwiększana wartość zmiennej w każdej iteracji). Na przykład wyświetlenie tylko liczb nieparzystych:

for i in range(1, 10, 2):

    print(i)

# wynik: 1, 3, 5, 7, 9

Zobaczmy zastosowanie pętli w bardziej rozbudowanych skrypcie. Napiszemy skrypt, który będzie w stanie wyświetlić na ekranie sumę wszystkich liczb podanych przez użytkownika. Na początku program ma pytać, ile liczb chcemy zsumować, a następnie prosi kolejno o ich wpisanie. Na końcu wyświetla ich sumę.

ilosc = int(input("Podaj ile liczb chcesz zsumować? "))

# tworzymy zmienną na sumę

suma = 0

for i in range(1, ilosc + 1):

    liczba = int(input("Podaj liczbę " + str(i) + ": ")) # pobieramy kolejną liczbę do dodania

    suma = suma + liczba

    # do zmiennej suma za każdym razem dodajemy to co w tej zmiennej już jest plus nowa podana liczba

    # po zakończeniu pętli wyświetlamy wynik

print("Suma podanych liczb: " + str(suma))

**MINECRAFT**

**STEVE ARCHER**

Tworzymy mini-grę, w której będziemy przy pomocy pętli for spawnować potwory - creepery.

Po utworzeniu potworów i krótkiej przerwie czasowej nakładamy efekt na potwory, żeby zaczęły się unosić - wtedy też dostajemy łuk i strzały.

Podczas tworzenia mobów musimy uciekać - jeśli creeper podejdzie zbyt blisko gracza to wybuchnie, zadając duże obrażenia.

Bieg: 2x klawisz W

Strzał: naciąganie cięciwy i strzał PPM

<https://minecraft-pl.gamepedia.com/Creeper>



def lewitacja():

    #dostajemy łuk

    mobs.give(mobs.target(NEAREST\_PLAYER), BOW, 1)

    #dostajemy strzały

    mobs.give(mobs.target(NEAREST\_PLAYER), ARROW, 64)

    #dodajemy efekt lewitacji do wszystkich creeperów

    mobs.apply\_effect(LEVITATION, mobs.entities\_by\_type(CREEPER),20,1)

def CreeperSpawn():

    #tryb przetrwania

    gameplay.set\_game\_mode(SURVIVAL, mobs.target(NEAREST\_PLAYER))

    #używamy pętli aby utworzyć potwory

    for i in range(10):

        mobs.spawn(CREEPER, randpos(pos(-30, 0, -30), pos(30, 0, 30)))

        loops.pause(4000)

    #wywołujemy funkcję lewitacja

    lewitacja()

player.on\_chat("creeper",CreeperSpawn)

#nowe zdarzenie, podczas śmierci gracza przełączamy tryb rozgrywki na kreatywny

def on\_died():

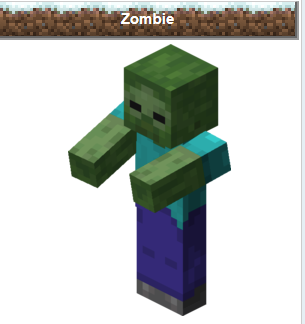
    gameplay.set\_game\_mode(CREATIVE, mobs.target(NEAREST\_PLAYER))

player.on\_died(on\_died)

**ZOMBIE RUN**

Tworzymy potwory w dużej ilości wokół gracza. Naszym zadaniem jest ucieczka aż do momentu, w którym wstanie dzień - wtedy zombie spłoną. Pozostałe potwory dobijamy otrzymanym toporem.

<https://minecraft-pl.gamepedia.com/Zombie>



def ZombieSpawn():

    #tryb przetrwania

    gameplay.set\_game\_mode(SURVIVAL, mobs.target(NEAREST\_PLAYER))

    mobs.give(mobs.target(NEAREST\_PLAYER), DIAMOND\_AXE, 1)

    gameplay.time\_set(DayTime.NIGHT)

    #używamy pętli, aby utworzyć potwory

    for i in range(100):

        mobs.spawn(ZOMBIE, randpos(pos(-30, 0, -30), pos(30, 0, 30)))

    #zmieniamy cykl dnia

    for i in range(100):

        gameplay.time\_add(100)

    gameplay.set\_weather(CLEAR)

player.on\_chat("zombie",ZombieSpawn)



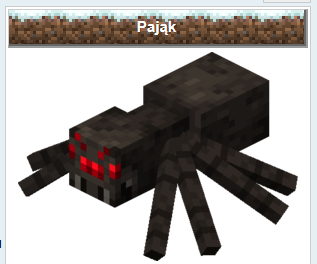
**Opcjonalnie:**

**PAJĘCZE SIEDLISKO**

Mini gra polega na utworzeniu jaskini w której gracz zostaje zamknięty, w której rozmieścimy bloki pajęczyny. Gracz wchodząc na taki blok wytraca prędkość. Dodatkowo spawnujemy pająki, które trzeba będzie pokonać. Dojście do jasnogłazu i zniszczenie go teleportuje gracza w bezpieczne miejsce.

[**https://minecraft-pl.gamepedia.com/Paj%C4%99czyna**](https://minecraft-pl.gamepedia.com/Paj%C4%99czyna)

[**https://minecraft-pl.gamepedia.com/Paj%C4%85k**](https://minecraft-pl.gamepedia.com/Paj%C4%85k)





pozycja=player.position()

x=pozycja.get\_value(Axis.X)

y=pozycja.get\_value(Axis.Y)

z=pozycja.get\_value(Axis.Z)

def jaskinia():

    blocks.fill(STONE, world(x-8,y-1,z-2),world(x+8,y+10,z+50))

    blocks.fill(AIR, world(x-7,y,z-1),world(x+7,y+10,z+49))

    spiderSpawn()

player.on\_chat("pająki", jaskinia)

def spiderSpawn():

    #tryb przetrwania

    gameplay.set\_game\_mode(SURVIVAL, mobs.target(NEAREST\_PLAYER))

    mobs.give(mobs.target(NEAREST\_PLAYER), DIAMOND\_AXE, 1)

    mobs.give(mobs.target(NEAREST\_PLAYER), BOW, 1)

    mobs.give(mobs.target(NEAREST\_PLAYER), ARROW, 64)

    gameplay.time\_set(DayTime.NIGHT)

    #używamy pętli aby utworzyć pajęczyny

    for i in range(100):

        blocks.place(COBWEB, randpos(world(x-7,y,z), world(x+7,y+4,z+45)))

    for i in range(6):

        mobs.spawn(SPIDER, randpos(world(x-7,y+1,z+10), world(x+7,y+3,z+45)))

    blocks.place(GLOWSTONE, world(x,y,z+49))

 #zdarzenie na zniszczenie konkretnego bloku

def on\_block\_broken():

    gameplay.set\_game\_mode(CREATIVE, mobs.target(NEAREST\_PLAYER))

    player.teleport(pos(50, 0, 50))

blocks.on\_block\_broken(GLOWSTONE, on\_block\_broken)

**Podsumowanie**

1. Co to jest pętla i jak działa?
2. Jakie poznaliśmy nowe zdarzenia?
3. Jak działa funkcja range()?
4. Jaka jest różnica pomiędzy pozycją względną gracza a pozycją świata gracza.?