

# 3D tisk domku na Raspberry Pi

Popis pro lektora



3D tisk získává na popularitě. Dnes není problém si vytisknout drobný model draka, přívěšek na klíče v podobě kostlivce nebo otvůrák na lahve, který umí vystřelit samotnou zátku. Přestavte si ale, že byste za pomoci 3D tiskárny vytiskli celý dvoupatrový dům. Aby ale 3D tiskárna tiskla to, co má, potřebuje k tomu robotické rameno, které bude poslouchat příkazy, co jak tisknout.

S Minecraftem si tento zdánlivě náročný úkol snadno vyzkoušíte. Od jednoduchých sloupků přes zídky až k vytisknutí celého domku se střechou. K tomu si zahrajete si na robota, který tiskne obrázky a zjistíte, jak musí programátor umět rozdělit velký problém na drobné úkoly, aby vytiskl správný obrázek.

## Obsah

Cíle .....	3
Doplňkové materiály.....	3
Požadavky .....	4
Příprava .....	4
Příprava prostředí Scratch.....	6
Příprava IDE Thonny pro programování v Pythonu.....	8
Postup řešení.....	8
Píšeme do chatu.....	8
První stavby.....	9
Stavba domu .....	12
Doplňková aktivita .....	15
Příprava.....	15
Pravidla .....	16
Ovládání hry.....	16
Klíčové výrazy.....	16
Seznam použitých knihoven.....	17

„Minecraft“ je ochrannou známkou společnosti Mojang AB. Tato webová stránka ani její součásti nejsou spojeny se společností Mojang AB.

Tento projekt vznikl jako ročníkový projekt v rámci Fakulty mechatroniky, informatiky a mezioborových studií na Technické univerzitě v Liberci

## Cíle

Cílem kurzu „3D tisk turtlem“ je seznámit děti s možnostmi 3D tisku, výukovými mikropočítači Rapsberry Pi a způsobem rozkládání komplexního problému na drobnější úkony. Kurz propojuje především různé informace a obory informačních technologií.

Seznamuje žáky s programovacím jazykem Python za využití několika málo knihoven, díky kterým naprogramují chování robota (turtla) ve hře Minecraft na mikropočítači Raspberry Pi.

Pokud jste pracovali s předchozím kurzem „Robotizace třídící linky“, je v tomto kurzu připraven zároveň podobný program na stavbu domu s komentáři ve vizuálním programovacím jazyce Scratch. Primárně se tento kurz zaměřuje na programování v jazyce Python.

## Doplňkové materiály

- 3D tiskárny
  - České 3D tiskárny Josefa Průši: <https://www.prusa3d.cz/>
- Stavba domů pomocí 3D tiskáren
  - 3D tišený hausbót v Praze – iDnes: [https://www.idnes.cz/ekonomika/domaci/3d-tisk-dum-stavba-beton.A200818\\_170423\\_ekonomika\\_mato](https://www.idnes.cz/ekonomika/domaci/3d-tisk-dum-stavba-beton.A200818_170423_ekonomika_mato)
  - Největší 3D tišený dům v Evropě – Designboom: <https://www.designboom.com/architecture/kamp-c-3d-prints-two-story-house-08-17-2020/>
- 3D tisk v potravinářství
  - 3D tisk kuřecího masa – Aktuálně.cz: <https://video.aktualne.cz/dvty/kfc-chysta-3d-tisk-kureciho-bude-to-produkt-farmaceutickeho/r~42b50444d18411eaa7deac1f6b220ee8/>
- Modely pro 3D tiskárny
  - MakerBot Thingiverse: <https://www.thingiverse.com/>
  - Cults3d: <https://cults3d.com/en>
- Mikropočítače
  - Raspberry Pi: <https://www.raspberrypi.org/>
  - BBC micro:bit: <https://microbit.org/>
  - Banana Pi: <http://www.banana-pi.org/>
  - Arduino: <https://www.arduino.cc/>
- Vizuální programovací jazyky
  - Scratch: <https://scratch.mit.edu/>
  - Microsoft MakeCode: <https://makecode.microbit.org/>
  - Code.org: <https://code.org/>

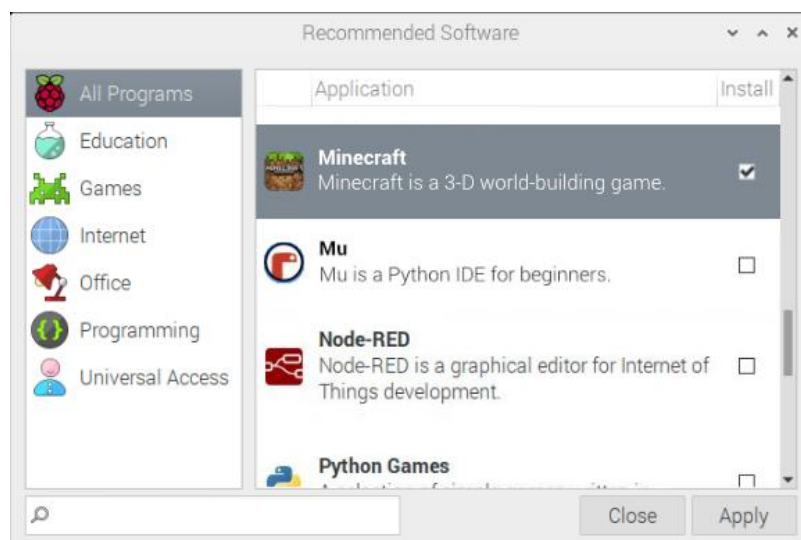
## Požadavky

- Mikropočítač **Raspberry Pi** s **Raspberry Pi OS** (dříve Raspbian) včetně periférií (monitor, klávesnice, myš) s funkčním internetovým připojením
  - Testováno na Raspberry Pi 3 Model B+
- Balíčky **minecraft-pi**, **python**
- Programovací prostředí (IDE) **Thonny** (předinstalované v distribuci Raspberry Pi OS)
- V případě využití kódu Scratch, externí knihovna **Scratch2MCPI** (instalace viz níže)
  - Soubor stavbaDomu.sb

## Příprava

Po instalaci operačního systému Raspberry Pi OS na microSD kartu je nutné nainstalovat dodatečné balíky.

Instalaci balíku Minecraft můžete provést v aplikaci **Recommended Software**, který najdete v Menu > Volby > Recommended Software. Potvrzením tlačítkem OK nainstalujete potřebné balíky:



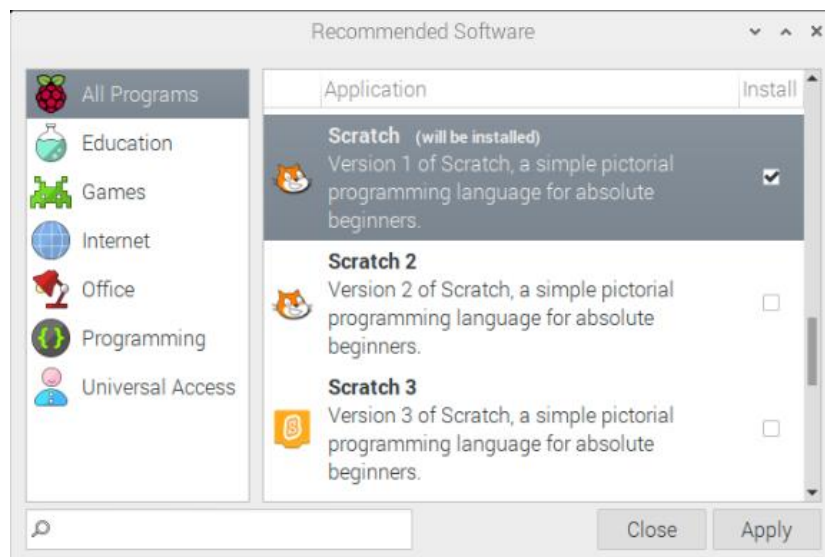
Pokud není aplikace Recommended Software k dispozici, spusťte v terminálu následující příkazy (vyžadují root oprávnění = jako normální uživatel je spouštějte v sudo režimu. Např. `apt update` spusťte jako `sudo apt update` apod.):

```
apt update
apt install python minecraft-pi
```

Dále do domovského adresáře stáhněte balík s ukázkovými kódy a knihovnami pro jednodušší práci s turtlem ve hře:

```
cd ~
git clone https://github.com/martinohanlon/minecraft-turtle.git
cd ~/minecraft-turtle
python setup.py install
```

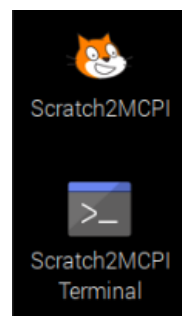
Chcete-li využít i kódu ve vizuálním programovacím jazyce Scratch, nainstalujte pomocí nástroje Recommended Software balík Scratch (verze 1):



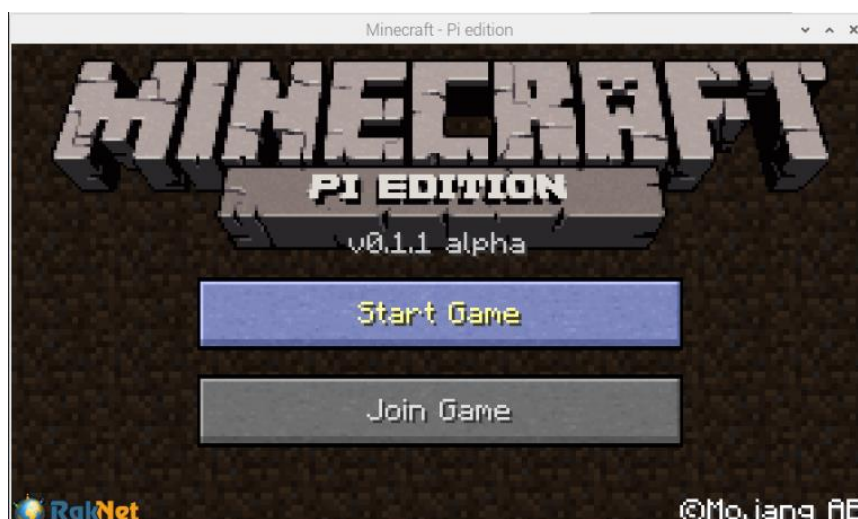
Pokračujte instalací doplňkové překladové knihovny v terminálu příkazem:

```
curl http://scratch2mcp.i.github.io/install.sh | sh
```

Po instalaci získáte na ploše dvě ikony:

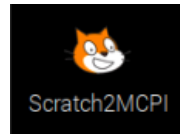


Spusťte **Minecraft: Pi Edition**, klikněte na tlačítko **Start Game** a tlačítko **Create New Game**. Tím vytvoříte novou mapu a připojíte se na ni:

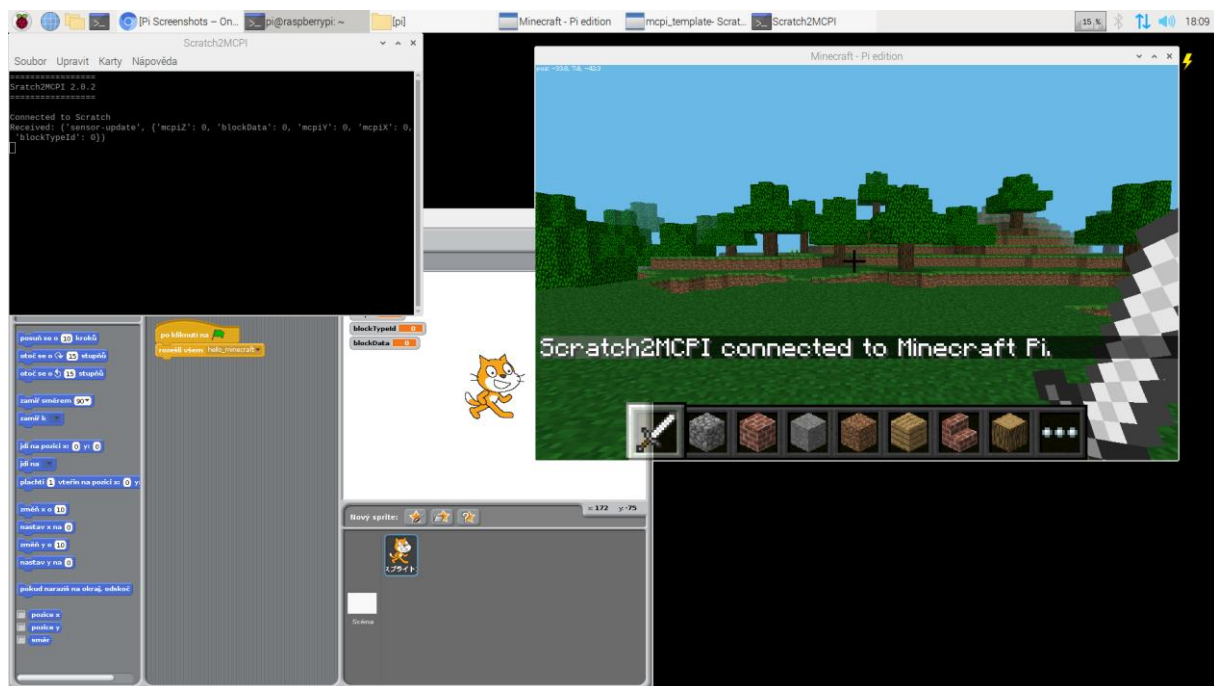


## Příprava prostředí Scratch

Pro spuštění Scratche a jeho automatické napojení na Minecraft je nutné spustit nově nainstalovanou ikonu Scratch2MCPI:



Spustí se prostředí Scratch a úspěšné připojení do hry ohlásí zpráva v chatu hry „Scratch2MCPI connected to Minecraft Pi.“. Prostředí vypadá momentálně podobně:



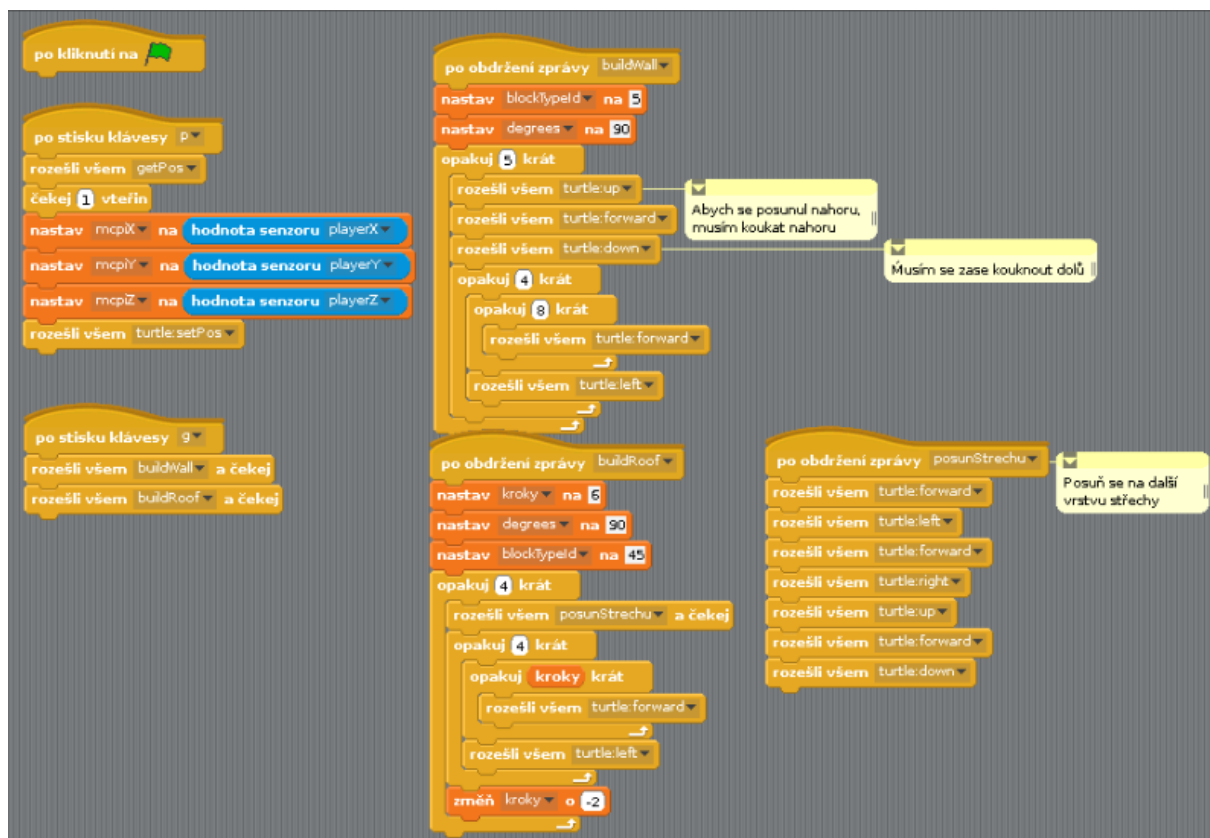
V prostředí Scratch klikněte na možnost Soubor > Otevřít a najděte přiložený soubor **stavbaDomu.sb**. Připravený soubor importuje blokový kód, při jehož spuštění lze postavit domek ve hře.

Uvedené funkce je nutné spouštět v prostředí Scratch. Stisknutím tlačítka **p** zjistí Scratch ze hry souřadnice hráče a uloží si je do přednastavených proměnných **playerX**, **playerY** a **playerZ**. Souřadnice, které jsou ve hře zobrazené v levém horním rohu se tak nyní zobrazily i v bílém poli okna Scratch, u položky „hodnota senzoru **playerX**“.

Stisknutím tlačítka **g**, se zavolají připravené bloky funkcí **buildWall** a následně **buildRoof**. V těchto případech je možné sledovat ve hře turtla (ve hře reprezentován diamantovým blokem), který vystaví stěny z bloků dřevěných planěk a střechu z bloků cihel.

Mezi hrou a prostředím Scratch je možné se přepínat klávesovou zkratkou **Alt + Tab**. Hra tak nezobrazí tzv. Pause menu.





Výsledná stavba poté vypadá podobně:



Poznámka: Při stavění je vhodné najít rovnou plochu bez stromů. Turtle umí tyto bloky přestavět, ale volba místa je důležitá, kvůli následnému dojmu.

## Příprava IDE Thonny pro programování v Pythonu

K programování v jazyce Python využijeme integrovaného vývojářského prostředí (IDE) Thonny, které v Raspberry Pi OS předinstalované. V menu spusťte aplikaci Thonny IDE a první soubor který vytvoříte, uložte a pojmenujte. Například **uvod.py**.

Soubory v jazyce Python, které nebudou uloženy, nebude možné spustit.

Aby se kód spouštěl, je nutné mít spuštěný Minecraft a musíte být připojeni na mapě, viz výše.

## Postup řešení

### Píšeme do chatu

Začněme prvními nutnými řádky:

```
from mcpi import minecraft
```

Díky tomuto importu zařídíme, že náš kód bude mít k dispozici příkazy, které komunikují se spuštěnou hrou. Pokračujme vytvořením proměnné s inicializací připojení:

```
mc = minecraft.Minecraft.create()
```

Nyní nám už nic nebrání napsat něco do chatu:

```
mc.postToChat("Ahoj svete")  
mc.postToChat(2+2)
```

První příkaz nám vypíše do chatu hlášku „Ahoj svete“. České znaky nejsou hrou podporované, proto je vhodné psát bez diakritiky. Druhý příkaz vypíše výsledek „4“.

Podobně můžeme využít i cyklus for:

```
for číslo in range(5):  
    mc.postToChat(cislo)
```

Do hry se vypíší postupně čísla od 0 do 4.

Zkusme si zjistit souřadnice hráče. Využijme objektu player a jeho funkce **getPos()**:

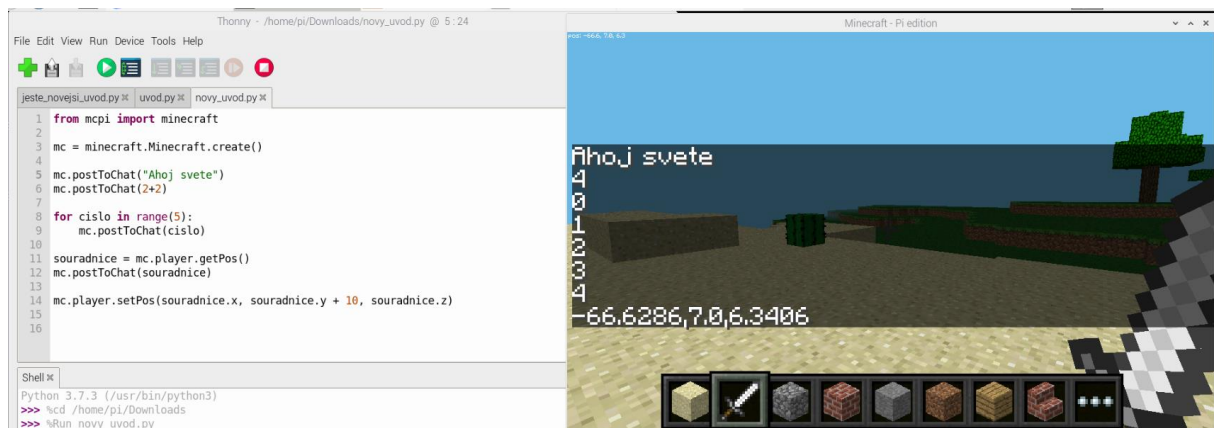
```
souradnice = mc.player.getPos()  
mc.postToChat(souradnice)
```

Do chatu se vypíší souřadnice X, Y a Z. Využijme souřadnic hráče a teleportujme ho o 10 bloků nad jeho aktuální pozici. Rozšířme aktuální kód o řádek:

```
mc.player.setPos(souradnice.x, souradnice.y + 10, souradnice.z)
```

Pokud vše fungovalo, jak mělo, hráč by nám měl poskočit o deset bloků výše.





Výsledný kód prvního souboru tak vypadá následovně:

```
from mcpi import minecraft

mc = minecraft.Minecraft.create()

mc.postToChat("Ahoj světe")
mc.postToChat(2+2)

for cislo in range(5):
    mc.postToChat(cislo)

souradnice = mc.player.getPos()
mc.postToChat(souradnice)

mc.player.setPos(souradnice.x, souradnice.y + 10, souradnice.z)
```

Tento úvodní kód je taktéž k dispozici v souboru uvod.py, který najdete ke stažení na webové stránce.

## První stavby

Nyní je vhodné, aby další příkazy byly oddělené ve vlastním souboru. Vytvořte nový soubor, uložte a pojmenujte, aby bylo možné program spouštět.

Abychom mohli využít turtla, je zapotřebí importovat více knihoven. Z nainstalované knihovny mcturtle importujeme funkce pro operaci turtla, který ve hře bude reprezentován diamantovým blokem. Další dva řádky importují samotné připojení k Minecraftu a databázi bloků, které lze ve hře využít:

```
from mcturtle import minecraftturtle
from mcpi import minecraft
from mcpi import block
```

Umístíme turtla na pozici hráče:

```
mc = minecraft.Minecraft.create()  
pos = mc.player.getPos()  
turtle = minecraftturtle.MinecraftTurtle(mc, pos)
```

Pokud se všechno povedlo, jak mělo, ve hře vedle sebe vidíme nově postavený diamantový blok.

Zkusme pokračovat a postavit rovnou řádku bloků o pěti blocích:

```
turtle.forward(5)
```

Turtle postaví 5 kamenných bloků za sebou. Abychom mohli odbočit, je nutné zadat příkaz na otočení o 90°. K tomu slouží příkazy `turtle.left(úhel)` respektive `turtle.right(úhel)`. Postavme písmeno L:

```
turtle.forward(3)  
turtle.left(90)  
turtle.forward(2)
```

Příkaz `turtle.left(90)` otočí turtla o 90° doleva, nepostaví však žádný blok. K tomu je nutné zadat příkaz pro posun vpřed.

Turtle momentálně staví relativně pomalu. Větší stavby by tak mohly trvat příliš dlouho. Zrychleme stavbu příkazem `turtle.speed(10)` Hodnota 1 – nejpomalejší, 10 – nejrychlejší, hodnota 0 vypne animaci úplně (postaví vše naráz).

Zkombinujme výše naučený cyklus **for** a postavme čtverec, který nám může pomalu začít tvořit základy domku:

```
for i in range(4):  
    turtle.forward(5)  
    turtle.left(90)
```

Čtyřikrát se zopakuje stavba stěn, pokaždé se turtle pootočí o 90° doleva. Vznikne tak čtverec o velikosti 5x5.

Aby turtle se mohl posouvat a přitom „nekreslit“ (nepokládat žádné bloky), bude potřeba oznámit turtlovi, aby tzv. zvedl své pero, kterým kreslí. K tomu využijeme příkazy `turtle.penup()` a `turtle.pendown()`, kterým kreslit opět začne. Využijme tohoto pro nakreslení přerušované čáry:

```
for i in range(5):  
    turtle.forward(2)  
    turtle.penup()  
    turtle.forward(2)  
    turtle.pendown()
```

Dosud jsme kreslili jedním druhem bloku. Dům je však postavený z různých materiálů. Jednotlivé bloky mají svá ID (identifikační čísla). Je možné tak turtlovi příkazem `turtle.penblock(ID)` přikázat, aby kreslil blokem s ID 1, nebo postupně při stavbě bloky měnil:

```
blok = 1
for i in range(10):
    turtle.forward(1)
    turtle.penblock(blok)
    blok += 1
```

Bude postavena řádka s deseti různými materiály. Od kamene, přes trávník a kámen, až po vodu, která se na místě rozteče.



Vzhledem k tomu, že jsme výše importovali knihovnu se seznamem bloků, není nutné znát jednotlivá ID bloků, postačí, když známe název daného materiálu:

```
turtle.penblock(block.DIRT.id)
```

V tomto případě jsme získali `block.DIRT` (hlínu) a vzhledem k tomu, že turtle vyžaduje jeho ID, zjistíme si tuto hodnotu pomocí `.id`

Tabulka nejdůležitějších bloků pro stavbu domku:

Název	Česky	ID	Název	Česky	ID
STONE	Kámen	1	GLASS_PANE	Skleněná deska	102
GRASS	Trávník	2	GLASS	Sklo	20
DIRT	Hlína	3	SANDSTONE	Pískovec	24
COBBLESTONE	Kamení	4	IRON_BLOCK	Blok železa	42
WOOD_PLANKS	Dřevěné plaňky	5	BRICK_BLOCK	Cihly	45
WOOD	Kmen dřeva	17	STONE_BRICK	Kamenné cihly	98

Celý seznam bloků a jejich ID najdete zde, v části `.id`:

<https://www.stuffaboutcode.com/p/minecraft-api-reference.html>

Aby bylo možné pokračovat ve stavbě domku, je zapotřebí stavět do výšky. Je potřeba, aby se turtle tzv. díval nahoru. Podobně, jako u pohledů do stran, lze turtla nasměrovat nahoru nebo dolů příkazy `turtle.up(úhel)` a `turtle.down(úhel)`. Postavme sloupek nahoru, vysoký 9 bloků:

```
turtle.up(90)
turtle.forward(9)
```

Pokud pokračujeme v psaní stejného souboru, chceme nyní postavit opět čtverec, který bude vodorovně položen. K tomu, abychom ale nasměrovali turtla správným směrem, je zapotřebí nejprve použít příkaz `turtle.down(úhel)`, aby nedošlo ke stavě vertikální:

```
turtle.down(90)
for i in range(4):
    turtle.forward(5)
    turtle.left(90)
```

Nutnost použití příkazu `turtle.down(90)` lze žákům zatajit, aby sami zkusili přijít na to, proč staví turtle svisle a ne vodorovně.

Jestliže se vše povedlo, jak mělo, může výsledná stavba vypadat podobně jako na následujícím obrázku:



Soubor `uvodni_kresleni.py` se všemi výše uvedenými příkazy, je dostupný ke stažení.

## Stavba domu

Nyní víme, jak pracovat s příkazy, zvedat a pokládat pero, měnit materiály, stavět jednoduché ornamenty a pohybovat se nahoru a dolů. Nic nebrání stavbě domu.

Stejně jako v předchozím případě, je vhodné vytvořit nový pracovní soubor a uložit ho, aby bylo možné skript spouštět. Na úvod je opět nutné importovat knihovny:

```
from mcturtle import minecraftturtle
from mcpi import minecraft
from mcpi import block
```

A umístit turtla na souřadnice hráče. Je nutné vytvořit připojení, získat pozici hráče a na ně umístit turtla. Pokud se vše podaří, jak má, zobrazí se na pozici hráče opět diamantový blok reprezentující turtla:

```
mc = minecraft.Minecraft.create()
pos = mc.player.getPos()
turtle = minecraftturtle.MinecraftTurtle(mc, pos)
```

Abychom si zkrátili čekání, zrychleme stavbu příkazem `turtle.speed(10)`

Stěny domu stavme z dřevěných planěk. Dle tabulky můžeme zjistit, že se jedná o ID 5, nebo můžeme použít `block.WOOD_PLANKS.id`:

```
turtle.penblock(block.WOOD_PLANKS.id)
```

Postavme čtvercové stěny dlouhé 8 bloků, do výšky 4 bloků. K tomu využijeme dva cykly **for** vnořené do sebe:

```
for i in range(5):
    for j in range(4):
        turtle.forward(8)
        turtle.right(90)
    vyskaTurtla = turtle.position.y
    turtle.sety(vyskaTurtla + 1)
```

Každý cyklus postavíme čtverec pro stěny, zjistíme souřadnici Y turtla a teleportujeme ho o řádek výše, nastavením souřadnice Y o 1 vyšší.

Robot nám již staví stěny. Dům potřebuje pyramidovou střechu. K té využijeme složitější logiky. Stále budeme stavět čtverce, ty ale musí být pokaždé o dva bloky kratší. K tomu využijeme **while** cyklus.

Začněme nastavením původní délky střechy podle délky stěn. Nastavíme proměnnou:

```
delkaStrechy = 8
```

Aby turtle stavěl střechu z jiných bloků, zadejme jiný druh bloku. Využijme cihel:

```
turtle.penblock(block.BRICK_BLOCK.id)
```

Nyní již k samotnému cyklu. Dokud budeme moci stavět střechu, stavme. Aneb dokud bude délka jedné stěny vrstvy střechy větší nebo rovna nule (protože poslední posun nahoru postaví špičku), opakujme cyklus stavění stran tak dlouhých, jaká je hodnota **delkaStrechy**:

```
while delkaStrechy >= 0:
    for i in range(4):
        turtle.forward(delkaStrechy)
        turtle.right(90)
```

Pokračujeme kódem uvnitř cyklu **while**. Nyní je potřeba stavět menší čtverec o vrstvu výše a ve vnitřní straně. To můžeme provést dvěma způsoby. Buď zvedneme pero a posuneme turtla o jeden blok výše a adekvátně o jeden blok dovnitř po souřadnicích X a Y.

Druhou, jednodušší, možností je zjistit si aktuální polohu turtla a bez zvedání a opětovného pokládání pera, můžeme okamžitě turtla teleportovat na správné souřadnice:

```
pozice = turtle.position
turtle.setposition(pozice.x + 1, pozice.y + 1, pozice.z + 1)
```

K tomu, abychom docílili efektu pyramidy, je potřeba dělat obdélníky každou vrstvu o dva bloky kratší:

```
delkaStrechy -= 2
```

Výsledný cyklus **while**, by tedy mohl vypadat následovně:

```
delkaStrechy = 8
turtle.penblock(block.BRICK_BLOCK.id)
while delkaStrechy >= 0:
    for i in range(4):
        turtle.forward(delkaStrechy)
        turtle.right(90)
    pozice = turtle.position
    turtle.setposition(pozice.x + 1, pozice.y + 1, pozice.z + 1)
    delkaStrechy -= 2
```

Následovně může vypadat turtle při stavbě domku podle kódu:



Výsledná stavba by měla vypadat totožně, jako v případě programu ve Scratchi, viz výše.

Stejně jako v předchozích případech, je i tento výsledný kód s dodatkovými komentáři kódu, přiložený ke stažení z webových stránek, jako soubor **domek.py**.



Ve zbylém čase je možné domek dozdobit. Přistavět dveře, okna, zahrádku, případně vybavit domek interiérem. Výsledná stavba tak může vypadat například takto:



## Doplňková aktivita

K následujícímu programování lze zařadit také doplňkovou aktivitu zaměřenou na pochopení komunikace s omezeným počtem konstrukcí mezi programátorem a samotným turtlem. Cílem aktivity je na připravené pole nakreslit předem určené obrázky, určenými barvami, aniž by se oba účastníci (nebo skupiny účastníků), navzájem viděli.

### Příprava

K této aktivitě připravte na zem malířskou lepicí páskou pole o velikosti 5×5, označte, která strana bude vlevo, vpravo, nahoře a dole. Skupině, která bude turtle, dvě sady barevných lepicích štítků (oranžové a žluté). Vytiskněte přiložený návod, který je k dispozici ke stažení na webové stránce.

## Pravidla

- Osoba, která reprezentuje turtla začíná vždy v levém horním poli.
- Navigující osoba – programátor, smí použít pouze pokyny z dostupné tabulky.
- Osoba reprezentující turtla nesmí opustit pole. Pokud je zadáný pokyn, kdy by opustil pole, zůstává turtle na místě.

Příkaz	Popis
Jdi vpřed o <počet>!	Turtle jde vpřed o počet políček, přitom pokládá barevné lístečky
Otoč se doleva!	Turtle se otočí doleva, od směru, kam kouká. Vpřed nejde!
Otoč se doprava!	Turtle se otočí doprava, od směru, kam kouká. Vpřed nejde!
Budeš kreslit!	Turtle při pohybu vpřed bude pokládat barevné lístečky
Nebudeš kreslit!	Turtle při pohybu vpřed nebude pokládat barevné lístečky
Kresli <barvou>!	Turtle bude kreslit vybranou barvou při pohybu vpřed

Navigaci je možné zjednodušit přidáním několika otázek typu „getter“:

Příkaz	Popis
Kam koukáš?	Turtle odpoví směr, kterým se dívá
Kterou barvou kreslíš?	Turtle odpoví barvu, kterou právě drží

Pro případ, že se turtle programátorovi ztratí na kreslicí ploše, je dobrým nápadem dostat turtla do nejjistějšího okraje plochy, odkud dále může pokračovat v pokynech kreslení a pohybu.

## Ovládání hry

<b>W</b>	Chůze vpřed	<b>2× mezerník</b>	Vznesení se do letu / přistání
<b>A</b>	Chůze doleva	<b>Mezerník za letu</b>	Stoupání
<b>S</b>	Chůze vzad	<b>Shift za letu</b>	Klesání
<b>D</b>	Chůze doprava	<b>E</b>	Inventář
<b>Mezerník</b>	Výskok	<b>Levý klik</b>	Interakce / zničení bloku
<b>Esc</b>	Pauza	<b>Pravý klik</b>	Interakce / položení blok
<b>1-9</b>	Výběr na liště bloků		

## Klíčové výrazy

**Inventář** – Seznam dostupných bloků a předmětů, které si lze přidat na hlavní nástrojovou lištu a stavět s nimi ve světě.

**Turtle** – Kódem ovládaný robot vykonávající příkazy k stavbě a pohybu, ve hře reprezentovaný diamantovým blokem.

**Scratch** – Vizuální programovací jazyk vyvinutý především k výukovým účelům univerzitou MIT.

## Seznam použitých knihoven

- O'HANLON, Martin, 2014. Minecraft Graphics Turtle. <Stuff about="code" />: about = (stuff == 'code') [online]. [cit. 2020-08-09]. Dostupné z: <https://www.stuffaboutcode.com/2014/05/minecraft-graphics-turtle.html>
- ISHIHARA, Junya. Scratch2MCPI(Scratch2MinecraftPi). In: Scratch2MCPI(Scratch2MinecraftPi) [online]. [cit. 2020-08-09]. Dostupné z: <https://github.com/scratch2mcpi/scratch2mcpi>