

www.moravia.education

www.vyuka-vzdelavani.cz

Obsah

Začínáme	. 3
1. Barevné světlo	. 5
2. Přehrávání zvuků	. 6
3. Reakce na úroveň zvuku	. 7
4. Pohyb motoru	10
5. Opakovaný pohyb	11
6. Držte úhly motoru	12
7. Postav si pick & place robota	13
8. Postavte třídícího robota	16
9. Postavte robota, který projede bludiště	19
10. Sestavte robota, který kreslí pomocí laseru	25
11. Házející robot	26
12. Vytvořte si bezpečnostního robota	28
13. Postavte společenského robota	33
15. Kopírování pohybu	47
16. Fable ukazuje sever	52

Začínáme

Otevřete Fable Blockly na vašem počítači nebo tabletu.

Zapojte dongle do USB portu nebo do sítě. Barevně se rozsvítí.

Zapněte Fable Joint zmáčknutím vypínače. Také se barevně rozsvítí – ujistěte se, že barva je shodná s barvou dongle.



Ve Fable Blockly otevřete sekci **Actions** a vyberte blok **move to**:



Změňte úhly X a Y motorů a ujistěte se, že byl připojen správný Joint modul (všechny moduly mají unikátní ID, které najdeme na zadní části modulu).



Stiskněte tlačítko **Play** pro otestování programu:

Přidejte další pohybový blok, ale nejdříve využijte **wait in sec** blok v sekci **Loops**:



Přidejte **wait in sec** blok pod **move to** blok a další **move to** blok přidejte pod **wait in sec** blok. V tomto bloku neměňte úhly, nebo je nastavte na jinou hodnotu než 60.

The move to X: angle 60° Y: angle 0° on 0XA
Wait in sec.
The move to X: angle 0° Y: angle 0° on 0XA -

1. Barevné světlo

Tento příklad ukazuje, jak ovládat barevné LED světlo na Fable donglu a modulech.

Vybavení:

- Dongle (Hub)
- Fable Joint
- Počítač nebo tablet s Fable Blockly

Popis:

Program nejprve nastaví světlo na modulu na žlutou a dongle na modrou. Potom počká 3 sekundy a poté změní barvy na zelenou a fialovou. O 3 sekundy později se program ukončí a barvy se vrátí do původního nastavení.



2. Přehrávání zvuků

Tento příklad ukazuje, jak nastavit rozhraní Fable k přehrávání standardních zvuků.

Vybavení:

- Dongle (Hub)
- Počítač s Fable Blockly

Popis:

Software Fable nabízí řadu standardních zvuků, které lze vybrat z nabídky v bloku **Play Sound**.

V programu níže se na reproduktorech počítače přehraje bzučivý zvuk. O 1,5 sekundy později se spustí zvuk smíchu, po kterém se program ukončí.



Přidání zvukových složek

Do programu Fable můžete také přidat zvukové soubory z počítače.

Zvukový soubor můžete buď stáhnout z internetu (musí být ve formátu .wav nebo .mp3) a uložit do složky: **Documents/My Fable Sounds**, nebo si můžete nahrát svůj vlastní zvuk a uložit ho do stejné složky.

V advanced mode pod sekcí Actions ve Fable Blockly programu najdete blok s názvem play sound file.



Nyní si můžete vytvořit vlastní symfonii! Inspiraci najdete v tomto videu: <u>https://youtu.be/xGCZxZebv58</u>



3. Reakce na úroveň zvuku

Tento příklad ukazuje, jak nastavit změnu barvy Fable dongle v závislosti na naměřené hladině zvuku. Hlasitý zvuk způsobí, že dingle zčervená a při nízké úrovni zvuku zezelená.

Vybavení

- Dongle (Hub)
- Počítač s Fable Blockly

Popis

Program běží v nekonečné smyčce, pokud je měřená hladina zvuku nad 10 (nejlepší hodnota se bude lišit dle prostředí). Pokud je hladina zvuku vyšší než 10, rozsvítí se dongle červeně a jinak bude svítit zeleně. Současně program také vytvoří graf naměřené hladiny zvuku.

Nejdříve pod sekcí Loops vyberte blok repeat forever.



Potom v sekci Logic vyberte if do else blok a přetáhněte ho do repeat forever.

repe	eat forever	
do	if i do	true
	eise	

Znovu běžte do sekce **Logic** a vyberte blok **0 = 0** ve složce **Comparison**.



Nyní v sekci **Senses** vyberte blok nazvaný **sound level** a přetáhněte ho do bloku, kde je první **O**.





Nový blok přetáhněte a přidejte ho do **if do else**, kde je napsáno **true**.

repe	at forever
do	O if I ↓ sound level >> ↓ 10
	do
	else

Blok pod názvem **light** přetáhněte do **do** pole i do **else** pole. A změňte barvu na červenou v prvním políčku a zelenou ve druhém.



Běžte do sekce **Data** a přidejte blok s názvem **time series** a na místo, kde je napsáno **5**, přetáhněte **sound level** blok. Po spuštění programu se vytvoří graf úrovně zvuku.

repe	eat forever	
do	💿 if	t Q sound level >= 10
	do 👰	light 📲 Hub -
	else 🝳	light e
	l-lil time s	series P 🖳 sound level with color blue -

Graf se zobrazí, pokud například tlesknete:



Stisknutím tlačítka **Stop** program ukončíte.

4. Pohyb motorů

Tento příklad ukazuje, jak nastavit motory v modulu Fable Joint, aby se pohybovaly.

Vybavení

- Dongle (Hub)
- Fable Joint module
- Počítač s Fable Blockly

Popis

Každý Joint module lze ovládat v intervalu od -90 do 90 stupňů na motoru v ose X i Y.

V tomto programu je moduk nastaven tak, aby se nejprve posunul na ose X o 60 stupňů a poté o -45 stupňů na ose Y. Jednu sekundu později program provede pohyb směrem k úhlu -30 a 60 stupňů.

Program se ukončí krátce poté, co se motory zastaví.



5. Opakovaný pohyb

Tento příklad ukazuje, jak nastavit motory modulu Fable Joint tak, aby opakovaly stále stejný pohyb.

Vybavení

- Dongle (Hub)
- Fable Joint modul
- Počítač s Fable Blockly

Popis

Tento program využívá nekonečnou smyčku, díky které modul Fable Joint opakuje pohyb tam a zpět.

Program nejprve zajistí, aby se motory modulu posunuly do nastavených úhlů a na jednu sekundu se zastaví. Poté se začne pohybovat k další sadě úhlů a znovu počká. Nakonec se vrátí na vrchol smyčky a zopakuje celou sekvenci.



Program ukončíte stisknutím tlačítka **Stop**.

6. Držení úhlu motoru

Tento příklad ukazuje, jak naprogramovat motory Joint modulu k posunu do určitých úhlů a držení této pozice.

Vybavení

- Dongle (Hub)
- Fable Joint modul
- Počítač s Fable Blockly

Popis

Tento program používá nekonečnou smyčku, aby opakovaně přikázal Joint modulu přejít na konkrétní úhly a podržet je po dobu 10 sekund. Po vypršení časového limitu vypne motory pro úsporu baterie.



Tento druhý program ukazuje, jak zabránit načasovanému vypnutí modulu po 10 sekundách. To lze provést odečtením úhlů motoru a v tomto případě vykreslit úhly jako funkci času.

repeat forever								
do	The move to X: angle 60° Y: angle -45° on OXA -							
	LILI time series C SA angle of X - on OXA - with color blue -							

V obou případech je program zastaven stisknutím tlačítka **Stop**.

7. Postav si pick & place robota

V tomto úkolu se naučíte, jak sestavit a naprogramovat "pick & place" robota, který je často využíván například ve výrobě.

Naprogramujte jej tak, aby pohyboval předmětem (míčkem) z jednoho místa na druhé.



Úkol 1

Sestrojte robota a umístěte před něj podstavec pro míček tak, jak je znázorněno na obrázku. Nezapomeňte, že záleží na tom, jakým způsobem se otočí Joint modul.

Úkol 2

Naprogramujte robota tak, aby uchopil míček umístěný na pozici A a přesunul jej do pozice B.

Robot se musí poté vrátit do pozice A a zopakovat sekvenci.

Vyzkoušejte program víckrát a diskutujte o následujících otázkách:

- Funguje program správně pokaždé, nebo můžete být i spolehlivější?
- Je program dostatečně rychlý? Dokážete sekvenci zrychlit?

Úkol 3

Experimentujte s programem a robotem – pokuste se jej přizpůsobit a autonomizovat.

Například můžete pomocí kamery zjistit, kdy je míček umístěn na pozici A, poté může robot provést sekvenci, kterou míček přesune na pozici B.

Pick & place robot – krok za krokem

Tento příklad ukazuje, jak sestavit jednoduchého "pick & place" robota.

Vybavení

- Podstavec
- 2 stavební moduly (3XY nebo 4XY (dá se pracovat i s jedním))
- 2 balanční tyče na míček
- 1 vidlice
- 1 pingpongový míček
- 1 Fable Joint
- 1 Dongle (Hub)

Úkol 1

Sestrojte robota a umístěte před něj podstavec pro míček. Nezapomeňte, že záleží na tom, jakým způsobem se otáčí Joint modul.

Úkol 2

Naprogramujte robota tak, aby uchopil míček umístěný na jednom stojanu a přesunul jej na druhý. Poté musí robot vrátit míček zpět do prvního stojanu a zopakovat sekvenci.

První věc, kterou byste tedy měli udělat, je spustit smyčku s tímto blokem:

repeat forev	er
do	

a poté vložit všechny ostatní bloky do tohoto bloku.

Další krok: Program Fable začíná v jednoduché verzi, kdy je možné problém vyřešit pomocí tohoto bloku:



Ten najdete v sekci **Actions**. (Program poběží plynuleji, pokud změníte Fable Blockly na "Pokročilý režim" (najdete jej v levém dolním rohu.)



Nyní najděte další **move** blok v sekci **Actions**, s pomocí kterého budete moci ovládat rychlost robota. Vyberte tento blok:



Nastavte pohyb na rychlost 20 a nezapomeňte vybrat správné ID Joint modulu.

A pak byste měli vložit čekací blok **wait** mezi všechny bloky **move**:



Váš program by měl vypadat nějak takto:

repeat forever					
do	The move to X: I angle 0° Y: I angle 0° with speed: I 20 on OXA -				
	wait in sec.				
	The move to X: I angle 0° Y: I angle 60° with speed: 20 on 0XA -				
	wait in sec.				
	The move to X: I angle 60° Y: I angle 60° with speed: 1 20 on 0XA -				
	wait in sec.				
	The move to X: I angle 0° Y: I angle 60° with speed: 20 on 0XA-				
	wait in sec.				
	The move to X: I angle 0° Y: I angle 0° with speed: I 20 on OXA -				
	wait in sec.				
	The move to X: I angle 60° Y: I angle 0° with speed: 1 20 on 0XA -				
	🛛 wait in sec. 📭 2				

Několikrát program vyzkoušejte a diskutujte o následujících otázkách:

- Funguje program správně pokaždé, nebo můžete být i spolehlivější?
- Je program dostatečně rychlý? Dokážete sekvenci zrychlit?

Úkol 3

Experimentujte s programem a robotem – pokuste se jej přizpůsobit a autonomizovat.

Například můžete pomocí kamery zjistit, kdy je míček umístěn na pozici A, poté může robot provést sekvenci, kterou míček přesune na pozici B.

8. Sestavte třídícího robota

Naprogramujete robota tak, aby roztřídil LEGO kostičky podle barvy.

Automatizované systémy, jako je tento třídicí robot, jsou ve výrobním průmyslu běžné. Monotónní úkoly, jako je třídění odpadu nebo likvidace spálených sušenek, mohou provádět roboti, kteří se nikdy neunaví a nevyžadují mzdu.

Vybavení:

- Dongle (Hub)
- 1 Fable Joint modul
- 3 stavební moduly (lze provést i s jedním)
- 2 základny (lze provést i s jednou)
- 1 webová kamera (nebo kamera v počítači)
- LEGO kostky k postavení ramena a kostky v různých barvách, které budete třídit

Úkol 1

Sestavte robota podle obrázku a připojte kameru k počítači. Další skupiny mohou otestovat svůj program pomocí integrované kamery počítače. Kameru lze vybrat v **Setting** a **Camera source**.

Připojte webkameru podle obrázku:

Úkol 2

Naprogramujte robota, aby posunul Lego kostičku (nebo třeba lentilku) na pravou stranu, když stisknete pravou šipku, a na levou stranu, když stisknete levou šipku. Pořádně program otestujte.

Rozšiřte program o využití kamery, díky které robot může detekovat barvu předmětu a podle toho jej posunout doleva nebo doprava.

Úkol 3

Experimentujte s programem a s robotem – pokuste se proces zobecnit.

Můžete například naprogramovat pouze třídění předmětů určité velikosti a barvy doleva a zbytek doprava.





Vytvořte kód Úkol 2

Vezměte si smyčku forever loop a smyčku if do. Rozbalte if do a najděte else if a else jako na obrázku:



Zavřete nastavení.



true

0

do

do

else

else if

Poté přidejte tři **move to** bloky s rychlostí **30**. Ve všech třech blocích nastavte osu X na 10 a osu Y na -90 v části **if**, 90 v části **else if** a 0 v části **else**.

repe	eat forever
do	💿 if 🛯 🐹 key pressed? 🛛 left
	do Show move to X: angle 10* Y: angle -90* with speed. 1 30 on OXA -
	else if 🛛 🔀 key pressed? right
	do 🌮 move to X: Fangle 10° Y: Fangle 90° with speed: 130 on OXA -
	else Z ⁺ move to X: angle 10° Y: angle 0° with speed: 30 on OXA-

Úkol 3

Nyní bychom měli naprogramovat Joint modul pro třídění objektů podle barvy.

Vezměte program z minulého úkolu a změňte **key pressed?** blok.

Běžte do sekce s logem fotoaparátu – **camera** a vyberte bloky s názvem **color found?** a přidejte je k **if** a **else if**. Experimentujte s různým nastavením v bloku **color found?** a sledujte rozdíly v chování robota.



Podívejte se na video:

https://drive.google.com/file/d/1Wxj7oEeQXb6tzUdkuURFqWG44mAu-z-B/view?usp=sharing



9. Postavte robota, který vyřeší bludiště

Naprogramujete robota tak, aby různými způsoby vyřešil kuličkové bludiště.

Hry jsou zábavné, a proto není divu, že je herní průmysl stále roste. Proč nevyužít roboty jako nový typ herní platformy?



Úkol 1

Sestavte robota podle obrázku. Umístěte kuličku na malou plošinu v rohu bludiště.

Nainstalujte si aplikaci Fable Face do smartphonu a připojte ji pomocí Bluetooth.

Úkol 2

Naprogramujte robota tak, aby vyřešil bludiště tím, že bude pohybovat motory v různých úhlech. Hra končí, když kulička spadne do díry. Získáte 5 bodů za první díru, 10 za druhou a 15 za třetí díru. Snažte se, abyste dostali kuličku co nejrychleji a nejplynuleji skrz bludiště.

Úkol 3

Experimentujte s programem a robotem a vymyslete, jakými dalšími způsoby lze bludištěm projít.

Například:

- Modul ovládejte dálkově pomocí šipek na klávesnici
- Modul ovládejte dálkově pomocí akcelerometru ze smartphonu (s využitím aplikace Fable Face nebo Chromebooku (není dostupné u všech modelů)
- Můžete také modul ovládat pomocí dotykové obrazovky na smartphonu nebo Chromebooku
- Ovládejte modul dálkově pomocí jiného Joint modulu

Užitečné tipy

Při řešení bludiště je dobré naprogramovat Joint modul tak, aby se pohyboval trochu pomaleji než s normálním **move to** blokem.

Toto nastavení lze provést v pokročilém režimu (v levém dolním rohu):

Advanced Mode

V sekci **Actions** nyní najdeme **move to with speed** blok. Nejlepší bude nastavit rychlost na 20 nebo méně. Nezapomeňte však, že čekací bloky **wait in sec** by měly být delší než jedna sekunda.

Potom můžete zkusit sestavit program podobný tomuto:

A move to X: angle 0° Y: angle 0° with speed: 20 on 6YE -
wait in sec.
The move to X: I angle 10° Y: I angle 0° with speed: 20 on 6YE -
wait in sec.
A move to X: 1 angle 10° Y: 1 angle -10° with speed: 20 on 6YE -

Další způsoby řešení bludiště

Dálkové ovládání pomocí směrových kláves



Zrcadlení pohybu druhého modulu

Pokud máte 2 moduly Fable Joint, můžete jeden modul nastavit tak, aby jste s jeho pomocí mohli snadno ovládal ten druhý (lze to udělat v jednoduchém režimu). Zvolte **repeat forever** smyčku a **move to** blok.



Potom v Senses vyberte blok angle of X.



Poté změňte ID modulu na ID druhého Joint modulu.



Nyní duplikujte blok a změňte jeden z bloků na úhel **Y**. Potom přetáhněte dva bloky do **move to** bloku, kde úhly pro X a Y jsou:



Váš program by měl vypadat takto (oranžové bloky jsou ty, které řídí pohyb):

Po spuštění programu můžete použít jeden Joint modul k ovládání druhého – jako joystick (na Chromebooku a iPadu může docházet k určitému zpoždění, protože roboti jsou připojeni přes Bluetooth a tím může dojít ke zpomalení).

repeat forever						
do	🌮 move to X: 🛛		Y: O	ℬ angle of	Y - on 4ZE -	on 6YE -

Ovládání pomocí dotykové obrazovky na smartphonu nebo Chromebooku

Pokud nemáte 2 moduly Joint, můžete využít senzory buď v chytrém telefonu, nebo v zařízení Chromebook nebo iPad.

Znovu vyberte **repeat forever** blok a **move to** blok a poté v nabídce **Senses** vyberte blok s názvem **get acceleration on X-axis** a změňte jej na **tap position on X-axis**.



Nyní vytvořte kopii tohoto bloku a změňte ji na **tap position on Y-axis**. Poté přetáhněte tyto dva bloky do bloku **move to**, kde jsou úhly.

Takto by měl program vypadat na PC/Macu:

repe	eat forever				_
do	🎢 move to X: 🛛		⊚ get	tap position on Y-axis -	on 6YE -

Nejlepší je použít **move to** blok v pokročilém režimu, kde můžete pracovat s rychlostí a nastavit ji na 20, aby se modul pohyboval trochu pomaleji. Nyní můžete ovládat pohyb robota prstem na obrazovce!

Ovládání akcelerometrem

Akcelerometr můžete používat ve smartphonu nebo v zařízení Chromebook/iPad. Se stejným programem jako v minulém odstavci můžete změnit na zrychlení X a Y.



Pokud byste chtěli, aby se modul pohyboval o více než -10 až 10, můžete přidat matematický blok. Poté vezměte vstup ze senzoru a vynásobte jej 2 nebo 10. Přejděte do matematické sekce a vezměte dva bloky s názvem **1 x 1**.



Přetáhněte **get acceleration on** bloky z předchozí části a přidejte je do každého z bloků **1 x 1** (namísto jednoho z čísel). Poté změňte číslo na 2, 4, nebo na jiné libovolné (vyzkoušejte různé možnosti).



Až budete hotovi, přetáhněte nové bloky zpět do svého **move to** bloku a uvidíte, co se stane!



10. Sestavte robota, který kreslí pomocí laseru

Naprogramujete robota tak, aby kreslil geometrické útvary laserovým ukazovátkem.

Roboti vybavení výkonnými lasery se používají ve výrobě ke svařování kovových dílů nebo jejich dělení na konkrétní tvary.



Úkol 1

Sestavte robota podle obrázku a umístěte laserové ukazovátko tak, aby mohlo sledovat tvar.

Úkol 2

Naprogramujte robota, aby sledoval laserovým paprskem obvod různých geometrických útvarů. Pokuste se o co nejpřesnější pohyby.

Úkol 3

Experimentujte s programem a robotem.

Například můžete naprogramovat robota tak, aby se při stisknutí mezerníku zastavil, nebo můžete do programu zahrnout různé proměnné a matematické vzorce, které usnadňují kreslení nových geometrických tvarů.

Vyzkoušejte si složitější programování.

I v tomto případě je nejlepší použít blok **move to with speed**:



Můžete také použít bloky if, abyste mohli část robota ovládat pomocí různých kláves.

Důležité upozornění: Pokud jste v pokročilém režimu, blok **key pressed** se trochu liší. Na kterou klávesu má robot reagovat jako na spouštěč musíte zadat tak, že na ni kliknete a tím příkaz vložíte. Díky tomu můžete používat celou klávesnici!



11. Vrhací robot

Tento příklad ukazuje, jak sestavit vrhacího robota, který umí hodit míčkem pomocí pružného ramene.

Vybavení:

- Dongle (Hub)
- Joint modul
- Vrhací rameno (katapult)
- Pingpongový míček
- Počítač nebo tablet s nainstalovaným Fable Blockly

Popis:

Připojte vrhací rameno podle obrázku.

Vytvořte program, který přejde do polohy, kde může vrhací rameno držet míček aniž by spadnul. Poté vyberte if blok:

do

a nastavte, jaká klávesa program spustí.





Potom použijte pohybový blok s rychlostí.

Poznámka: Program musíte nastavit v rozšířeném režimu v levém dolním rohu:



Nastavte rameno tak, aby se před hodem zaklonilo dozadu (tak využije svou pružnost ke zrychlení pohybu míčku). Rychlost nastavte na 100 %.



Program by měl vypadat podobně tomuto (nezapoměňte vybrat správné ID modulů):



Vrhací rameno funguje pouze při stisknutí mezerníku.

12. Hlídací robot

Sestavte a naprogramujte robota, který bude monitorovat danou oblast a fotografovat pohyb lidí, které zaznamená.

Inteligentní bezpečnostní systémy se již používají na mnoha místech. Kombinace s inteligentními roboty, kteří jsou mobilní, otevírá nové možnosti v mnoha oborech.



Vybavení:

- Joint modul
- Webová kamera
- Nějaký držák na přichycení webkamery
- 2 stavební moduly (stačí i jeden)
- 1 víko
- Dongle (Hub)

Úkol 1

Sestavte robota podle obrázku a připojte webkameru k počítači. Ostatní skupiny mohou otestovat program pomocí integrované kamery počítače. Kameru lze vybrat v rozhraní **Settings** a **camera source**.

Úkol 2

Naprogramujte robota tak, aby monitoroval místnost, ve které je umístěn: tiše, plynule a pravidelně.

Pokud robot detekuje pohyb, musí pořídit obrázek. Obrázky by měly být automaticky ukládány do složky na ploše počítače.

Naprogramuj robota tak, aby spustil poplach při detekci osoby v místnosti.

Úkol 3

Experimentujte s programem a robotem – pokuste se vytvořit program, která bude přesněji reagovat na pohyb a bude uživatelsky přívětivější.

Například naprogramujte mezerník jako tlačítko pro zapnutí / vypnutí robota. Můžete také kombinovat detekci pohybu s detekcí zvuku s využitím mikrofonu počítače, aby robot mohl být ještě přesnější při detekování lidí.

Vytvořte kód

Pokud sestavíte robota jako obrázku, nastavte pohyb na ose Y buď 90 nebo -90 stupňů. Tím zajistíte, aby se robot otáčel ze strany na stranu a hlídal pohyb. Pokud spatří pohyb, měl by změnit barvu donglu na červenou, spustit alarm a vyfotografovat vetřelce.

Nejprve vytvořte dvě proměnné nazvané **x** a **myModule**.

Poté **myModule** nastavte na ID vašeho Joint modulu a proměnnou **x** na 90.



Přidejte smyčku **repeat forever** a **if do else** blok. Vyberte proměnnou **set x to**, přesuňte ji pod matematický blok s názvem **1 - 1** a pod první 1 přidejte proměnnou **x**. Mělo by to vypadat takhle:



Tento blok vložte pod **else**.

V sekci **else** vyberte **move to with speed** blok a nastavte Y-motor na 90 (nebo -90) a X-motor na proměnnou **x.** ID nastavte na proměnnou **myModule** a rychlost na 30.

C							
🎖 move to X: 🛚	х •	Y: (angle 90°	with speed:	30	on # 🔻	myModule *

Vložte blok pod **else**.

Nyní přidejte další **if do** do **else**. Vyberte porovnávací blok v sekci **Logic** a přidejte jej pod **if**. Poté vyberte proměnnou **x**, vložte ji pod první parametr, aby to bylo <, a v části pro číslo zadejte -90. Bude to vypadat takto:



Poté přidejte další **set x to 90** blok jako na začátku. To zajistí, aby se modul vrátil zpět na 90 poté, co dosáhne téměř -90 na Y-motoru. Přidejte **wait in** a zadejte 0,5 sekundy.



Dále musíte naprogramovat, co robot udělá, když detekuje pohyb.

V sekci **Camera** vyberte blok **motion found?** a přidejte ho k **if**.

Nyní přidejte blok pro dongle, kterým změníte jeho barvu na červenou.

V sekci **Camera** vyberte blok s názvem **take picture** a **wait** blok, který nastavíte na 3 sekundy.

Poté na konec bloku **if** přidejte blok s názvem **reset color**.

Teď by váš program měl vypadat takto:



Fotoaparát se otevře ve vyskakovacím okně a pro jeho fungování budete muset udělit oprávnění. Obrázky, které program pořídí, se uloží do složky v Dokumentech nazvané **Fable images**. V Pythonu by program vypadalo takto:

1	myModule ='6YE'
2	x = 90
3 -	while True:
4 -	<pre>if api.detectedMotion(2):</pre>
5	<pre>api.setColor([100,0,0], 'Hub')</pre>
6	<pre>api.setColor([100,0,0], myModule)</pre>
7	<pre>api.playSound("angry.wav", 'PC')</pre>
8	api.takePicture()
9	api.wait(2)
10	<pre>api.recoverChannelColor("Hub")</pre>
11	<pre>api.recoverChannelColor("myModule")</pre>
12 -	else:
13	api.setPos(x,90,myModule)
14	<pre>api.setSpeed(30,30, myModule)</pre>
15	x = x - 1
16	api.wait(0.05)
17-	if x < -90:
18	x = 90
19	api.wait(0.5)

Uvědomte si, že v Pythonu lze nastavit prodlevu **wait** nižší než 0,1, takže je zde 0,05.

13. Společenský robot

Naprogramujete robota, který bude komunikovat s lidmi.

Robot musí být naprogramován tak, aby reagoval na zvuk a pohyb, když uvidí osobu.

Očekává se, že sociální roboti postupně přijmou mnoho různých druhů zaměstnání, např. v obchodech, hotelech a v sektoru sociální péče.



Úkol 1

Nainstalujte aplikaci Fable Face na chytrý telefon (z App Store nebo Google Play) a sestavte robota podle obrázku.

Otevřete Fable Face a spárujte ji s donglem.

Úkol 2

Naprogramujte robota, aby zamával, usmál se a vydával různé zvuky, pokud zachytí pohyb.

Jako výchozí kameru použijte integrovanou kameru počítače, ale můžete také použít externí webovou kameru.

Naprogramujte robota tak, aby paže sklonil, když uplyne delší doba od posledního kontaktu s člověkem. Můžete také naprogramovat, aby robot změnil výraz a vypadal unaveně, když se chvíli nic neděje.

Úkol 3

Experimentujte s programem a robotem – zkuste vymyslet, jak aby komunikace s ním mohla být ještě zábavnější.

Například můžete využít mikrofon počítače k detekci hlasitých zvuků, které mohou "probudit" robota.

Tento příklad ukazuje, jak sestavit jednoduchého sociálního robota, který je spouštěn pohybem.

Vybavení

- Dongle (Hub)
- 2 Joint moduly
- 2 stavební moduly 4XY
- Modul na držení telefonu
- Smartphone s nainstalovanou Fable Face App

Popis:

Sestavte robota podle obrázku:



Příklad společenského robota

Program primárně využívá zabudovanou kameru v PC. Pokud zaznamená pohyb, robot změní výraz, zasměje se a začne pohybovat pažemi nahoru a dolů.

To pokračuje tak dlouho, dokud je v záběru kamery pohyb. Když pohyb ustane, robot svěsí ruce a zatváří se unaveně.

Jak vytvořit kód

Ve Fable Blockly se ujistěte, že jste v **Advanced mode**. Začněte s nastavením tří různých proměnných: **left arm**, **right arm** a **X**.



Set Left arm a **Set Right arm** vložte nahoru a z nabídky **Senses** vyberte **module** blok a upravte jej na správné ID:



Duplikujte blok a změňte na **Right arm**.

set	Left arm 👻	to 🕻	module	JCC -
set	Right arm -	to	module	OXA -

Teď přidejte smyčku **repeat forever** a **if then else** blok.



V sekci **Camera** vyberte **motion found?** a přidejte k **if** bloku.



Poté vyberte blok **repeat while**, duplikujte **motion found?** blok a přidejte ho tam, kde je napsáno **true**.



V sekci **math** vyberte **sin** blok a **1 x 1** blok a duplikujte jej.



Pak napište **45** do prvního číselného pole, vezměte blok **sin** a vložte jej do druhého číselného pole.



Poté ve druhém **1 x 1** bloku vepište 250 do prvního číselného pole, pod **senses** přesuňte blok **time in sec.** a vložte jej do druhého číselného pole.



Nyní vezměte poslední číselný blok a vložte jej do prvního, kde je uvedeno **angle**



a potom na začátek bloku přidejte proměnnou set X to.



Tento nový blok bude přecházet do repeat while bloku.



Vyberte **move to** blok a přidejte proměnnou **X** k X motoru. U **Y** by mělo stát 60.

Pod ID sjeďte dolů k # a přidejte proměnnou **Left arm**.



Tuto proměnnou duplikujte a změňte na **Right arm** a přidejte obě k **repeat while** smyčce.

set Left arm - to t module JCC -	set L
repeat forever	repea
do 🧿 if 🕇 🙆 motion found? amount 📭 2	do
do repeat while - 1 () motion found? amount (2)	
do set X - to 1 #45 × - sin - #250 × - # () time in sec.	
A move to X: • X • Y: • angle 60• on #• • Left arm •	
The second secon	
else	

Mimo **repeat while** smyčku (ale pořád v bloku **if do**) by měly být dva **move** bloky, které stahují paže dolů – to znamená **90** nebo **-90**.

Nejdříve ale vyberte **wait in sec** blok a nastavte jej na 3 sekundy.

set	Left an	m -	to C module JCC -
set	Right a	arm 👻	to module OXA -
repe	at fore	ver	
do	۲	1	The motion found? amount
	do	repe	eat while - 1. O motion found? amount 2
		do	set X - to 1 45 X sin - 6 250 X C (time in sec.
			P move to X: ▼X Y: angle 60° on # Left arm ▼
			The move to X: The X of Y: C angle 60* on # - The Right arm -
		8	wait in sec. 03
		P	move to X: angle 90* Y: angle 0* on #- teltarm -
		8.	move to X: I angle -90* Y: I angle 0* on #- I Right arm -
	else	C	

Pak, pokud máte připojený Fable Face, můžete přidat bloky **expression** a **set color to**, díky kterým můžete měnit výraz robota a barvu jeho očí.

Výraz (expression) nastavte třeba na happy a barvu (set color to) na blue.

Přidejte tyto bloky na vršek if do.



Poté dvakrát zkopírujte blok **expression**.

Jeden z nich změňte např. na **neutral** a ten další na **sad** nebo **tired**. První blok potom přesuňte za blok **wait in 3 seconds** a druhý do bloku **else**.

set	Left an	m - to module JCC -
set	Right a	rm - to C module OXA -
repe	at fore	ier i
do	۰	r r (O) motion found? amount
	do	expression happy -
		💿 set color to iris - I
		repeat while - 1 (i) motion found? amount (2)
		do set X - to 45 x sin - 250 x C time in sec.
		The move to X: IX Y: Langle 60° on #- Left arm -
		The second secon
		🛛 wait in sec. 🖬 3
		expression neutral →
		The move to X: I angle 90* Y: I angle 0* on #- I Left arm -
		The move to X: I angle -90° Y: I angle 0° on # - FRight arm -
	else	expression tired -
	_	

Robot může vydávat i zvuky. Můžete si vybrat některý z již předprogramovaných zvuků v bloku **play sound** (například smích – **laugh**).



Můžete také nahrát vlastní zvuk (.mp3 nebo .wav), uložit jej do složky nazvané: **Documents/Fable/My Fable Sounds** a poté ho přidat na začátek bloku **if**.



Nyní by váš program měl vypadat takto:



Video o tom, jak vytvořit kód: https://drive.google.com/open?id=1yhFoInpHuRLrNlCkKij2XOOM83-PEhSd



Pokud kód napíšete v Pythonu, bude vypadat takto:

```
1 import math
 2
3 Left_arm = None
 4 Right_arm = None
5 X = None
6
 7
8 Left_arm = 'JCC'
9 Right_arm = '0XA'
10 while True:
11 if api.detectedMotion(2):
       api.playSound("laught.wav", 'Face')
12
       api.setFaceEmotion('Happy')
13
       api.setEyeColor('Iris', [20, 40, 100])
14
       while api.detectedMotion(2):
15 -
16
         X = 45 * math.sin((250 * api.getTime()) / 180.0 * math.pi)
         api.setPos(X, 60, Left_arm)
17
         api.setSpeed(50, 50, Left_arm)
18
         api.setPos(X, 60, Right_arm)
19
20
         api.setSpeed(50, 50, Right_arm)
21
       api.wait(3)
22
       api.setFaceEmotion('Neutral')
23
       api.setPos(90, 0, Left_arm)
24
       api.setSpeed(50, 50, Left_arm)
25
       api.setPos(-90, 0, Right_arm)
26
       api.setSpeed(50, 50, Right_arm)
27 -
     else:
28
       api.setFaceEmotion('Tired')
```

14. Zpracování dat pomocí seznamů

Tento příklad ukazuje, jak vytvořit seznamy a provést některé jednoduché zpracování dat.

Vybavení:

- Dongle (Hub)
- Počítač s nainstalovaným Fable Blockly

Nejprve vytvořte seznam v jedné z nejjednodušších a nejdůležitějších datových struktur v programovacím jazyce (Blockly nebo Python).

Seznam je v podstatě sbírka položek, například čísel. Seznam shromažďuje položky postupně v několika datových strukturách.

Úkol 1

Vytvořte proměnnou nazvanou **data** a přiřaďte ji k seznamu s položkami, např. 9, 43, -11.

Úkol 2

Vytvořte nový prázdný seznam, nechte jej změřit hladinu zvuku a připojte seznam s tímto měřením.

Seznam bude rychle narůstat. Současně vytvoří graf, kde zjistíte například nejvyšší číslo ze seznamu (graf naroste pokaždé, když se změří nový záznam zvuku).

Jak vytvořit kód Úkol 1

Ujistěte se, že máte nastavený pokročilý režim ve Fable Blockly.

Vytvořte proměnnou s názvem **Data**.



Vyberte nový proměnný blok Set Data to:



V sekci Lists vyberte create list with a přidejte jej k Set Data to – čísla nastavte libovolně.



Takto by vypadal stejný program v Pythonu:



Úkol 2

Použijte znovu Set Data to, ale odstraňte create list with. Místo toho přidejte create empty list:



Poté v lists vyberte blok in list .. set ..as (poslední blok).



Přidejte proměnný blok **Data** do **create empty list**, změňte **set** na **insert at**, změňte **#** na **last** a namísto **5** vyberte z nabídky **Senses** blok **Sound level**.



V sekci Data vyberte blok s názvem Time series.



Potom v sekci Math vyberte blok sum of list.



Změňte sum na max a přidejte proměnný blok Data, kde bude napsáno create list....



Nyní tento nový blok přidejte pod dříve zmíněný blok time series, kde stojí hodnota 5.



Dokončete program přidáním dvou posledních bloků (**in list** a **time series**) do **repeat forever** smyčky. Program bude vypadat takto:



V Pythonu by kód vypadal takto:

1	Data = None
2	
3	
4	Data = []
5 -	while True:
6	<pre>Data.append(api.getSoundLevel())</pre>
7	api.plot(max(Data), "A")

Jak vytvořit kód – video: https://drive.google.com/open?id=1wtdn51_cNFflVpjtmtvffTFnPfPr2ExC



15. Kopírování pohybu

Vybavení

- 2 Joint modul
- 2 víka (dá se udělat i s jedním)
- 1 Dongle (Hub)
- Držák na telefon
- Smartphone s nainstalovaným Fable Face (úkol lze provést i bez něj)
- Počítač s nainstalovaným Fable Blockly

Popis

Přidělejte každý Joint modul na kryt a na jeden z nich přidělejte držák telefonu.

Spusťte Fable Blockly v počítači a připojte dongle k USB portu.

Ujistěte se, že Joint modul a dongle mají stejnou barvu a připojte telefon (nezapomeňte zapomnout Bluetooth).



Nyní sestavte program podobný tomuto:

set Leader - to & module CVB -
set Follower - to In module L3E -
set X - to create empty list
set Y + to create empty list
O expression tired -
💿 set color to iris - 💼
repeat 1 20 times
do in list 1 X - insert at - last - as D 3- get angle of servo X - on # - D Leader -
in list "Y-" insert at - last - as to 2- out angle of servo Y - ou # - C Leader -
wait in sec. # 0.5
save X in logfile Fable-log.csv
ave several in logfile Fable-log.csv
expression happy -
💿 set color to iris - 📭
count with iv from a 1 to a 20 by a 1
do 🚰 move to X. 🖬 in list 🛛 X - get - 🖉 - li - Y - t in list 🛛 Y - get - 🖉 - t - on # - t Follower -
X wait in sec. 0.5

Jak bude program fungovat

Vyberte Joint modul s názvem **Leader** a hýbejte s ním v různých úhlech. Díky tomuto programu se bude Joint modul s názvem **Follower** pohybovat stejně jako Joint modul s názvem **Leader**. Jakýkoliv pohyb provedený s **Leader**em bude napodoben **Follower**em.

Vytvořte kód

Nejprve se ujistěte, že jste v pokročilém režimu ve Fable Blockly. Poté vytvořte čtyři proměnné s názvy **Leader**, **Follower**, **X** a **Y**. Proměnné **Leader** a **Follower** přiřaďte ke dvěma různým Joint modulům a nastavte **X** a **Y** na **create empty list**.



V nabídce Lists vyberte blok create empty list.... Přidejte proměnnou X tam, kde stojí create empty list, změňte # na last, namísto 5 vyberte v Senses blok s názvem get angle of a vyberte modul Leader:

in list	X -	insert at 👻	last 🗸 as 🕻	P get angle of servo X - on # - t Leader -

Zkopírujte tento nový blok a proměnnou změňte na **Y** a **angle of** změňte na **servo Y**. Nyní vložte tyto dva bloky do smyčky **repeat 20 times** a přidejte blok **wait in sec** na konec smyčky a změňte jej na 0,5:

repe	at 20 times		
do	in list 🛛 🗙 🚽 insert at 🗸	last → as 🌘	get angle of servo X → on # → b Leader →
	in list Y - insert at -	last 🕶 as 💧	ở get angle of servo Y → on # → t Leader →
	X wait in sec. 0.5		

V nabídce **Data** vezměte blok **save** a přidejte proměnnou nazvanou **X**. Poté nový blok zkopírujte a změňte jej na **Y**:



Vyberte smyčku s názvem **count with i**.



Poté v lists vyberte in list create ... get # 1.



Přidejte proměnnou **X** do **create empty list**, duplikujte **X** a přidejte ji na konec (kde stojí 1) a změňte ji na **i** (proměnná s názvem i je skutečně vygenerována).



Duplikujte tento nový blok, změňte první proměnnou v druhém novém bloku na Y, přidejte je obě do **move to** bloku, který přesune **Follower** a nastaví **move** blok do **count** smyčky. Potom přidejte na konec smyčky **wait in sec** s hodnotou 0,5.



Můžete přidat také výraz v aplikaci Fable Face.

Můžete také napsat kód v Pythonu:

1 Leader = None 2 Follower = None 3 X = None4 Y = Nonei = None 5 6 7 8 Leader = '6YE'9 Follower = '4ZE' 10 X = [7]11 Y = [] $12 \cdot \text{for count in range}(20)$: X.append(api.readJointSensor('angleX', Leader)) 13 Y.append(api.readJointSensor('angleY', Leader)) 14 15 api.wait(0.5) 16 api.log(X, 'Fable-log.csv') 17 api.log(Y, 'Fable-log.csv') 18 for i in range(1, 21): api.setPos(X[int(i - 1)], X[int(i - 1)], Follower) 19 api.setSpeed(50, 50, Follower) 20 api.wait(0.5) 21

V tomto videu najdete inspiraci: https://youtu.be/SLzWD5jNGQg



Stáhněte si hotový programový soubor: copy cat – follow the leader https://drive.google.com/file/d/15podNVyLb2DvtXirQcRLhxPchbWnaYar/view?usp=sharing



16. Fable ukazuje sever

Vybavení:

- Dongle (Hub)
- 2 stavební moduly
- 2 Joint moduly
- 1 víko
- 1 držák na telefon
- 1 telefon s nainstalovaným Fable Face (najdete v App Store nebo Google Play)
- Volitelně příslušenství (aby bylo lépe vidět, kam směřuje Fable)
- Počítač s nainstalovaným Fable Blockly



V tomto projektu použijeme magnetometr nebo kompas, který je v chytrých telefonech.

Fable ukazuje sever

Musíte postavit a naprogramovat robota tak, aby vždy směřoval na sever (nebo jiným směrem).

Začněte shlédutím tohoto videa: https://www.youtube.com/watch?v=Ec4hlUSzFVo



Nejprve musíte zjistit, jakým směrem je sever. To lze provést pomocí kompasu nebo s pomocí Google map tak, že si najdetet budovu, kterou znáte. Sever je na mapě vždy nahoře, takže s pomocí mapy a záchytného bodu v okolí můžete přibližně určit, jakým směrem je sever.

Poté je dalším logickým krokem sledování výstupu z magnetometru (senzoru kompasu) v telefonu ve Fable Blockly.

To lze provést vytvořením následujícího programu:



Když je program spuštěn, otáčejte telefonem, dokud nezískáte stabilní číslo.

Poznamenejte si, v jakých intervalech magnetometr vytváří výstup v závislosti na tom, jak s telefonem otáčíte.

Jakou hodnotu vám dává magnetometr, když míříte na sever?

Nyní je čas naprogramovat robota.

Musíte vzít v úvahu, že magnetometr pracuje se stupni od 0 do 360, ale modul Fable Joint se pohybuje pouze od -90 do 90 stupňů. Proto bude nutné použít dva moduly společně tak, že jeden modul bude pokrývat počátečních 180 stupňů (0 – 180) a druhý potom následujících 180 stupňů (180 – 360).

Proto musíte vytvořit proměnnou, kterou můžete nazvat **"magnetometer**" nebo "**angle**", a dát ji do stavu, v němž bude uvedeno, že:

- Pokud je Magnetometer/angle větší než 180 stupňů, musí se pohybovat "pravá ruka" ve vztahu k magnetometru.
- Jinak (pokud je menší než 180 stupňů), musí se pohybovat "levá ruka".

Uvědomte si, že je třeba provést matematický výpočet, od 360 stupňů magnetometru po -90 – 90 stupňů robota. Proto budete potřebovat tento blok:



Poté vyberte **if else** blok ze sekce **Logic**.



Začněte využitím přímého čísla z magnetometru.

Robot se trochu otřese kvůli působení magnetických polí.

Proměnnou magnetometru nejprve nastavte na 0 (mimo smyčku) a poté uvnitř; tentokrát to nastavte na senzor:



Nyní přidejte tento blok k **if**:



Přidejte proměnnou k jednomu z čísel a nastavte magnetometr na hodnotu vyšší než 180.

set repe	magnetometer - to to 0
do	set magnetometer - to to get magnetometer -
	😧 if 🔄 magnetometer - 🕞 - 🕇 180
	do
	else
	LILL time series D 💿 get magnetometer - with color blue -

Přidejte dva bloky **move to** do **if** a dva **move to** bloky do **else**. V prvním bloku nastavte **X** na 0 a **Y** na 270 – proměnnou. Druhý **move to** blok v **if** nastavte na -90 a 0.



Potom v bloku **else** nastavte první **move to** blok na **90** a **0**. A druhý **move to** blok na **0** a **variable -90**. Nyní program vypadá takto:

set	magne	stometer - to 0
repe	at fore	ver
do	set	nagnetometer - to 1 💿 get magnetometer -
	۲	magnetometer - >- t 180
	do	The move to X: planck 0° Y: C 270 - magnetometer - on OXA -
		The move to X: plangle 0* Y: plangle 0* on JCC -
	else	The move to X in angle 90* Y: a angle 0* on OXA -
		The move to X: Fangle 0* Y: C magnetometer 0 90 on JCC -
	Em (time series a content magnetometer - with color blue -

Která ruka je OXA v tomto programu?

Která ruka je JCC v tomto programu?

Uvažujte, že úhel Y bude vždy v intervalu -90 – 90 stupňů.

Bonusový úkol

Na vstupu, který dostanete z magnetometru, je mnoho rušení. Proto můžete zkusit vymyslet, jak to udělat, aby byly pohyby robota plynulejší.

TĚŽKÉ: Zkuste, jestli můžete naprogramovat algoritmus, který převede vstup na přímější linii (aby to vypadalo spíše jako lineární rovnice).

Bonusové info

Důvodem velkého rušení magnetometru je, že kolem nás je mnoho magnetických polí a skutečnost, že magnetická pole Země nejsou zvlášť silná. Přečtěte si více o magnetických polích zde:

https://www.khanacademy.org/science/physics/magnetic-forces-and-magnetic-fields/



https://en.wikipedia.org/wiki/Magnetic_field



Možné řešení, když vezmeme v potaz rušení

set magnetometer - to t 0. repeat forever		
do	set	nagnetometer - to . 0.1 x . 00 ext magnetometer - + . 0.9 x . magnetometer -
	۲	magnetometer - >- = 180
	do	The move to X stands Of Y: 1 270 - I magnetometer - on OXA -
		The move to X is angle -90° Y: Is angle 0° on JCC -
	else	P* move to X: 1 angle 90* Y: 1 angle 0* on OXA -
		The move to X is angle Of Y: to magnetometer + 90 on UCC -
Ltt time series a 💿 get magnetometer - with color blue -		time series I 💿 get magnetometer - with color blue -
	ш	time series magnetometer - with color blue -

Pokud zapíšete program v Pythonu, mohl by vypadat takto:

```
1 magnetometer = None
 2
 4 magnetometer = 0
 5 while True:
     magnetometer = 0.1 * api.readFaceSensor('compass') + 0.9 * magnetometer
 6
     if magnetometer > 180:
       api.setPos(0, 270 - magnetometer, '0XA')
 8
 9
       api.setSpeed(50, 50, '0XA')
       api.setPos(-90, 0, 'JCC')
10
11
       api.setSpeed(50, 50, 'JCC')
12 -
     else:
13
       api.setPos(90, 0, '0XA')
14
       api.setSpeed(50, 50, '0XA')
       api.setPos(0, magnetometer - 90, 'JCC')
15
       api.setSpeed(50, 50, 'JCC')
16
     api.plot(api.readFaceSensor('compass'), "A")
17
18
     api.plot(magnetometer, "A")
```

Graf by mohl vypadat takto:



Pokud chcete vědět, jak funguje magnetometr, podívejte se zde: <u>https://www.youtube.com/watch?v=_ZiLwoClRGQ</u>



Pokud vás zajímá, proč byste měli zvážit magnetometr s 0,1, můžete si přečíst: <u>https://en.wikipedia.org/wiki/Moving_average#Exponential_moving_average</u>



Užijte si programování!

×

UAWEI



Výhradní distributor:

MORAVIA Consulting spol. s r.o. Olomoucká 83 Brno 627 00, Česká republika Telefon: +420 541 422 511 Email: info@moravia-consulting.com

www.moravia.education

www.vyuka-vzdelavani.cz