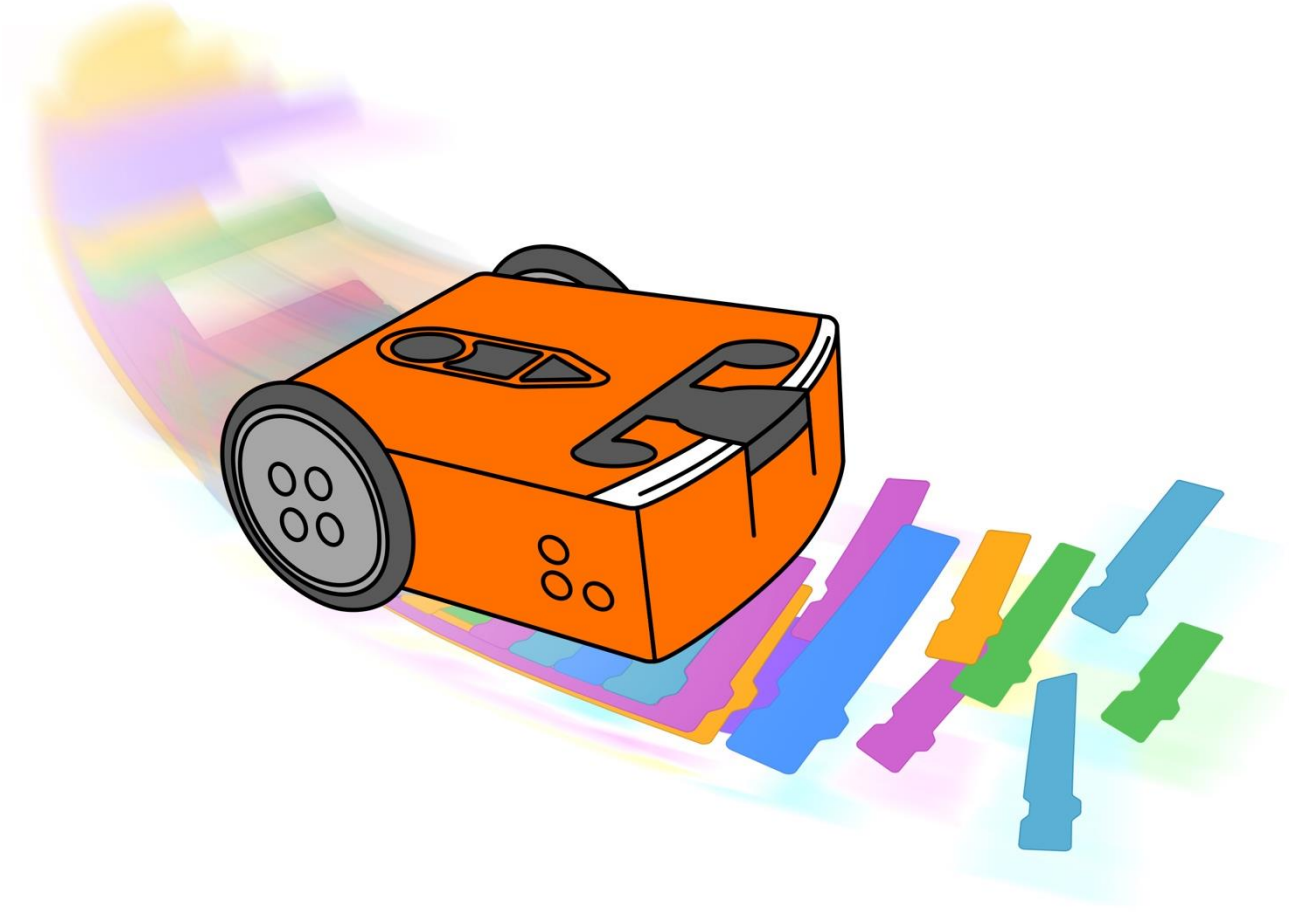
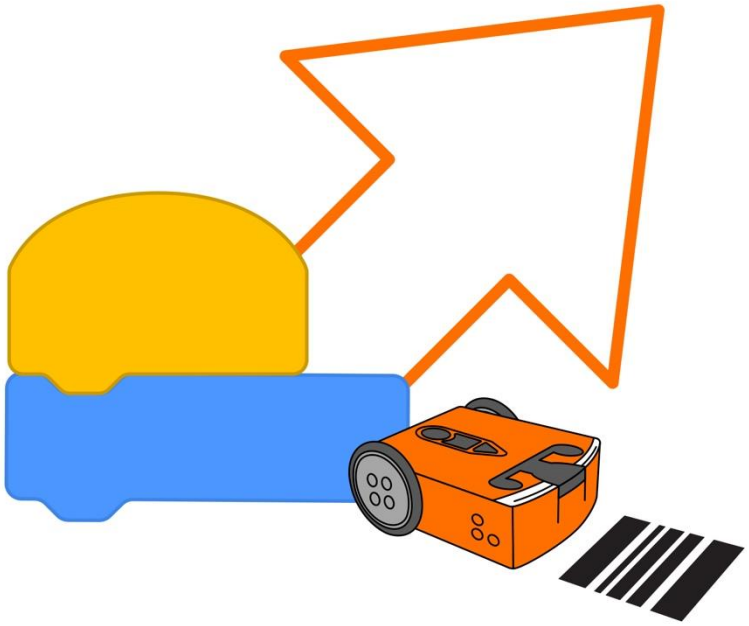
Výukové aktivity EdScratch

Pracovní listy pro studenty



Obsah

Lekce 1: Začínáme

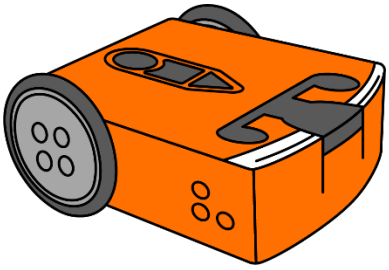


Edison

Jméno_____

U1-1.1 Prozkoumejte roboty Edison

Toto je Edison, programovatelný robot.



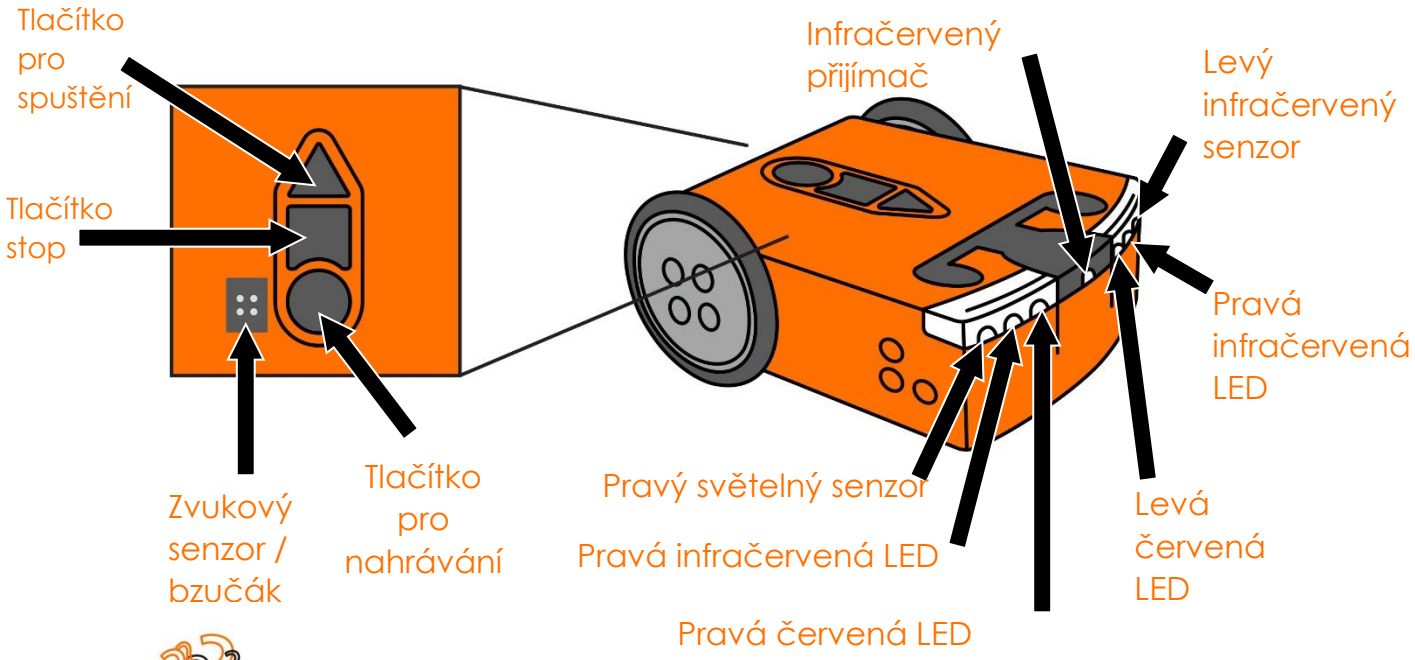
S našimi roboty Edison toho dokážeme mnoho. Můžeme je například naprogramovat, aby používali své motory, blikali LED světly nebo vydávali zvuky. Roboty Edison můžeme využít i k vyrábění robotických výtvorů, vytváření či procházení bludišť a k mnoha dalším úkolům!

Než začneme Edisona používat, měli bychom se o tomto robotovi dozvědět trochu více.

Edison používá k interakci se světem senzory a motory. Robot je rovněž vybaven třemi tlačítky, vypínačem a několika odnímatelnými částmi. Pokud budete vědět, kde se jednotlivé části Edisona nacházejí a k čemu slouží, pomůže vám to lépe robota využívat.

Úkol 1: Podívejte se na Edisona seshora

Podívejte se na horní část robota Edison. Zkuste najít všechny součásti označené na obrázku.



Proč je to tak?

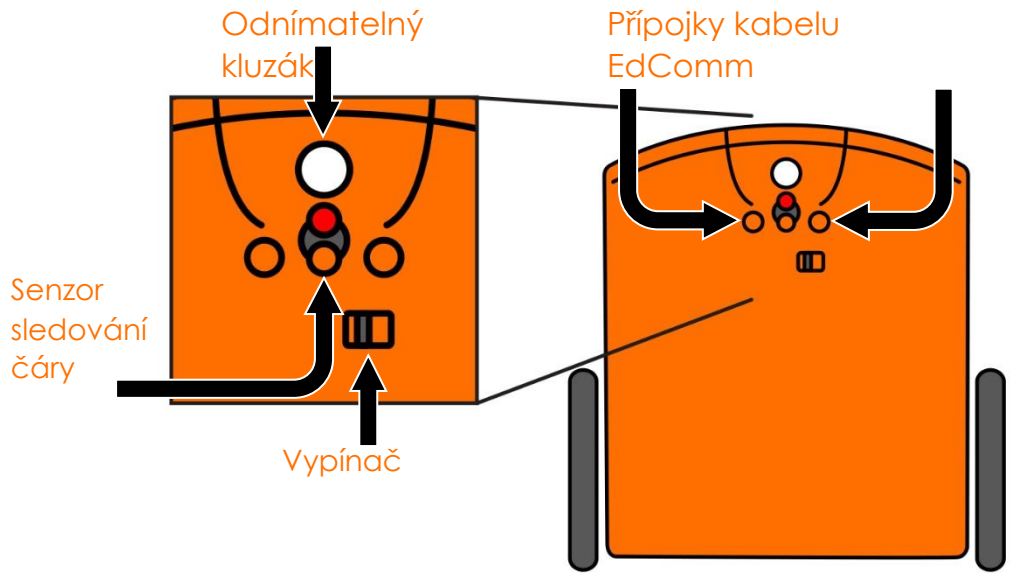
Horní část Edisona je vyrobena z čirého plastu. Tak můžete vidět elektronické komponenty, díky kterým Edison funguje. Jednou z nejdůležitějších částí je černobílý čtverec, který leží těsně nad špičkou trojúhelníkového tlačítka 'play' (spuštění). Vidíte jej?

Jde o mikročip robota. Mikročip je v podstatě malý počítač, proto se mu někdy říká mikropočítač. Obsahuje centrální procesorovou jednotku (CPU). Je to vlastně Edisonův mozek!

Jméno_____

Úkol 2: Podívejte se na Edisona zespoda

Obraťte Edisona. Podívejte se na obrázek a zkuste najít všechny součástky označené na obrázku přímo na svém robotovi Edison.



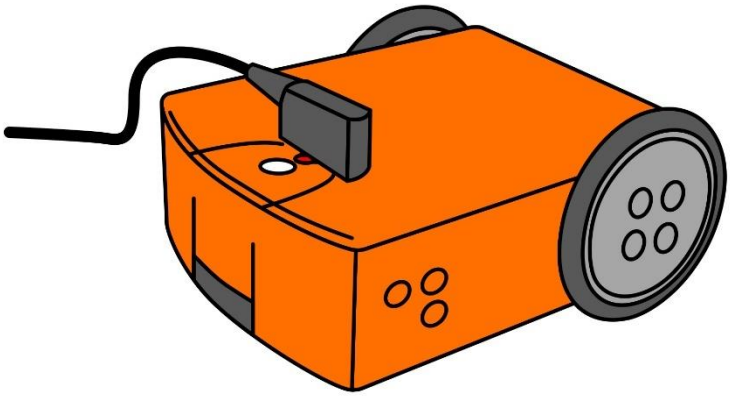
Úkol 3: Vyjměte a připevněte kola, kluzák a kabel EdComm

Někdy možná budete chtít použít Edisona různými způsoby, např. položeného na boku. Proto je možné některé součásti Edisona odpojit. Obě Edisonova kola lze sundat. Pokuste se oddělat jedno z kol jeho vytažením ven z robota. Podívejte se na napájecí zdířku, do které se kolo připojuje. Nezapomeňte vrátit kolo zpět!

Pak si prohlédněte Edisonův plastový kluzák. Tento kluzák je kousek průhledného plastu na spodní části Edisona, poblíž senzoru pro sledování čáry. Většinou by měl kluzák zůstat v robotovi. Kluzák je velmi malý. Protože je z průhledného plastu, může být snadno přehlédnutelný. Pokud jej oddělate, dávejte na něj dobrý pozor. Neztraťte jej!

Další součástí Edisona, kterou budete často používat, je kabel EdComm.

Kabel EdComm budete používat ke stahování svých programů do Edisona z programovacího zařízení, např. ze svého počítače. Kabel EdComm má na jednom konci připojení pro Edisona, zatímco druhý konec se připojuje ke sluchátkovému konektoru v počítači.



Vyzkoušejte si připojení kabelu EdComm do Edisona.

Jméno_____

Úkol 4: Zapněte Edisona

Kdykoliv chceme Edisona použít, musíme jej nejprve zapnout. Zkuste nyní Edisona zapnout.

1. Co se stane, když zapnete robota? Popište, co se děje, včetně toho, co jste viděli a co jste slyšeli. Svou odpověď napište sem:

_____

_____

_____

_____

_____

Nezapomeňte

Kdykoli dokončíte používání Edisona, ujistěte se, že jste robota opět vypnuli!

Jméno_____

U1-1.1a Změňte to: Cihly, bloky a Edison

Podívejte se na Edisona pořádně. Vidíte všechny ty výstupky a díry v horní části robota, na bocích i zespoda?

Výstupky podobné těm, které má Edison na své horní části i na kolech, jste již nejspíš viděli. Proč si myslíte, že na robotu tyto výstupky jsou, spolu s dírami na bocích i ve spodní části?

Jde o přípojné body, s jejichž pomocí je možné Edisona rozšířit a stavět s jakýmkoli stavebnicovým systémem kompatibilním s LEGEM.

Pomocí Edisona a různých druhů stavebnicových systémů je možné postavit mnoho nejrůznějších různých věcí. V rámci této aktivity je vaším úkolem postavit pomocí Edisona nějaký předmět z kostek LEGA.

Co máte dělat

Vezměte si svého Edisona, kostky či bloky a popusťte uzdu své kreativitě a představivosti!

Zkuste přidat bloky na Edisona shora, zboku či na jeho kola. Ozdobte Edisona, jakkoli chcete!

Jakmile tento úkol dokončíte, popište nebo namalujte, jak Edison vypadal poté, co jste jej vybavili kostkami. Jak se vám s Edisonem stavělo?

Jméno_____

U1-1.2 Prozkoumejte programování čárových kódů

Stejně jako všichni roboti a všechny počítače potřebuje Edison ke svému fungování programy.

Co je to počítačový program?

Slovníček pojmů

Počítačový program je soubor instrukcí, které přikazují počítači provést konkrétní úkol.

Pojmem **žargon** označujeme speciální slova nebo výrazy používané lidmi v určité skupině, např. v určitém zaměstnání. Žargon je pro lidi mimo tuto profesi nebo skupinu často obtížně pochopitelný. Počítačové programování používá některá slova a fráze, které se vám nyní mohou jevit jako „žargon“, ale s touto novou slovní zásobou se brzy velmi dobře seznámíte. Tyto nové pojmy vás naučí Slovníček pojmů v každé lekci.

Nyní se vraťme k programování Edisona.

Použití čárových kódů k programování Edisona

Edison je vybaven několika přednahranými programy. K těmto programům se lze dostat a spustit je pomocí speciálních čárových kódů.

Proč je to tak?

Do Edisonova mikročipu je možné ukládat např. programy. Tyto programy jsou uloženy v paměti robota. Pokyn, který program má spustit, můžeme Edisonovi dát přejetím přes speciální čárové kódy.

Kdykoliv budete chtít využít některý ze speciálních čárových kódů, musíte provést následující čtyři kroky:

1. Umístěte Edisona tak, aby směřoval k čárovému kódu **zprava**.
2. Stiskněte **třikrát** tlačítko pro nahrávání (kulaté).
3. Počkejte, než Edison přejede a naskenuje čárový kód.
4. Ke spuštění programu stiskněte **jednou** tlačítko pro spuštění (trojúhelníkové).

Vyzkoušejme si nyní některé z Edisonových čárových kódů.

Jméno_____

Úkol 1: Pohyb ovládaný tleskáním

Tento program využívá Edisonova zvukového senzoru. Zvukový senzor dokáže detekovat hlasité zvuky, např. tleskání rukou. Program přikazuje Edisonovi, aby „poslouchal“, zda neuslyší tlesknutí.

Nechejte Edisona přečíst čárový kód.

Nezapomeňte

K naprogramování Edisona pomocí čárového kódu vždy postupujte dle následujících kroků:

1. Umístěte Edisona tak, aby směřoval k čárovému kódu **zprava**.
2. Stiskněte **třikrát** tlačítko pro nahrávání (kulaté).
3. Počkejte, než Edison přejede a naskenuje čárový kód.
4. Ke spuštění programu stiskněte **jednou** tlačítko pro spuštění.

Naskenujte čárový kód a položte Edisona na podlahu či stůl. Poté stiskněte tlačítko pro spuštění (trojúhelníkové). Po stisknutí tlačítka jednou tleskněte. Edison zatočí doprava.

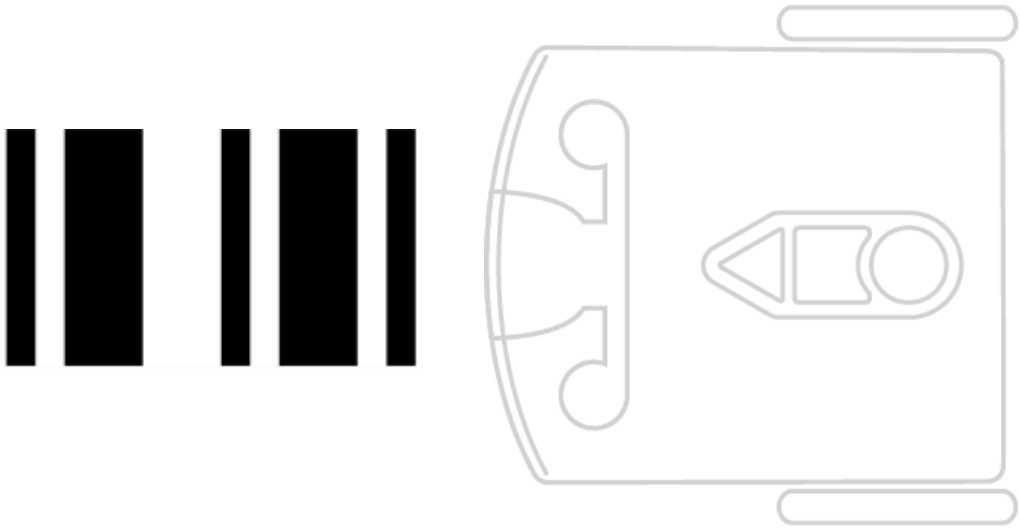
Nyní tleskněte dvakrát. Edison pojede dopředu.

Pokud není Edison schopen detekovat tleskání, zkuste namísto toho poklepat prstem na horní část robota blízko zvukového senzoru.

Úkol 2: Vyhýbání se překážkám

Tento program využívá Edisonova infračerveného senzoru k detekci překážek a vyhnutí se jim.

Nechejte Edisona přečíst čárový kód.



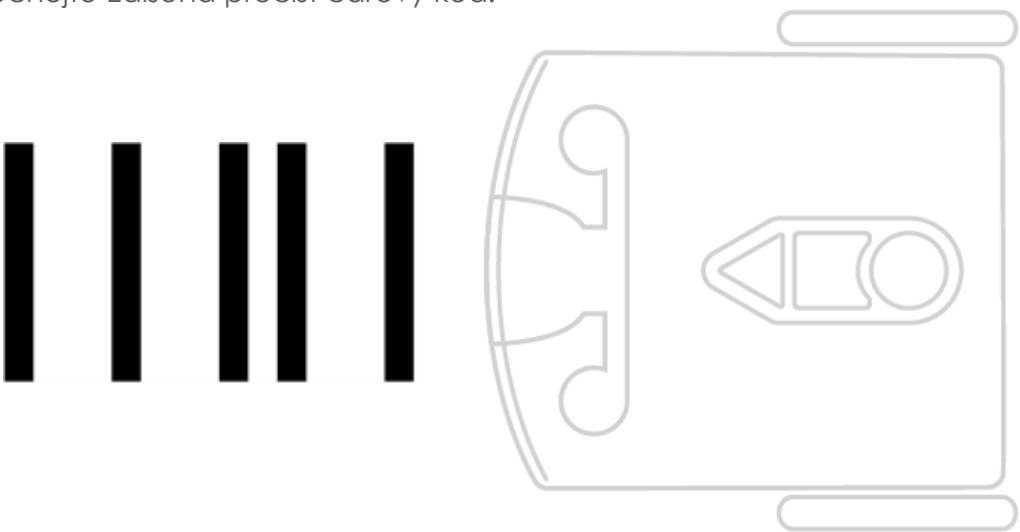
Než stisknete tlačítko pro spuštění (trojúhelníkové), umístěte Edisona na podlahu nebo stůl s překážkami. Vytvořte Edisonovi překážky např. tak, že kolem něj položíte různé předměty. Vyberte předměty, které jsou nejméně stejně velké jako Edison a nejsou průhledné. Můžete rovněž Edisonovi rukama vybudovat malé „stěny“.

Stiskněte tlačítko pro spuštění (trojúhelníkové). Sledujte, co se stane, když Edison narazí na překážku.

Úkol 3: Pohyb za světlem

Tento program používá Edisonův světelný senzor k detekci a sledování jasného světla. K jeho spuštění budete potřebovat svítilnu, baterku nebo nějaký jiný způsob, jak vytvořit jasné světlo.

Nechejte Edisona přečíst čárový kód.

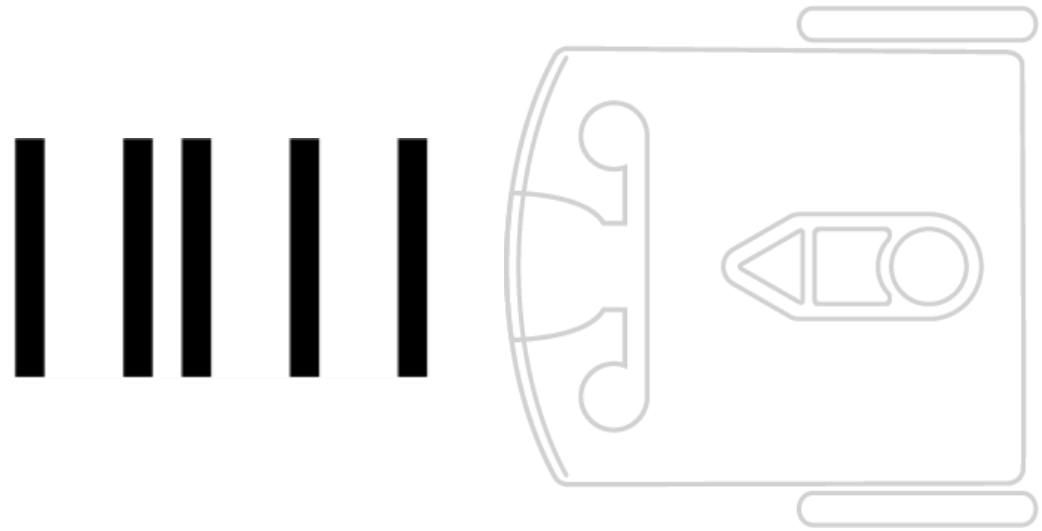


Položte Edisona na podlahu nebo stůl a mějte po ruce připravenu svítilnu. Stiskněte tlačítko pro spuštění (trojúhelníkové). Posviťte světlem na Edisona. Robot bude sledovat jasné světlo.

Úkol 4: Pohyb po čáře

Tento program využívá Edisonův senzor k detekci, sledování a pohybu podél tmavé čáry. Budete potřebovat tmavou čáru, po které Edison pojede. K vytvoření čáry pro Edisona použijte pracovní list U1-1, EdMat nebo vytvořte Edisonovi vlastní čáru.

Nechejte Edisona přečíst čárový kód.

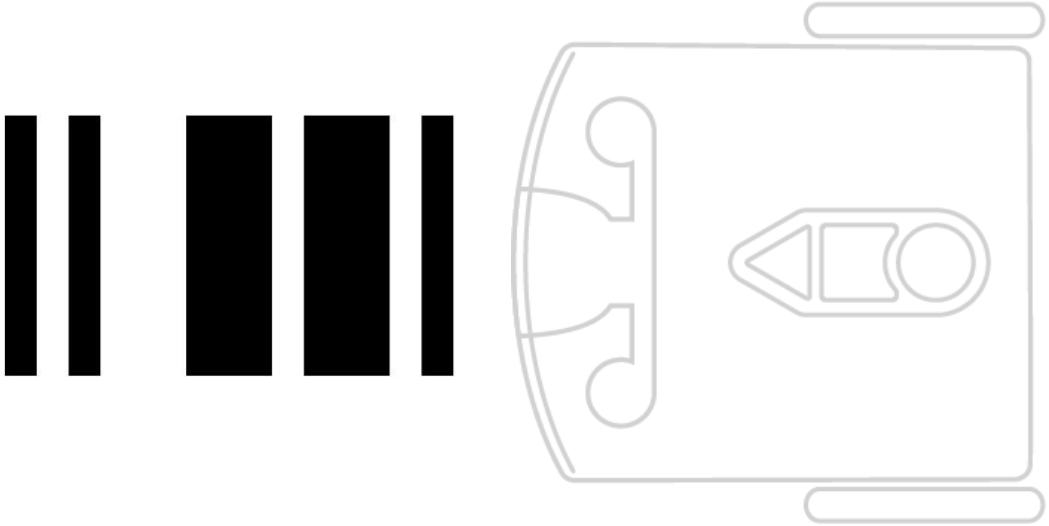


Připravte si pracovní list. Robota musíte spustit na bílém povrchu v blízkosti černé čáry. Položte Edisona vedle černé čáry, ale **nikoliv přímo na čáru**. Použijte tlačítko pro spuštění (trojúhelníkové). Edison najde čáru a pojede po ní.

Úkol 5: Pohyb ve vymezených hranicích

Tento program využívá Edisonův sledovací senzor k detekci a vyhýbání se tmavým povrchům. Budete potřebovat tvar s tmavým obrysem, do nějž Edisona „zachytíte“ jak do pasti. Použijte pracovní list U1-1, pracovní list U1-2, EdMat nebo si k zachycení Edisona vytvořte vlastní tvar.

Nechejte Edisona přečíst čárový kód.



Připravte si list pracovní list. Robota musíte spustit na bílém povrchu ohraničeném černými čarami. Edisona položte vedle černé čáry, **ale nikoliv přímo na čáru**. Stiskněte tlačítko pro spuštění (trojúhelníkové). Edison se bude pohybovat uvnitř prostoru ohraničeného tmavými čarami.

Jméno_____

U1-1.2a Změňte to: Zápas sumo

Jeden z Edisonových přednastavených programů je vlastně kombinací dvou jiných programů – pohybu ve vymezených hranicích a vyhýbání se překážkám.

Co tento kombinovaný program dělá? Umožňuje dvěma či více robotům Edison vzájemně zápasit v sumu!

Část programu pro detekci překážek pomáhá každému z robotů najít ostatní roboty. Program pro detekci čáry pak umožňuje Edisonovi najít čáru a vystrnadit jiného robota z ringu.

Budete potřebovat tvar s tmavým obrysem, který bude ringem pro sumo zápas bojujících Edisonů. Použijte pracovní list U1-2, EdMat nebo si vytvořte vlastní ring.

Na tomto úkolu spolupracujte s ostatními. Naskenujte čárový kód nejméně dvěma robotům Edison.

Nezapomeňte

K naprogramování Edisona pomocí čárového kódu vždy postupujte dle následujících kroků:

5. Umístěte Edisona tak, aby směřoval k čárovému kódu **zprava**.
6. Stiskněte **třikrát** tlačítko pro nahrávání (kulaté).
7. Počkejte, než Edison přejede a naskenuje čárový kód.
8. Ke spuštění programu stiskněte **jednou** tlačítko pro spuštění.

Připravte si ring pro zápas sumo. Pokud chcete, můžete označit jednotlivé roboty Edison štítky nebo připojením barevných Lego kostiček. Vložte všechny Edisony do ringu.

Stiskněte na všech robotech ve stejný okamžik tlačítko pro spuštění (trojúhelníkové).

Každý Edison začne pomalu projíždět vnitřkem ringu a hledat další roboty. Když některý z Edisonů detekuje jiného robota, zrychlí se, aby ho zasáhl a pokusil se ho vytlačit z ringu.

Edison, který zůstane v ringu jako poslední, vyhrává!

Jméno_____

U1-2.1 Prozkoumejte prostředí EdScratch

Jednou z nejlepších věcí na Edisonovi, je, že pro svého robota můžete vytvářet vlastní programy! Chcete-li napsat pro Edisona program, musíte použít speciální **software**.

Slovníček pojmů

Všechny počítače se skládají ze dvou hlavních částí: hardware a software.

Hardware jsou fyzické části počítače (nebo robota).

Software je sada programů a aplikací, které umožňují hardwaru, tj. např. počítači či robotovi, běžet.

Softwarem, který budeme pro Edisona používat, je **programovací jazyk** pro roboty.

Slovníček pojmů

Programovací jazyk je soubor pravidel a instrukcí používaných k psaní počítačových programů. EdScratch je programovací jazyk speciálně navržený pro programování robotů Edison.

Programovací jazyk, který budeme používat, se nazývá EdScratch. Pojďme se s ním seznámit trochu blíže.

Úkol 1: Vyzkoušejte EdScratch

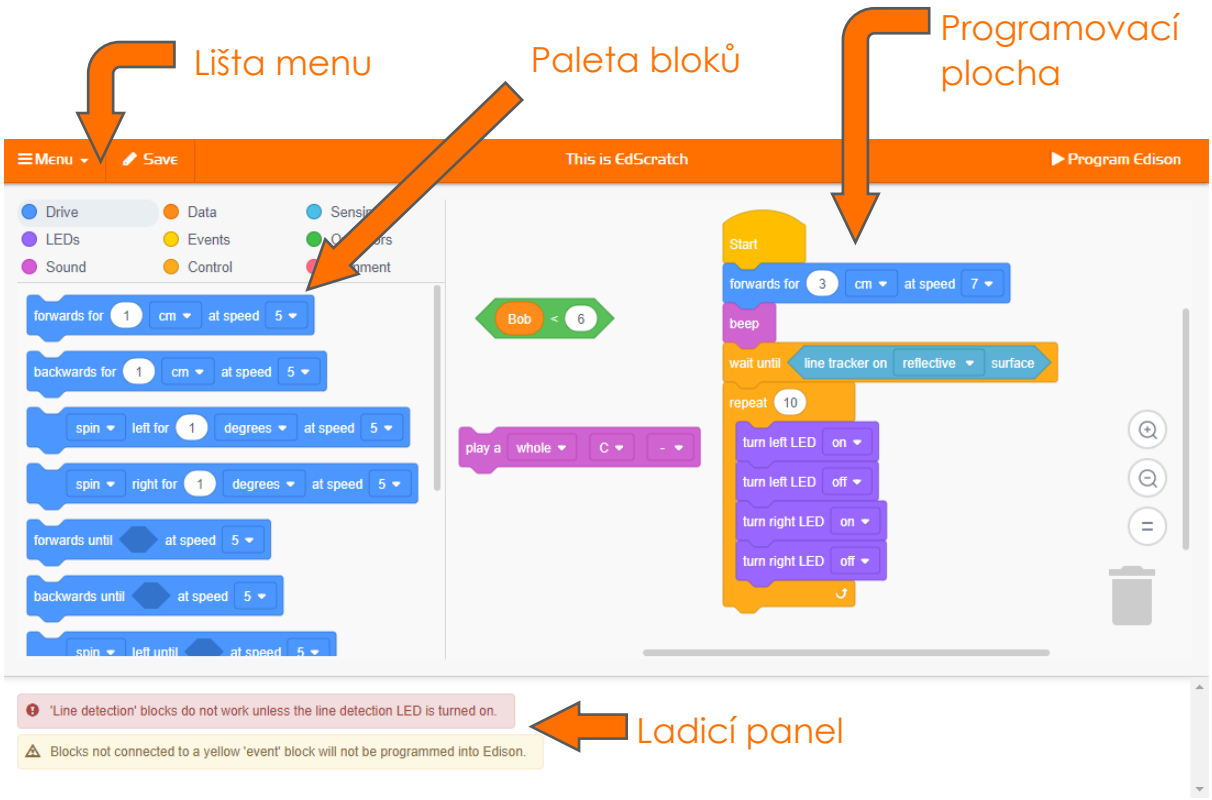
EdScratch je k dispozici online.

Použijte tento odkaz

Přejděte na www.edscratchapp.com

Kdykoliv budete chtít programovat Edisona v programovacím jazyce EdScratch, budete muset spustit aplikaci EdScratch.

Prostředí aplikace EdScratch vypadá takto:



Programovací prostředí EdScratch se skládá ze čtyř hlavních částí:

Paleta bloků

Všechny bloky, které můžete použít, najdete v paletě bloků. Chcete-li použít některý blok, vyberte jej z palety bloků a přetáhněte na programovací plochu.

Programovací plocha

Velká plocha, v níž můžete spojovat bloky do programů, se nazývá programovací plocha. Přetáhněte bloky z palety bloků na tuto plochu a použijte je ve svém programu.

Lišta menu

V liště menu jsou dostupné různé možnosti, např. „Save“ (Uložit) či „Load“ (Nahrát). V této liště naleznete i tlačítko „Program Edison“ (programování Edisona).

Ladicí panel

Pod paletou bloků a programovací plochou se nachází ladicí panel. V něm se objevují varovná hlášení.

Prohlédněte si EdScratch na svém počítači. Najděte každou ze čtyř hlavních částí prostředí EdScratch.

Úkol 2: Nahrajte a stáhněte testovací program

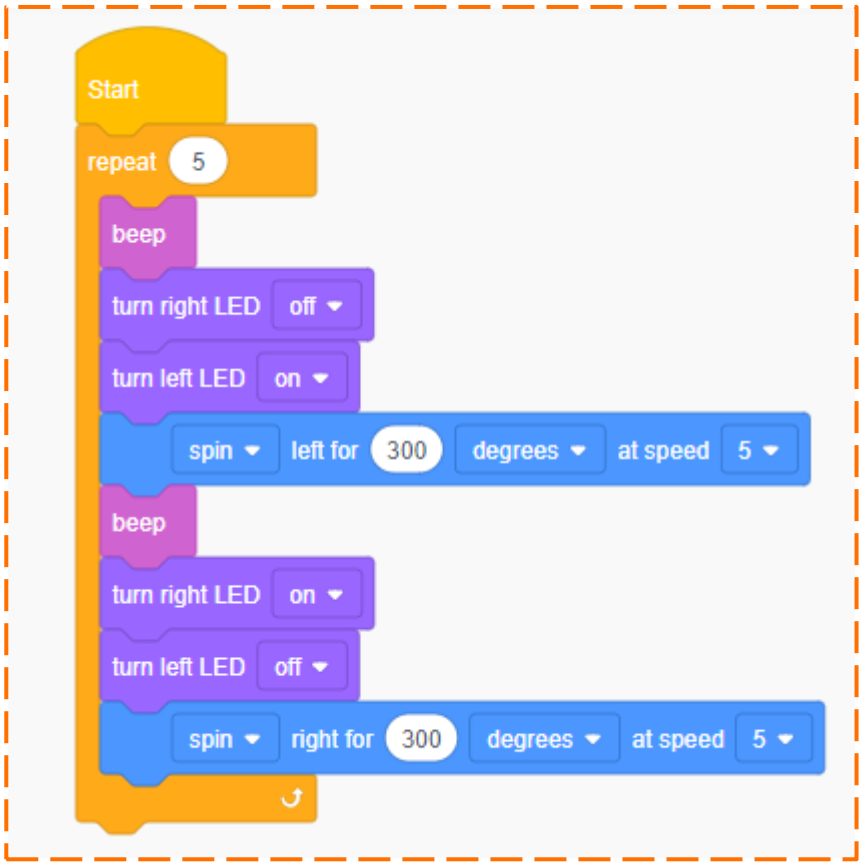
V EdScratchi je k dispozici několik hotových demo programů. Zkuste načíst a stáhnout demo program nazvaný **Test_program**.

Načtěte předváděcí program Test_program

Pro načtení předváděcího programu **Test_program** se řiďte následujícími kroky:

1. Přejděte v EdScratchi do lišty menu a vyberte rozbalovací nabídku. Vyhledejte a vyberte možnost „Load Demos“. Otevře se vyskakovací okno se všemi předváděcími programy.
2. Najděte a vyberte program nazvaný **Test_program**. Tento program se zobrazí na programovací ploše.

Test_program vypadá následovně:

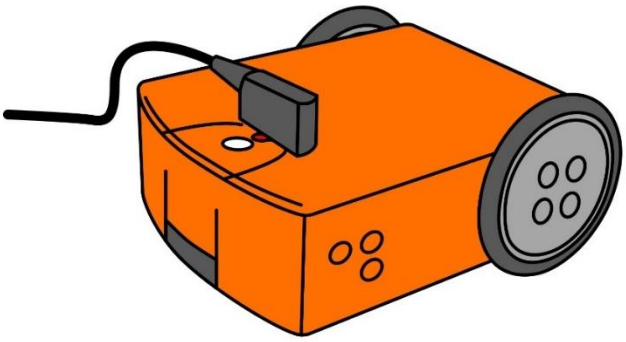


Jakmile se program zobrazí na programovací ploše, můžete si jej stáhnout do svého robota Edison.

Stažení programu Test_program do Edisona

Kdykoli chcete stáhnout program z EdScratche do Edisona, postupujte podle následujících kroků:

1. Připojte Edisona k počítači pomocí kabelu EdComm.
2. Ověřte, zda je hlasitost na počítači nastavena na maximum.
3. Stiskněte **jednou** tlačítko nahrávání (kulaté) na Edisonovi.
4. Přejděte do lišty menu v EdScratchi a klikněte na tlačítko Program Edison.
5. Otevře se vyskakovací okno. Jakmile je program připraven, objeví se ve spodní části vyskakovacího okna tlačítko Program Edison.
6. Klikněte na tlačítko Program Edison ve vyskakovacím okně.



Proč je to tak?

Edison nedokáže porozumět blokům v EdScratchi tak, jak jsou zobrazeny na obrazovce vašeho počítače. Bloky musí být před stažením změněny na formát, kterému Edison dokáže porozumět. To může chvíli trvat.

Proto může nějakou dobu trvat, než se ve vyskakovacím okně objeví tlačítko Program Edison.

Uslyšíte, jak se program stahuje do Edisona. Po skončení stahování se ozve pípnutí potvrzující úspěšné stažení. Dokud neuslyšíte toto pípnutí, Edisona neodpojujte!

Proč je to tak?

Edison vám potvrdí úspěšné stažení programu krátkým potvrzovacím pípnutím. Jde o stejný zvuk, který se ozve při spuštění Edisona.

Pokud se program nestáhne správně, Edison vydá jiný zvuk. Tento zvuk oznamuje selhání stahování. Pokud jej Edison vydá, zkuste zahájit stahování znovu.

Poté, co uslyšíte zvuk potvrzující úspěšné stažení programu, odpojte od Edisona programovací kabel. Pro spuštění programu stiskněte jednou tlačítko pro spuštění (trojúhelníkové).

Jméno_____

Vyzkoušejte si to!

Načtěte do EdScratche předváděcí program **Test_program**. Stáhněte jej do Edisona a spusťte. Poté zodpovězte následující otázky.

1. Ve které části prostředí EdScratch se nachází tlačítko Program Edison?

_____

_____

2. Kolik varovných hlášek se zobrazí v ladícím panelu po načtení programu **Test_program**?

_____

_____

3. Co robot provede po spuštění programu **Test_program**? Popište, co se děje.

_____

_____

_____

_____

_____

Jméno_____

U1-2.2 Prozkoumejte varovné hlášky

Některé programovací jazyky nabízejí speciální funkce usnadňující jejich používání. Jedním z příkladů takové funkce je Ladicí panel v EdScratchi.

Když píšeme program v EdScratchi, můžeme někde udělat chybu. V takovém případě se v Ladicím panelu objeví varovná hláška.

Nezapomeňte

Ladicí panel se nachází v EdScratchi pod paletou programovacích bloků a programovací plochou.

Existují dva druhy varovných hlášek: žluté a červené varovné hlášky.

Proč je to tak?

Žluté varovné hlášky slouží k upozornění. Těmito hláškami vám EdScratch sděluje: „Pozor! Tohle možná nebude fungovat tak, jak byste si představovali." Programy se žlutými varovnými hláškami v ladicím panelu však přesto můžete stáhnout.

Červené varovné hlášky slouží jako „stopka". Těmito hláškami EdScratch upozorňuje: „Je mi líto! Tomuto programu Edison nebude rozumět." Pokud se v ladícím panelu objeví nějaká červená varovná hláška, nebudete moci daný program do Edisona vůbec stáhnout.

Kdykoli píšete programy pro Edisona, je rozumné před stažením programu do robota zkontrolovat ladící panel. Varovné hlášky vám mohou pomoci program opravit!

Vyzkoušejte si to!

Najděte v EdScratchi předváděcí program **Warning_messages_demo** a načtěte jej.

Nezapomeňte

EdScratch najdete na adrese www.edscratchapp.com

Přejděte v EdScratchi do lišty menu a vyberte rozevírací nabídku. Vyhledejte a vyberte možnost **Load Demos**. Otevře se vyskakovací okno se všemi předváděcími programy. Najděte a načtěte program **Warning_messages_demo**.

Jméno_____

Po načtení programu do EdScratche zodpovězte následující otázky

1. Zkuste tento program stáhnout do robota Edison. Co se stane? Funguje to? Proč ano nebo proč ne?

_____

_____

_____

2. Přečtěte si červenou varovnou hlášku v ladícím panelu. Podívejte se na program. Dokážete tento problém napravit? Popište, jaké kroky jste podnikli k nápravě červené varovné hlášky.

_____

_____

_____

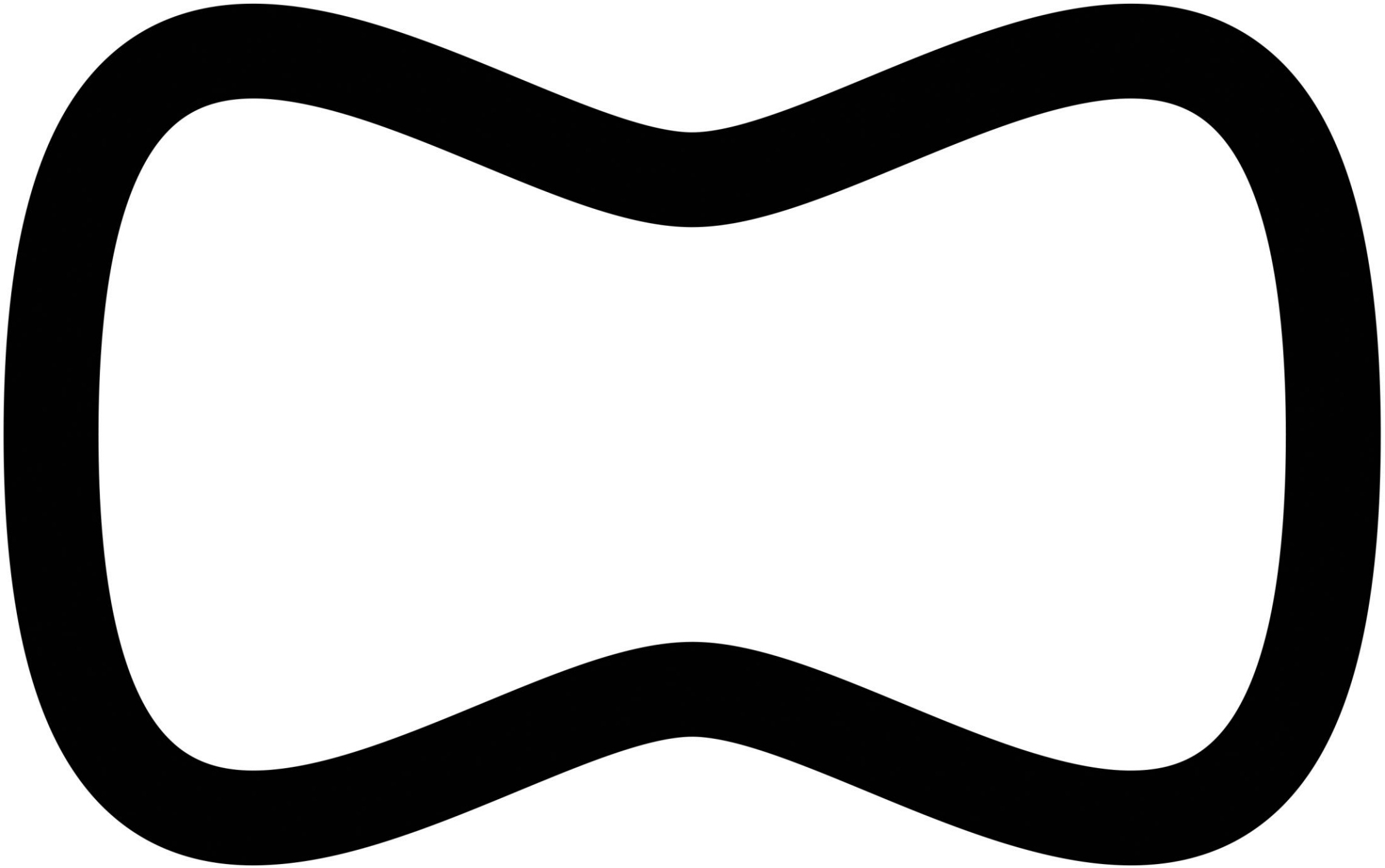
3. Přečtěte si žlutou varovnou hlášku v ladícím panelu. Pokud stáhnete program do Edisona i se žlutými varovnými hláškami, které bloky do něj nebudou naprogramovány?

_____

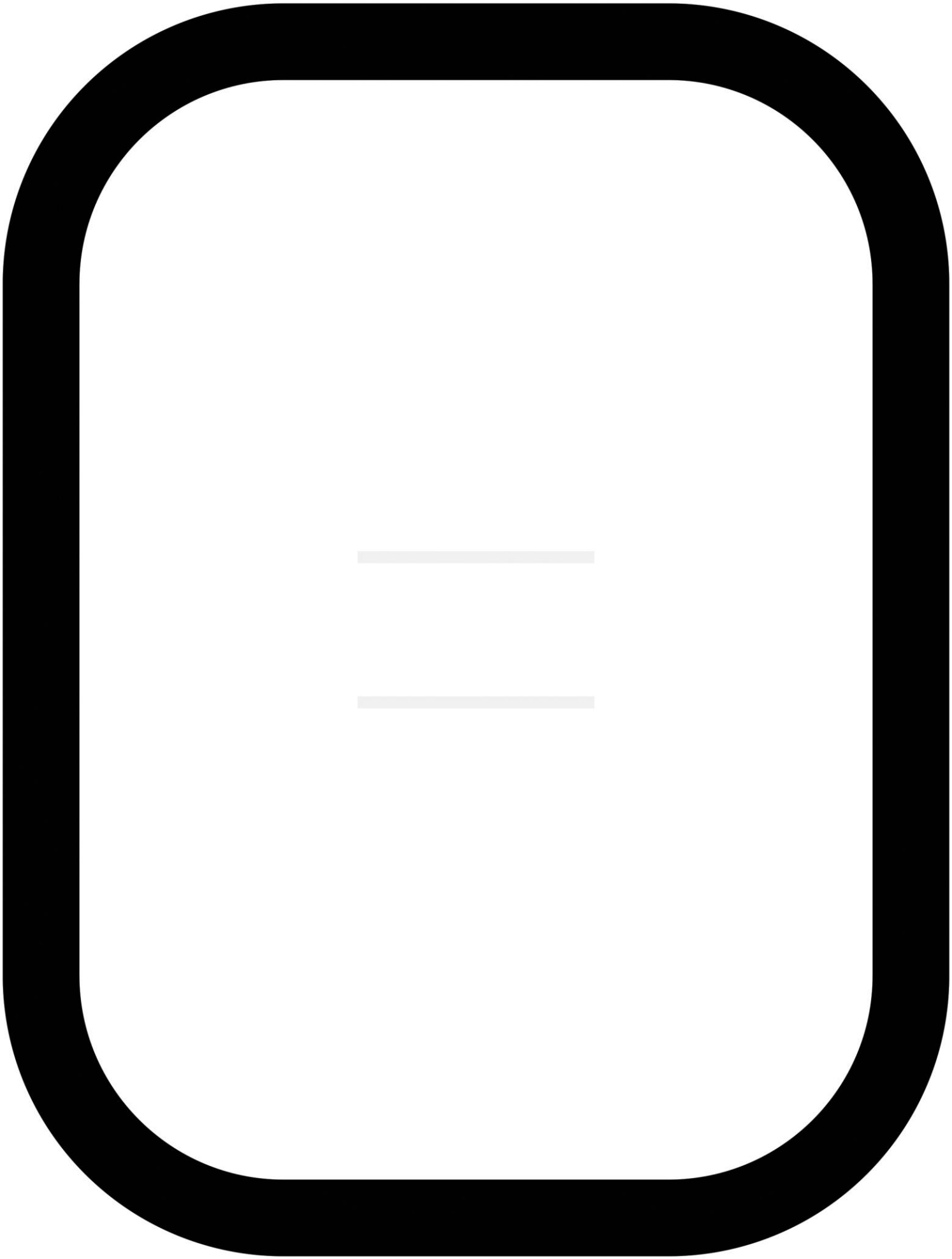
_____

_____

Pracovní list U1-1: Hranice pro sledování čáry



Pracovní list U1-2: Ring pro zápas sumo



Pracovní list U1-3: Dálkový ovladač TV

Pohyb vpřed / kód 0

Pohyb vzad / kód 1

Otočení doprava na místě / kód 4

Otočení doleva na místě / kód 5

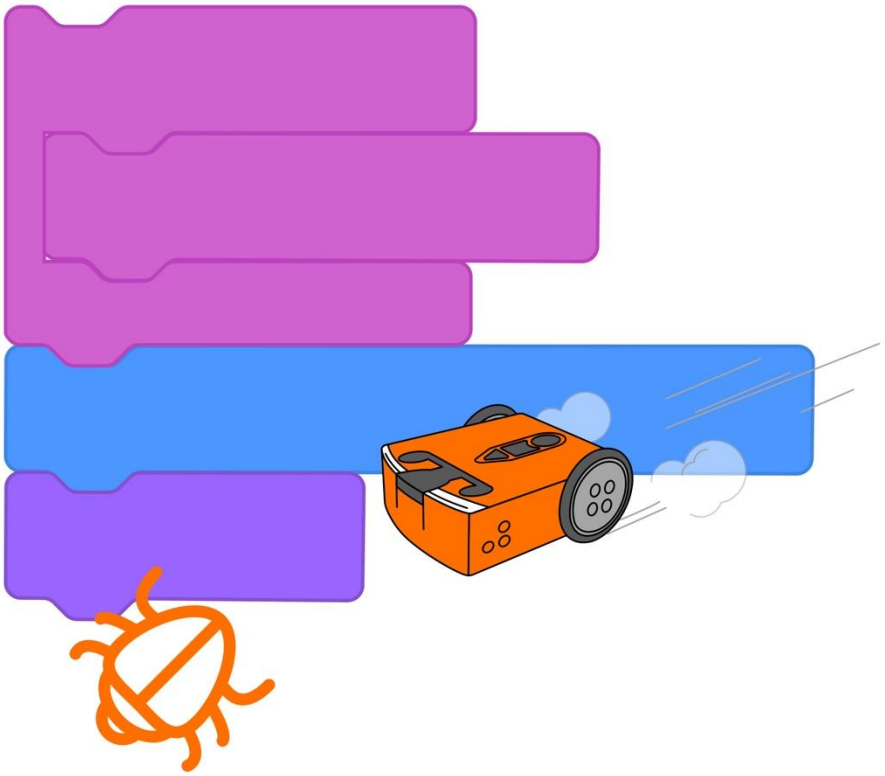
Zatočení doprava / kód 2

Zatočení doleva / kód 3

Přehrát pípnutí / kód 6

Přehrát melodii / kód 7

Lekce 2: Rozpohybujte to!



Edison

Jméno_____

U2-1.1 Prozkoumejte, jak počítače „přemýšlejí"

Představte si počítač. Nyní si představte člověka. Kdo je podle vás chytřejší?

Tato otázka je trochu záludná. Věřte nebo ne, většina počítačů bez lidské pomoci nedokáže udělat nic.

Proč je to tak?

Jak počítače, tak lidé, se mohou řídit pokyny, ale jen lidé dokáží přemýšlet sami za sebe. Umíme se poučit a změnit to, co děláme, na základě nových znalostí.

Většina počítačů to však nedokáže. Robot Edison například nedokáže myslet sám za sebe. Umí se pouze řídit pokyny. Od koho tyto pokyny pocházejí? Od člověka, jako jste vy!

Lidé dávají počítačům pokyny pomocí počítačových programů.

Abychom pro našeho Edisona nebo počítač vytvořili dobrý počítačový program, musíme tento program napsat způsobem, kterému bude počítač rozumět. Proto musíme o věcech přemýšlet tak, jako bychom sami byli počítačem.

Nezapomeňte

Počítačový program je soubor pokynů, které říkají počítači, že má splnit konkrétní úkol.

Tento druh myšlení se nazývá **algoritmické myšlení.**

Slovníček pojmů

Algoritmické myšlení znamená, že o problému nebo úkolu přemýšlíte podobně jako počítač. Je to způsob, jak logicky řešit problémy jejich rozdělením na menší kousky, nalezením vzorců a následným vytvořením postupného řešení s využitím informací.

Jinými slovy, algoritmické myšlení je způsob plánování, řešení problémů a analýzy informací stejným způsobem jako počítač.

Kdykoliv budete chtít napsat program pro Edisona, budete muset použít algoritmické myšlení, abyste přišli na to, co máte dělat. Tím, že se naučíte přemýšlet tak, aby vám Edison rozuměl, budete schopni dát robotovi pokyny, díky nimž vykoná to, co chcete.

Jednou z nejdůležitějších věcí při zadávání pokynů Edisonovi je pořadí, v němž pokyny dáváte.

Jméno_____

Důležitost postupu krok za krokem

Počítače, včetně robotů Edison, se velmi dobře řídí pokyny, které jim dáváme formou počítačových programů. Ve skutečnosti se robot Edison bude řídit pokyny v programu *přesně* tak, jak jsou napsány. Proto je jednou z nejdůležitějších součástí algoritmického myšlení použití **sekvence**.

Slovníček pojmů

Sekvence znamená postup krok za krokem.

Představte si, že chcete upéct dort. Vyhledáte si recept v kuchařce. Při přípravě dortu postupujete krok za krokem. Přesně to je sekvence!

Kdykoliv budete psát program pro Edisona, budete muset používat sekvenci přesně tímto způsobem. Musíte Edisonovi přesně říct, co má dělat, a v přesném pořadí, v němž chcete, aby tyto kroky vykonal.

Úkol 1: Krok za krokem

Když vám učitel řekne, abyste šli ke dveřím, jaké úkony musíte udělat, abyste se tam dostali? Pravděpodobně nepřemýšlíte o tom, kolik kroků musíte udělat. Prostě to uděláte! Pokud vám stojí v cestě stůl, jednoduše se otočíte a obejdete jej.

Robot takto nefunguje. Chcete-li dostat svého robota ke dveřím, musíte mu dát velmi pečlivé pokyny s podrobným vysvětlením jednotlivých kroků, jednoho po druhém. Jinými slovy, musíte robotovi sdělit každou akci, kterou má provést v rámci dané sekvence.

Naučit se, jak dělat věci sekvenčně, vyžaduje trochu praxe. Lidé jsou navyklí věci jednoduše udělat a obvykle nepřemýšlejí o tom, v jakých krocích každou jednotlivou aktivitu provádějí.

Zkuste se řídit pokyny krok za krokem, ať víte, jaké to je. Odpovězte na následující otázky v pracovním sešitu U2-1.

1. Položte robota na kornout zmrzliny a nasměrujte ho k srdci. Odbočte vpravo. Postupte vpřed o 2 čtverce. Kde jste?

   _____

2. Položte robota na pandu a nasměrujte ho ke kolu. Posuňte se o 1 čtverec zpět. Odbočte vlevo. Posuňte se vpřed o 2 čtverce. Odbočte vpravo. Posuňte se vpřed o 1 čtverec. Kde jste?

   _____

3. Položte robota na hvězdu a nasměrujte ho ke kočce. Odbočte vlevo. Znovu zahněte doleva. Posuňte se o 2 čtverce zpět. Odbočte vpravo. Posuňte se vpřed o 1 čtverec. Odbočte vpravo. Posuňte se vpřed o 1 čtverec. Odbočte vlevo. Posuňte se o 2 čtverce zpět. Kde jste?

   _____

Jméno_____

Úkol 2: Zadávejte pokyny krok za krokem

Procvičme si zadávání přesných pokynů a popis jednotlivých položek krok za krokem. Odpovězte na následující otázky v pracovním sešitu U2-1.

Nápověda!

K zapsání odpovědí použijte tyto příkazy:

move forwards	*move backwards*	*turn left*	*turn right*
(posunout vpřed)	*(posunout vzad)*	*(zatočit vlevo)*	*(zatočit vpravo)*

4. Napište pokyny pro tuto akci: *Začni na duze, směrem k psovi. Skonči na ptákovi.*

_____

_____

_____

5. Napište pokyny pro tuto akci: *Začni na duze, směrem k psovi. VYHNI se psovi. VYHNI se kočce. Skonči na ptákovi.*

_____

_____

_____

6. Napište pokyny pro tuto akci: *Začni na diamantu, směrem k plážovému balónu.* NEPOUŽÍVEJTE příkaz 'move forwards' (posunout vpřed). *Skonči na plážovém balónu.*

_____

_____

_____

Jméno_____

U2-1.1b Změňte to: lidští roboti

Existuje spousta známých a zdánlivě velmi snadných věcí a činností, které ve skutečnosti sestávají z mnoha jednotlivých kroků. Roboti nedokáží příliš dobře spojovat jednotlivé kroky. Místo toho potřebují jasné pokyny, které jim krok za krokem vytýčí každý jednotlivý pohyb. Proto je důležité vysvětlit pomocí jasných pokynů seřazených v **sekvenci**, co se má odehrát - jedině tak robot vykoná to, co po něm chcete.

Slovníček pojmů

Sekvence znamená postup krok za krokem.

Úkol 1: Napište pokyny krok za krokem

S přípravou na tento úkol vám pomůže váš učitel.

1. Co musíte udělat, aby se lidský robot dostal k cíli? Zapište si postupně každý krok.

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Úkol 2: Řiďte se pokyny

S přípravou na tento úkol vám pomůže váš učitel. Ujistěte se, že se řídíte pokyny PŘESNĚ tak, jak byly napsány. Nepřidávejte žádné další kroky!

2. Dosáhl lidský robot cíle? Museli jste něco změnit ve svých pokynech, aby fungovaly?

_____

_____

Jméno_____

U2-1.2 Prozkoumejte postup krok za krokem v EdScratchi

Když píšete program pro Edisona v EdScratchi, píšete pokyny, které řeknou robotovi, co má dělat a v jakém pořadí má každý krok udělat. Každý blok v EdScratchi je jednou akcí, k jejímuž provedení dáváte robotovi pokyn.

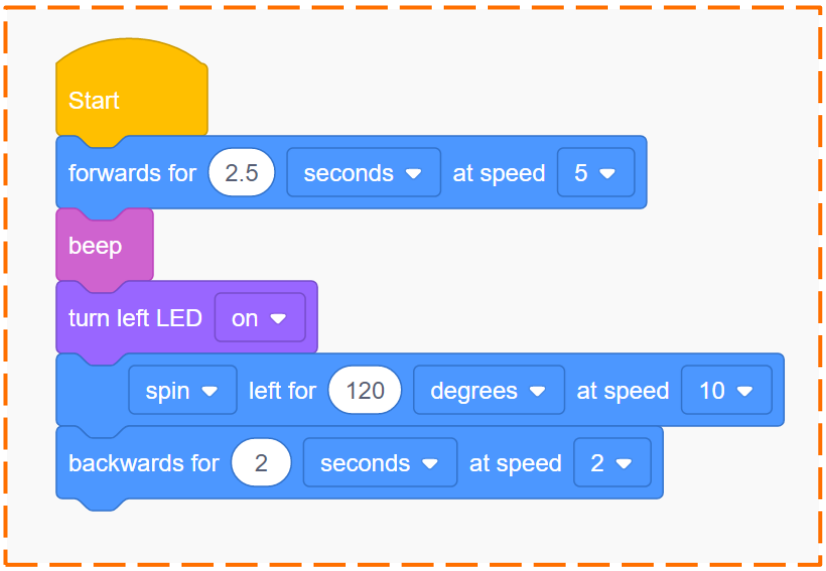
Pořadí, v němž spojíte bloky ve svém programu, říká robotovi, v jakém pořadí má provést každou akci. Edison bude provádět kroky postupně, jeden po druhém, počínaje horním blokem.

Nezapomeňte

Sekvence znamená postup krok za krokem.

Úkol 1: Co Edison udělá?

Prohlédněte si tento program v EdScratchi:



Všechny programy v EdScratchi musí začít žlutým blokem Start. Ten robotovi sděluje, že program začíná prvním blokem pod blokem Start.

Přečtěte si každý řádek blok po bloku, počínaje horním blokem umístěným pod blokem Start.

1. Pokud stáhnete tento program do Edisona, co robot udělá? Nezapomeňte napsat odpověď ve správném pořadí.

_____

_____

_____

_____

_____

Jméno_____

Úkol 2: Napište program

Přejděte do aplikace EdScratch ve svém počítači a napište program podle obrázku. Najděte každý blok v paletě bloků a přetáhněte jej na programovací plochu.

Nápověda!

Čísla v bloku můžete změnit kliknutím na číslo a zadáním pomocí klávesnice.

Rozbalovací položku v bloku můžete změnit kliknutím na šipku dolů a výběrem požadované možnosti.

Ujistěte se, že jste ve svém programu umístili všechny bloky ve správném pořadí.

Po napsání programu si jej stáhněte do svého robota Edison. Spusťte program a sledujte, co Edison dělá.

Nyní se podívejte, co jste napsali v odpovědi v úkolu č. 1, že podle vás robot udělá. Porovnejte, co Edison při spuštění programu dělá, se svou odpovědí. Dělá robot to, co jste předpověděli? Pokud ne, co je jinak? Zkontrolujte svůj program a své odpovědi a zkuste zjistit, zda dokážete identifikovat a napravit rozdíly.

Úkol 3: Změňte sekvenci

Zkuste změnit sekvenci svého programu. Změňte pořadí nejméně tří bloků v programu. Nepřidávejte žádné další bloky ani neměňte žádnou z možností uvnitř bloku - změňte pouze pořadí.

Stáhněte nový program do Edisona a spusťte jej.

2. Které bloky jste přesunuli, abyste změnili sekvenci programu? Zapište si svůj nový program.

_____

_____

_____

_____

_____

Jméno_____

U2-1.3 Prozkoumejte, jak Edison jezdí

Jednou ze skupin programovacích bloků v paletě bloků v EdScratchi je kategorie Drive (Jízda). Všechny bloky v této kategorii souvisejí s používáním Edisonových motorů. Jednou z možností je řídit robota jako auto.

Řidič není třeba! Jen programátor

Jak můžete s Edisonem 'jezdit'? **Naprogramováním kódu** do robota!

Slovníček pojmů

Programy jsou soubory pokynů, které vytvoříte pro svůj počítač nebo robota Edison. Obsahu programu se někdy říká **kód**.

Lidé často používají pojmy **kód** a **program** zaměnitelně. Můžete např. říci, že „napíšete program pro Edisona", i že „napíšete kód pro Edisona". V obou případech to znamená „napsat příkazy, které dají Edisonovi pokyny pomocí programovacího jazyka, kterému rozumí."

Když použijete EdScratch k vytvoření souboru pokynů pro Edison, můžete říci, že **programujete** nebo že **kódujete** nebo obojí. Což z vás dělá jak **programátora**, tak **kodéra**!

Vyzkoušejte si programování Edisona pomocí bloků pro jízdu.

Úkol 1: Jízda po rovné trati

V tomto úkolu je vaším úkolem přimět Edisona, aby jel po rovné trati. Použijte pracovní list U2-2. Potřebujete napsat program, který Edisonovi umožní jet po trati. Spusťte Edisona na okraji a nechejte jej zastavit poté, co překročí „cílovou" čáru.

Přejděte do aplikace EdScratch ve svém počítači. Prohlédněte si bloky v kategorii Drive (Jízda). Které bloky budete potřebovat k napsání programu? Stáhněte si program do Edisona a otestujte jej na pracovním listu. Fungoval?

Nezapomeňte

Čísla v bloku můžete změnit kliknutím na číslo a zadáním pomocí klávesnice.

Rozbalovací položku v bloku můžete změnit kliknutím na šipku dolů a výběrem požadované možnosti.

1. Řešení tohoto úkolu můžete nakódovat pomocí jediného bloku! Který blok bude fungovat? Vyplňte níže uvedený blok podle bloku, který jste použili ve svém úspěšně fungujícím programu.

Úkol 2: Jízda bludištěm

V rámci tohoto úkolu musíte Edisona naučit, aby projel bludištěm. Podívejte se na bludiště na pracovním listu U2-3. Přemýšlejte o jednotlivých krocích, které bude Edison muset udělat, aby projel bludištěm. Nezapomeňte vzít v úvahu také pořadí těchto kroků!

2. Jaké kroky podle vás bude Edison muset udělat, aby projel celým bludištěm? Napište plán, jak zajistit, aby Edison projel bludištěm.

_____

_____

_____

_____

Pomocí tohoto plánu napište v EdScratchi program, který umožní Edisonovi projet bludištěm. K projetí celého bludiště budete potřebovat použít několik různých bloků. Budete rovněž muset přijít na to, jaké **vstupní parametry** v každém bloku použít.

Slovníček pojmů

Údaje v bloku, které můžete změnit, např. čísla a volby v rozbalovacích menu, se nazývají **vstupní parametry**.

Spusťte Edisona na obrysu bludiště a nechejte jej zastavit poté, co překročí „cílovou“ čáru. Ujistěte se. že Edison zůstane v celém bludišti mezi čarami – bez podvádění!

Nápověda!

Váš program možná napoprvé nebude fungovat – to nevadí! Součástí psaní kódů je i experimentování a řešení problémů. Pokud váš program nefunguje, zamyslete se, jaké kroky musí Edison provést k projetí bludiště.

Chybí vám v programu některé kroky? Jsou některé kroky ve špatném pořadí?

Můžete také zkusit změnit některé vstupní parametry. Někdy i malá změna vstupního parametru způsobí velký rozdíl!

Jméno_____

U2-1.3a Další výzva: Šílenství v bludišti

Přiměli jste Edisona, aby úspěšně projel bludiště na pracovním listu U2-3? Zkuste i další bludiště! Nezapomeňte, že Edison musí zůstat mezi čarami po celou cestu bludištěm – žádné podvádění!

Zrcadlová trasa

Projeďte bludiště od „cílové“ čáry po startovní bod.

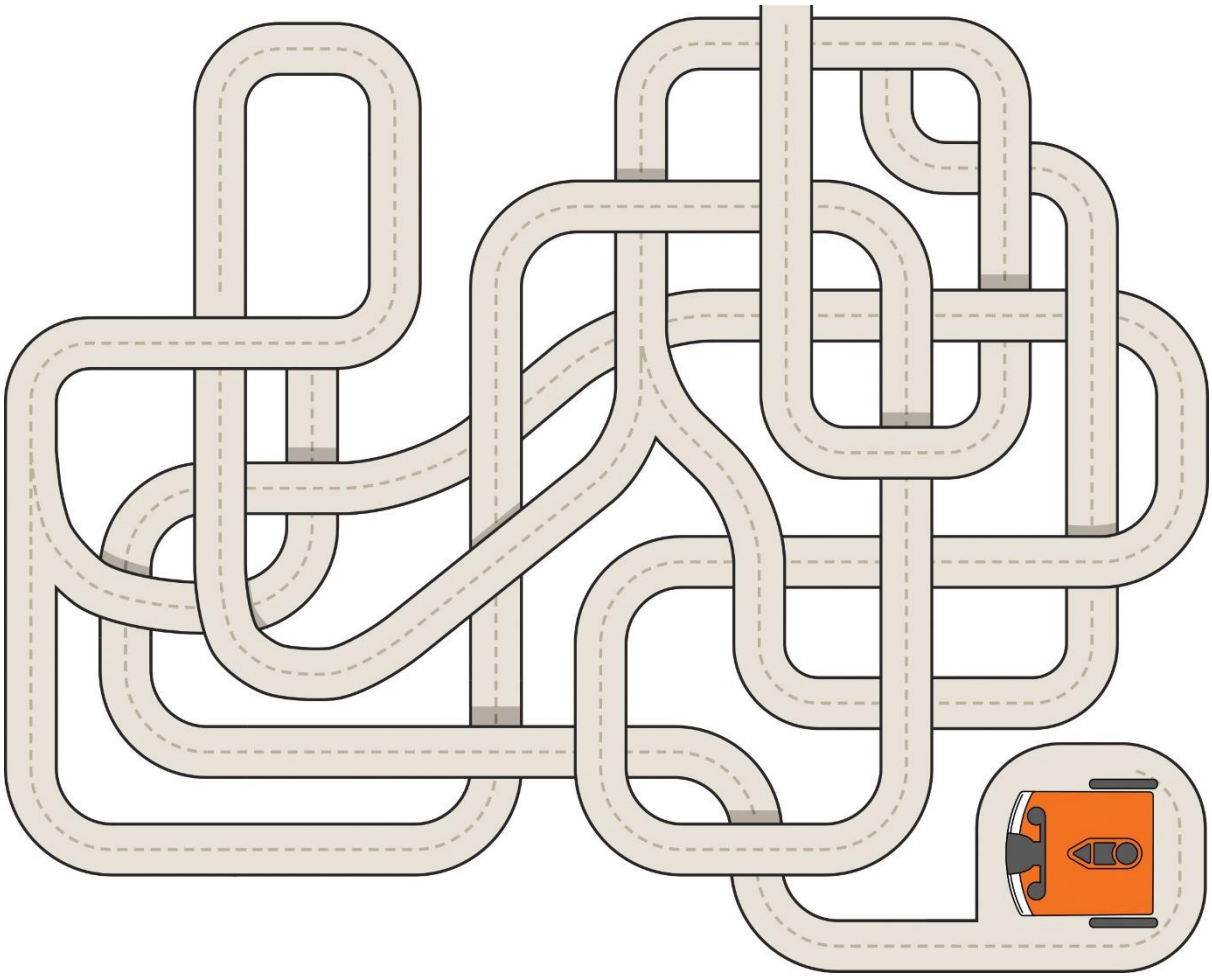
Jízda pozpátku

Projeďte bludiště pozpátku, od začátku až do cíle.

Vytvořte si vlastní bludiště

Vytvořte Edisonovi vlastní bludiště a poté napište v EdScratchi program, který mu umožní toto bludiště projet.

Až vyřešíte své bludiště, vyměňte se s partnerem. Zatímco on bude řešit, jak přimět Edisona projet vaším bludištěm, vy budete muset přijít na to, jak dostat robota celým jeho bludištěm!



Jméno_____

U2-2.1 Prozkoumejte Edisonovy výstupy

Co dělají počítače? Počítače **zpracovávají** informace. To znamená, že počítače odněkud berou informace a něco s těmito informacemi dělají. Například můžete dát počítači dvě čísla a zadat mu úkol, aby je sečetl. Počítač pak může tato čísla sečíst a zobrazit výsledek.

Takový cyklus, do kterého informace vstupují, počítač s nimi provede nějaký úkon a vytvoří výsledek, se nazývá **cyklus input-process-output (vstup-zpracování-výstup)**.

Slovníček pojmů

Vstupy jsou informace a pokyny, které dáváte počítači.

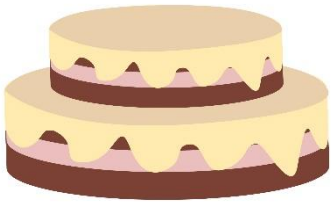
Zpracování označuje akci, kterou počítač provede pomocí počítačového programu s informacemi a pokyny. Někdy se tomu říká „spuštění“ programu.

Výstupy jsou výsledky, které získáte z počítače. Výstupem je to, co počítač zobrazí nebo co robot udělá na základě zadaných informací a pokynů.

Tomuto procesu, při němž do počítače přicházejí **vstupy**, počítač informace zpracuje v rámci **zpracování** a vygeneruje určitý druh **výstupu**, říkáme **cyklus input-process-output (vstup-zpracování-výstup)**.

Cyklus vstup-zpracování-výstup se nepoužívá jen v počítačích. Můžete se s ním často setkat i v každodenním životě.

Dobrým příkladem je pečení dortu. **Vstupem** jsou přísady vložené do formy, kterou následně dáte do trouby. V troubě dojde ke **zpracování** – pečení. Po nějaké době je hotov **výstup** – dort!



Vstupy, výstupy a Edison

Když píšete program pro svého Edisona, říkáte robotovi, co má dělat tím, že mu poskytujete vstupy. Edisonův mikročip pak tyto informace zpracuje a sdělí robotovi, jaký má být výstup.

Edison má tři hlavní druhy výstupů: výstupy využívající motory, výstupy využívající LED a výstupy využívající zvuky. V EdScratchi jsou bloky vztahující se k hlavním výstupům Edisona rozčleněny do tří různých kategorií: Drive (Jízda), LEDs (LED) a Sound (Zvuk).

Jméno_____

1. Prohlédněte si bloky v kategoriích Drive, LEDs a Sound v EdScratchi. Která kategorie obsahuje bloky kódu, které byste museli vložit do programu, aby Edison vygeneroval níže uvedené výstupy? Spojte požadovaný výstup s kategorií, v níž najdete potřebné bloky.

Výstup		Kategorie
	Vypnutí světel ☐	● Drive
	Zahrání tónu ☐	● LEDs
	Zatočení doprava ☐	● Sound

Úkol 1: Blikněte a pípněte

Edison je vybaven dvěma červenými LEDkami. Tyto LEDky začnou po spuštění Edisona blikat.

Edison má také speciální zařízení, které můžete vidět na levé straně kulatého tlačítka na horní straně robota. Jedná se o bzučák a zvukový senzor v jednom. Dokáže rozpoznat zvuk, ale také jej vytvářet!

V rámci tohoto úkolu je třeba napsat program, v němž bude Edison používat jak výstupy LED, tak bzučáku. Napište program v EdScratchi **pomocí osmi bloků**, které přikáží Edisonovi, aby postupně vykonal tyto kroky:

- ☐ zapnout levou LED
- ☐ pípnout
- ☐ vypnout levou LED
- ☐ pípnout
- ☐ zapnout pravou LED
- ☐ pípnout
- ☐ vypnout pravou LED
- ☐ pípnout

Stáhněte program do Edisona a vyzkoušejte jej.

2. Fungoval program v souladu s vaším očekáváním? Viděli jste, jak robot plní každý jednotlivý krok programu?

_____

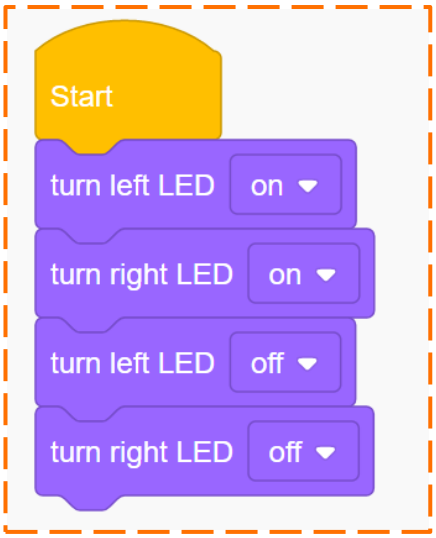
_____

_____

Úkol 2: Naučte Edisona blikat

Některé z Edisonových výstupů, jako např. zapnutí či vypnutí LED, se odehrávají velmi rychle. Mohou být tak rychlé, že je velmi těžké je zahlédnout.

Zkuste napsat v EdScratchi následující program:



Stáhněte jej do robota a spusťte. Vidíte, jak Edison bliká?

Protože robot v pohotovostním režimu bliká LEDkami, je tento program velmi obtížně pozorovatelný.

Proč je to tak?

Edison uskutečňuje bloky EdScratch jednotlivě, robot však dokáže každý blok zpracovat velmi rychle. Počítače mohou zpracovávat informace velmi rychle – to je jedna z věcí, kvůli nimž jsou tak užitečné!

Pokud chcete, aby Edison udělal přestávku mezi jednotlivými bloky, musíte mu to říct.

Jednou z kategorií bloků v EdScratchi je Control (Ovládání). Bloky v kategorii Control vám umožňují ovládat tok programu. Jedním z bloků v této kategorii je blok wait (počkat):

Tento blok řekne Edisonovi, aby před přechodem na další blok vyčkal po vámi určený časový interval. Zkuste tento blok použít v programu.

Pomocí tohoto nového ovládacího bloku upravte dříve vytvořený „blikací“ program tak, aby fungoval lépe. Chcete vytvořit program, v němž všichni snadno uvidí, že Edison bliká.

Jméno_____

Experimentuje s použitím alespoň jednoho bloku wait (čekat). Vyzkoušejte využití bloků wait na různých místech programu a zjistěte, kde fungují nejlépe.

3. Jak vypadá váš program? Které bloky používá, v jakém pořadí? Napište svůj program níže. Nezapomeňte uvést použité vstupní parametry.

Mini výzva!

Co se přesně stane, když někdo zamrká? Otevře oči a pak…

Prohlédněte si svůj blikací program. Dokážete jej upravit tak, aby to vypadalo, že robot mrká?

Jméno_____

U2-2.1a Další výzva: Projeďte bezpečně bludiště

Umíte napsat pro Edisona programy, které mu přikazují používat různé druhy výstupů.

Nezapomeňte

Edison má tři hlavní druhy výstupů: výstupy využívající motory, výstupy využívající LED a výstupy využívající zvuky. V EdScratchi jsou bloky vztahující se k hlavním výstupům Edisona rozčleněny do tří různých katerogorií: Drive (Jízda), LEDs (LED) a Sound (Zvuk).

V rámci této aktivity bude vaším úkolem napsat program, který přikáže Edisonovi projet bludiště na pracovním listu U2-3. Tentokrát však musí jet Edison velmi opatrně. Řidiči v silničním provozu upozorňují ostatní řidiče pomocí směrových světel a klaksonů. Edison to dokáže taky!

Projeďte bludiště od okraje směrem k cílové čáře. Program by měl skončit poté, co Edison přejede cílovou čáru.

Program by měl Edisonovi přikázat před každým zatočením v bludišti zastavit a udat směr jízdy zablikáním LEDkami. Ujistěte se, že dá Edison signál ostatním řidičům!

Nápověda!

Pokud se chystáte odbočit doleva, kterou LEDku použijete k vyslání signálu? A co při odbočení doprava?

Váš program by rovněž měl alespoň jednou využít blok „beep“ (pípnutí).

Nápověda!

Použijte blok „beep“ jako klakson v autě. Na kterém místě v programu dává smysl, aby Edison zapípal?

Jméno_____

U2-2.2 Prozkoumejte vstupní parametry

Každý blok v EdScratchi obsahuje vstupy, které Edisonovi říkají, co chcete, aby udělal. Pravděpodobně jste si již všimli, že mnohé z bloků mají také **vstupní parametry**.

Slovníček pojmů

Údaje v bloku, které můžete změnit, např. čísla a volby v rozbalovacích menu, se nazývají **vstupní parametry**. Vstupní parametry jsou konkrétní informace potřebné pro daný vstup.

Pokud například chcete, aby se Edison pohyboval kupředu, musíte robotovi poskytnout konkrétní informace pro tento příkaz, například jak daleko a jakou rychlostí má jet.

V EdScratchi najdete tři druhy vstupních parametrů:

- **Čísla**, která zadáváte do bloku pomocí klávesnice,
- **rozbalovací menu**, v nich si vybíráte možnost,
- **díry ve tvaru kruhu nebo kosočtverce**, do nichž umísťujete speciální bloky.

Každý vstupní parametr v bloku poskytuje jinou informaci, kterou Edison potřebuje ke splnění daného příkazu. Vstupní parametry si můžete představit jako odpovědi na otázky, které má robot ohledně toho, co od něj požadujete. Například, pokud chcete, aby Edison jel dozadu, musíte odpovědět na tři otázky:

1. **Otázka:** Jak daleko má robot jet?

   → **Odpověď:** vstupní parametr „distance“ (vzdálenost)

2. **Otázka:** V jakých jednotkách se má měřit vzdálenost?

   → **Odpověď:** vstupní parametr „distance unit“ (délková jednotka)

3. **Otázka:** Jak rychle chcete, aby robot jel?

   → **Odpověď:** vstupní parametr „speed“ (rychlost)

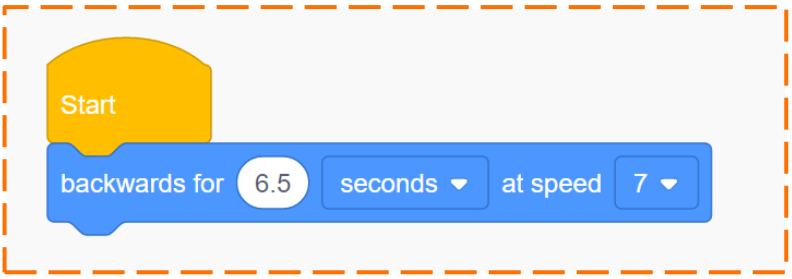
Je třeba vždy vyplnit všechny vstupní parametry v bloku, abyste robotovi poskytli všechny potřebné informace.

Jméno_____

Vyzkoušejte si to!

Vstupní parametry programu vám poskytují spoustu informací o tom, co program od Edisona požaduje. Přečtěte si následující programy a odpovězte na otázky.

Prohlédněte si tento program:



1. Jaký je v programu vstupní parametr pro ***distance units*** **(délkové jednotky)**?

   _____

2. Jaký je v programu vstupní parametr pro ***speed*** **(rychlost)**?

   _____

3. Co celý program Edisonovi přikazuje udělat? Každý vstupní parametr ve své odpovědi zvýrazněte jinou barvou či potržením nebo zakroužkováním.

_____

_____

Prohlédněte si následující program:

4. Kolik je v tomto programu vstupních parametrů?

   _____

5. Zapamatujte si, že vstupní parametry jsou odpovědi na otázky, které robot potřebuje znát. Na jakou otázku odpovídá vstupní parametr ***spin*** **(otočení)**? *Nápověda: při promýšlení odpovědi se podívejte na další možnosti pro tento vstupní parametr v EdScratchi.*

_____

_____

U2-2.3 Prozkoumejte Edisonův hudební talent

Výstupy využívající zvuky jsou jedním ze tří hlavních Edisonových výstupu. V EdScratchi naleznete bloky související se zvukovými výstupy v kategorii Sound (Zvuk).

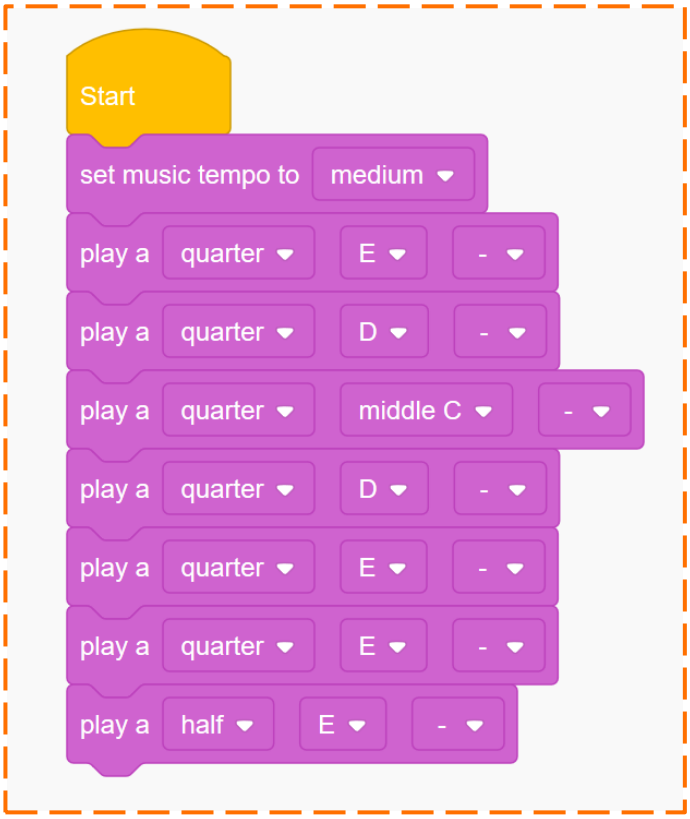
Nezapomeňte

Edison má také speciální zařízení, které můžete vidět na levé straně kulatého tlačítka na horní straně robota. Jedná se o bzučák a zvukový senzor v jednom. Dokáže rozpoznat zvuk, ale také jej vytvářet, např. pípat nebo vydávat tóny.

Projděte si v EdScratchi kategorii Sound (Zvuk). Není příliš rozsáhlá a není v ní mnoho bloků, takže byste si mohli myslet, že Edison nedokáže se zvuky příliš pracovat. Seřazením zvukových bloků do různých posloupností s různými vstupními parametry však můžete zahrát celou písničku!

Úkol 1: Zahrajte písničku

Napište v EdScratch následující program:



Stáhněte program do Edisona a spusťte jej. Tento program je první částí anglické písničky, která je vám možná povědomá. Znáte ji?

Jméno_____

Prvním blokem v programu je set music tempo (nastavit tempo hudby). Zkuste v rozbalovacím menu kliknout na možnost set music tempo a projděte si možnosti vstupních parametrů pro tento blok. Zvolte jiný vstupní parametr a stáhněte upravený program do Edisona. Spusťte nový program.

1. Jaký nový vstupní parametr jste pro blok set music tempo zvolili?

   _____

2. Co se stalo při přehrání nového programu? Jaká změna nastala v porovnání s původním programem?

_____

_____

_____

3. Jak blok set music tempo funguje?

_____

_____

_____

4. Co by se stalo, kdybyste blok set music tempo umístili na konec programu místo na začátek? *Nápověda: Mějte na paměti, že Edison čte bloky v EdScratchi po jednom od prvního po poslední.*

_____

_____

_____

Nápověda!

Nejste si jisti, co by se stalo, kdybyste provedli změnu nějakého programu? Nejlepší způsob, jak to zjistit, je tuto změnu udělat, stáhnout upravený program do Edisona a vyzkoušet jej!

V programování můžete vždy experimentovat!

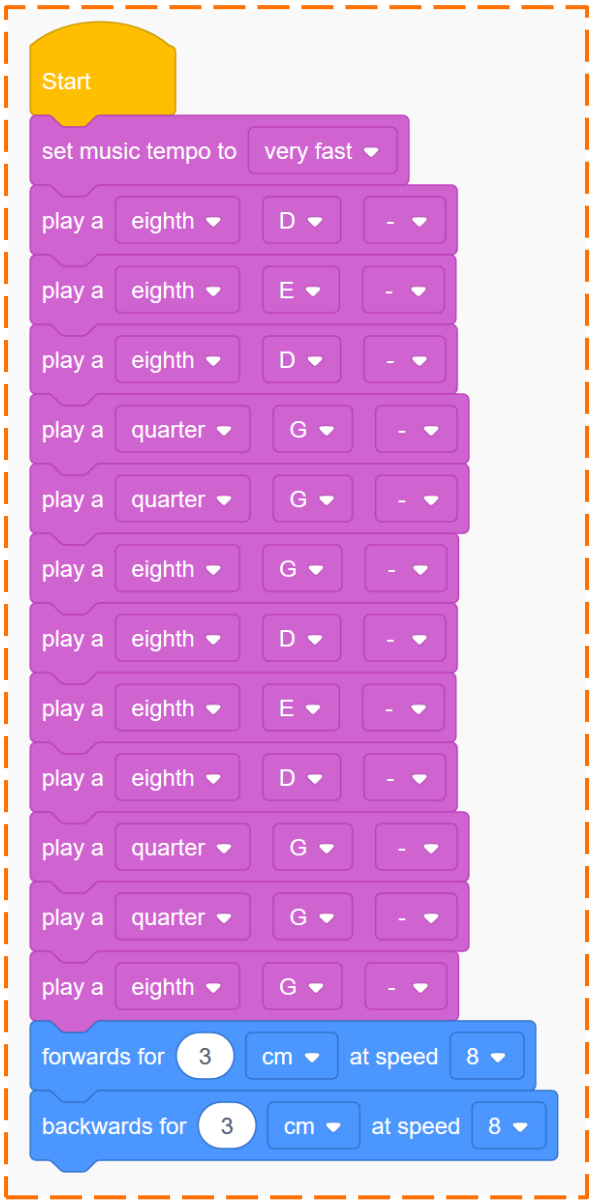
Použijte tento odkaz

Chcete, aby Edison zahrál z této písničky více? Otevřete program pod následujícím odkazem: https://www.edscratchapp.com?share=Eb12x3Dm

Jméno_____

Úkol 2: Tancuj podle muziky

Prohlédněte si následující program v EdScratchi:



Tento program obsahuje hudbu k první části písně *The Hokey-Pokey*. Dobře s u ní tancuje – píseň vám sama říká, jaké pohyby máte dělat!

Napište taneční program Hokey-Pokey v EdScratchi. Stáhněte program do Edisona a přehrajte jej.

5. Pohyboval se Edison podle hudby? Proč tomu tak podle vás je?

_____

_____

_____

Jméno_____

Edison dokáže přehrát tón, a přitom dělat i něco jiného, např. pohybovat se. Budete však muset robotovi říct přesně, co má dělat. K tomu budete potřebovat speciální blok v EdScratchi v kategorii Sound. Vypadá takto:

Blok play music in background (přehrát hudbu na pozadí) je seskupovací blok. V tomto seskupovacím bloku mohou být shromážděny jiné bloky.

Proč je to tak?

Tvar bloku v EdScratchi vám napoví, k čemu se daný blok využívá v programech. Podívejte se na blok play music in background (přehrát hudbu v pozadí). Podobá se trochu ústům, že? Ostatní bloky je možné do těchto „úst" vložit.

Tento seskupovací blok ovlivní všechny bloky umístěné v bloku play music in background. Pamatujte si, že Edison plní příkazy z bloků v EdScratchi po jednom. Robot tak nejprve uvidí seskupovací blok a dostane informace, že všechny bloky v něm budou plněny v režimu „přehrávání hudby v pozadí".

Přidejte blok play music in background do tanečního programu Hokey-Pokey. Zamyslete se nad tím, do které části programu tento blok patří.

Stáhněte upravený program do Edisona a spusťte jej. Edison by měl začít přehrávat píseň a zároveň se pohybovat!

Na Edisonových tanečních pohybech je však třeba ještě zapracovat.

Nápověda!

Nezapomeňte, že pokud něco není v programu zcela v pořádku, zobrazí se v ladicím panelu varovná zpráva. Tyto zprávy vám pomohou přijít na to, co by mělo být uvnitř seskupovacího bloku, a co ne.

Úkol 3: Tancuj podle hudby

Přidejte do tanečního programu Hokey-Pokey více tanečních pohybů. Můžete naučit Edisona řídit se podle textu písně Hokey-Pokey nebo si vymyslet vlastní tanec!

Dokážete Edisona naučit, aby tančil podle hudby?

Jméno_____

U2-2.4 Prozkoumejte chyby a jejich odlaďování

Chcete se dozvědět tajemství o práci s počítači a psaní kódů? Tady je:

VŽDYCKY SE NĚCO POKAZÍ!

Dobře, ve skutečnosti to není tajemství, ale je to velmi důležité. Významnou součástí algoritmického myšlení a práce s počítači je řešení problémů.

Nezapomeňte

Algoritmické myšlení znamená, že o problému nebo úkolu přemýšlíte podobně jako počítač. Jde o plánování, řešení problémů a analýzy informací stejným způsobem, jako to dělá počítač.

Pamatujte si, že když něco nefunguje tak, jak chcete, nic se neděje! Znamená to jen, že je třeba vyřešit problém. Jedním z hlavních druhů řešení problémů v informatice je hledání a odladňování **chyb**.

Ladění chyb je významnou součástí kódování a práce s roboty. Počítačoví vědci, programátoři a robotici stráví odladňováním chyb spoustu času. Nejspíš dokonce více než samotným psaním kódu!

Slovníček pojmů

Když něco v počítačovém programu nefunguje, říká se tomu **chyba (bug)**.

Hledání a opravě chyb v počítačovém programu říkáme **ladění chyb (debugging)**.

Proč je to tak?

Označovat chybu v programu slovem „hmyz“ (bug) se může zdát zvláštní, ale skutečně se jim tak v angličtině říká.

Proč se problémům u počítačů říká bugs? Tento pojem zavedla Grace Murray Hopperová, jedna z vynálezkyň moderního programování počítačů. Kdysi totiž zjistila, že problémem, který způsoboval nefunkčnost jejího počítače, byla opravdová můra, která se dostala do hardwaru! Řešením problému tak byl skutečně debugging (odstranění hmyzu).

Název se ujal a problémům se softwarem či hardwarem se v angličtině říká bugs a jejich odstraňování nazýváme debugging!

Jméno_____

Odlaďování chyb v EdScratchi

Ladicí panel v programovacím prostředí EdScratch je speciální funkce umožňující vám najít a odstranit chyby v kódu.

Nezapomeňte

Ladicí panel je oblast v EdScratchi pod paletou bloků a programovací plochou. Pokud je v programu chyba nebo se zdá, že něco není zcela v pořádku, zobrazí se v ladicím panelu chybová hláška.

Varovné hlášky, které se zobrazují v ladicím panelu, vám podávají informace o existujících či potenciálních problémech ve vašem kódu. Těmito zprávami vám EdScratch sděluje, že je v kódu chyba nebo že může k chybě dojít, když spustíte program v Edisonovi.
V kódování existují dva hlavní druhy chyb: **syntaktické chyby** a **logické chyby**.

Slovníček pojmů

Syntax je sada pravidel fungování programovacího jazyka. Všechny jazyky mají svá pravidla. U jazyků používaných lidmi, jako je např. čeština či angličtina, se tato pravidla týkají hláskování, gramatiky nebo psaní písmen či znaků v daném jazyce. Syntax je vlastně totéž, ale v počítačovém jazyce.

Syntaktické chyby jsou způsobeny tím, že při psaní kódu porušíte pravidla daného jazyka. Tyto chyby lze přirovnat k překlepům, nesprávnému hláskování nebo pravopisu.

Logika představuje organizovaný způsob myšlení, kterému počítač rozumí. Logika určuje tok programu, řazení položek v programu a to, jaké vstupy vedou k požadovaným výstupům.

Logické chyby jsou problémy v logice programu. Logické chyby jsou v podstatě problémy se způsobem myšlení použitým v programu. K příkladům patří programy, kterým počítač nerozumí nebo programy, které počítači přikazují něco, co nedokáže udělat. Pokud program funguje jinak, než jste očekávali, je pravděpodobné, že je v něm logická chyba.

Porozumění, zda je chyba výsledkem chyby syntaktické nebo logické chyby, vám může pomoci problém snadněji vyřešit. Pokud se v EdScratchi vyskytne syntaktická chyba, zobrazí se v ladicím panelu červená varovná zpráva a program nebudete moci do Edisona stáhnout. Pokud program můžete stáhnout a v Edisonovi spustit, ale nefunguje tak, jak byste chtěli, pravděpodobně to znamená, že došlo k logické chybě.

Jméno_____

Vyzkoušejte si to!

Prohlédněte si následující program:

Tento program je plný chyb. Vaším úkolem je opravit jej!

Nápověda!

Nejlepším způsobem, jak zjistit, co v programu funguje špatně – a co správně – je otestovat je v EdScratchi a v Edisonovi!

Jméno_____

1. Programátor, který napsal tento program, udělal dvě chyby v bloku set music tempo. V jednom případě jde o syntaktickou chybu, v druhém o logickou chybu. Napište vysvětlení obou chyb tak, aby byly pro programátora srozumitelné. *Nápověda: programátor chtěl, aby hudba hrála ve velmi rychlém tempu.*

Syntaktická chyba: _____

_____

_____

Logická chyba: _____

_____

_____

Podle programátora, který tento kód napsal, by měl program fungovat následovně:

> *„Chci, aby Edison hrál melodii velmi rychle. Zatímco bude hudba hrát, měl by robot udělat úplnou otočku doleva (360°), pak se otočit zpět doprava stejnou rychlostí a o stejnou vzdálenost.“*

2. Program nefunguje tak, jak si programátor představoval. Vaším úkolem je program odladit a upravit tak, aby fungoval podle programátorových představ. Vyzkoušejte program v Edisonovi a laďte jej tak dlouho, dokud nebude fungovat podle výše uvedeného popisu. Popište, jaké chyby jste odhalili a jak jste je napravili.

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Jméno_____

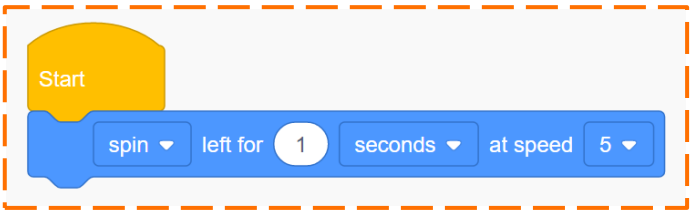
U2-2.5 Prozkoumejte Edisonovy motory

Robot Edison má dva motory: jeden na levé straně a druhý na pravé. Výstupy využívající těchto motorů jsou jedním ze tří hlavních typů Edisonových výstupů. V EdScratchi bloky související s výstupy motorů naleznete v kategorii Drive (Jízda).

Když píšete Edisonovi program v EdScratchi s použitím bloků z kategorie Drive, přikazujete motorům, co mají dělat. Většina bloků ovládá oba Edisonovy motory. Znamená to, že oba motory dělají totéž?

Úkol 1: Otočte robota

Pokud byste chtěli napsat kód přikazující robotovi „otočit se doleva“, stačí vám napsat jednoduchý s jedním blokem:



Vstupní parametry v tomto bloku informují Edisona o směru, vzdálenosti, délkových jednotkách a rychlosti, které má robot v programu použít.

Zvoleným směrovým vstupním parametrem je spin (otočit). To znamená, že celý vstupní parametr ohledně směru je spin left (otočit doleva). Jaký výstup tento vstup přikazuje Edisonovým motorům?

Napište program v EdScratchi s použitím stejných vstupních parametrů jako na obrázku. Stáhněte program a spusťte jej v Edisonovi na stole nebo na podlaze.

Nyní program spusťte znovu, ale tentokrát držte Edisona v rukou. Čeho jste si všimli?

1. Jakým směrem se otáčí levé kolo?

   _____

2. Jakým směrem se otáčí pravé kolo?

   _____

Edisonovy motory nemusí dělat současně totéž. Znamená to, že byste mohli napsat program pohybující pouze jedním z motorů? Můžete napsat program, který každému motoru řekne, co má dělat samostatně?

Otevřete si aplikaci EdScratch a prohlédněte si kategorii Drive (Jízda) v paletě bloků. Prostudujte si různé bloky a hledejte bloky, které byste mohli použít, pokud byste chtěli zajistit výstup pouze z jednoho z Edisonových motorů.

Jméno_____

3. Které z bloků podle vás využívají pouze jeden z Edisonových motorů? Proč tomu podle vás tak je?

_____

_____

_____

4. Edisona můžete využít ke stavění či vynalézání mnoha nejrůznějších věcí. Představte si, že byste potřebovali vytvořit něco s využitím pouze jednoho z Edisonových motorů. Co byste mohli postavit? Jak by váš výtvor využíval tento motor?

Nezapomeňte

Edisonova kola lze vyjmout z pohonných zdířek, v nichž jsou usazena. Edisonovy motory pohybují těmito zdířkami.

_____

_____

_____

_____

_____

_____

Úkol 2: Směr = dopředu

Aby Edison fungoval tak, jak chcete, ujistěte se, že jste mu zadali všechny potřebné informace. Pokud robot nemá všechny potřebné vstupy a instrukce, program pravděpodobně nebude fungovat tak, jak byste chtěli. Tento druh logické chyby může být velmi frustrující, především pokud se domníváte, že jste robotovi poskytli všechny požadované informace.

Jedním z hlavních způsobů, jak robotovi poskytnout tyto informace, jsou vstupní parametry v EdScratchi.

Nezapomeňte

Každý vstupní parametr v bloku poskytuje Edisonovi jinou informaci k provedení požadovaného příkazu. Vstupní parametry jsou vlastně odpovědi na robotovy otázky ohledně toho, co od něj požadujete.

Jméno_____

Některé bloky v EdScratchi získávají všechny informace, které potřebují, ze svých vlastních vstupních parametrů. Jiné bloky získávají informace z bloku, ale rovněž vyžadují další informace z programu.

Vytvořte Edisonovi program, který dá do pohybu jeho motory. Ke spuštění programu musíte zodpovědět čtyři otázky:

Otázka č. 1: Jakým směrem má robot jet?

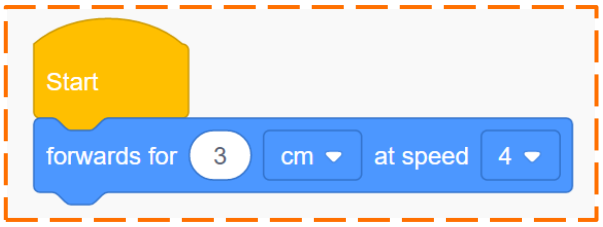
Otázka č. 2: Jak daleko má robot jet?

Otázka č. 3: V jakých jednotkách je měřena vzdálenost?

Otázka č. 4: Jak rychle má robot jet?

Váš program musí robotovi zodpovědět všechny tyto otázky

Prohlédněte si následující program:



Pokud byste spustili tento program v Edisonovi, dostal by robot všechny potřebné informace? Jinými slovy: sděluje tento program robotovi směr, vzdálenost, délkové jednotky i rychlost pohybu motorů?

5. Vyplňte následující tabulku. Pokud je daná informace obsažena v programu, napište hodnotu daného vstupu do sloupce „hodnota“. Např. hodnota „vzdálenost“ odpovídá na otázku „jak daleko má robot podle programu jet?“

Informace	Je v programu?	Hodnota
Směr		
Vzdálenost		
Délková jednotka		
Rychlost		

Napište program v EdScratchi, stáhněte jej a spusťte v robotovi.

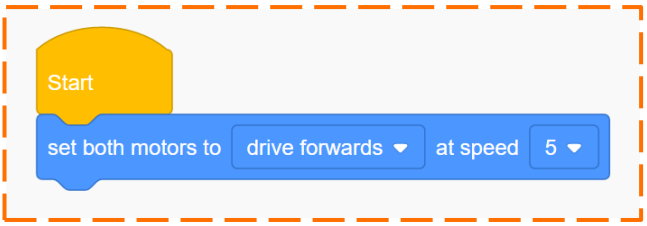
6. Co robot po spuštění programu provedl?

_____

_____

Jméno_____

Nyní si prohlédněte následující program:



Poskytuje tento program robotovi všechny potřebné informace?

7. Vyplňte následující tabulku. Pokud je daná informace obsažena v programu, napište hodnotu daného vstupu.

Informace	Je v programu?	Hodnota
Směr		
Vzdálenost		
Délková jednotka		
Rychlost		

Napište program v EdScratchi, stáhněte jej do robota a spusťte.

8. Co robot po spuštění programu udělal? Proč se choval právě tak?

_____

_____

_____

Jak můžete poskytnout robotovi zbytek potřebných informací, aby mohl uvést do provozu motory? Experimentujte v EdScratchi a ověřte si, zda umíte napsat program využívající blok set both motors (nastav oba motory) a žádný další blok z kategorie Drive a přitom robota rozjet vpřed.

Nápověda!

Nevzdávejte se! Experimentováním, testováním a řešením problémů se při psaní kódu učíte nové věci.

Zdá se vám, že jste se zasekli? Zkuste se podívat na to, čeho se snažíte dosáhnout, z jiného úhlu. Prohlédněte si různé bloky v EdScratchi a položte si otázku: 'pokud použiju tento blok, vyřeší to problém, kterým se zabývám?'

Pokud si nejste jisti, vyzkoušejte to a uvidíte, co se stane! Nezapomeňte si přečíst nápovědy v ladícím panelu!

Jméno_____

9. Co jste vyzkoušeli? Fungovalo to? Popište, jaké kroky jste podnikli, co fungovalo a co ne a jaké to pro vás bylo.

_____

_____

_____

_____

_____

_____

_____

_____

_____

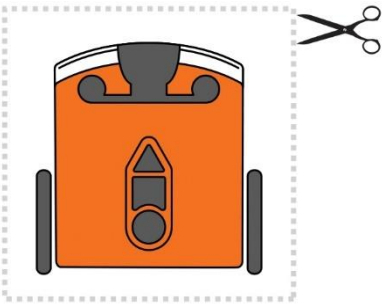
_____

_____

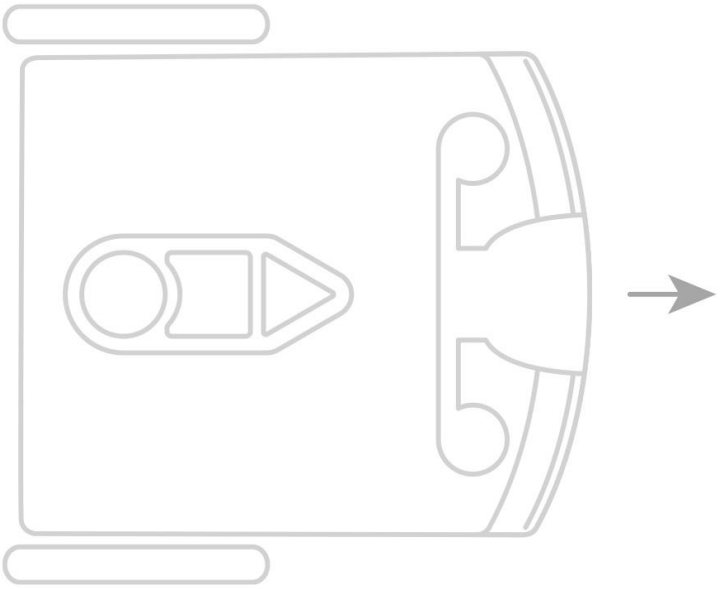
_____

Jméno_____

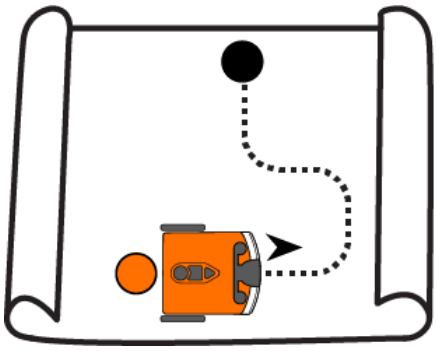
Pracovní list U2-1: Postupuj krok za krokem



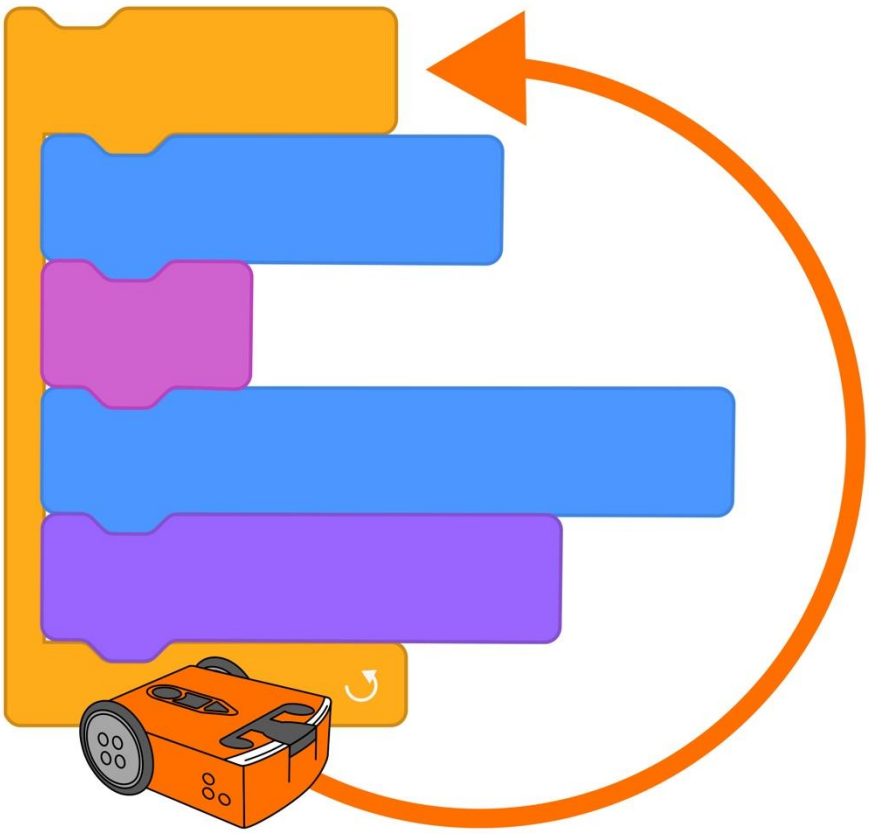
Pracovní list U2-2: Závodní dráha



Pracovní list U2-3: Mini bludiště



Lekce 3: Znáte cykly?



Edison

Jméno_____

U3-1.1 Prozkoumejte opakování kroků

Pokud si přejete, aby počítač, jako je např. robot Edison, udělal to, co chcete, musíte mu dát velmi konkrétní pokyny. Musíte napsat kód, v němž bude přesně uvedeno, v jakém pořadí se mají jednotlivé akce odehrát.

Nezapomeňte

Když píšete program pro Edisona v EdScratchi, říkáte robotovi, co má dělat a v jakém pořadí. Každý blok v EdScratchi představuje jednu akci, kterou robotovi říkáte, že má provést. Pořadí, v němž spojíte bloky v programu, informuje robota o sekvenci jednotlivých kroků. Edison tyto kroky provede postupně od prvního do posledního bloku.

Úkol 1: Jízda podél čtverce

Napište v EdScratchi program pro Edisona, který mu umožní jezdit po čtvercové trase. V programu byste měli k ovládání motorů použít výhradně bloky z kategorie Drive (Jízda). Stáhněte program do robota a otestujte jej na Pracovním listu U3-1. Ujistěte se, že Edison skončí jízdu přesně v místě, kde začal.

1. Kolik bloků obsahuje váš program? (Blok start nezapočítávejte.)

   _____

2. Prohlédněte si bloky ve svém programu. Všimli jste si něčeho? Lze v jejich použití najít nějaký pravidelný vzor?

_____

_____

Úkol 2: Použijte cyklus k jízdě podél čtverce

Aby Edison dokázal jet po čtvercové trase, musíte jej naprogramovat tak, aby jel podél každé strany čtverce a v každém rohu zatočil. Možná jste si všimli, že právě tato pravidelnost vede k pravidelné struktuře kódu: jeď po čáře a zatoč – jeď po čáře a zatoč – jeď po čáře a zatoč – jeď po čáře a zatoč naposledy do výchozí pozice.

V mnoha programech dochází k opakování – část kódu je využívána opakovaně. Opakování je jednou z věcí, v nichž jsou počítače velmi dobré. Na rozdíl od člověka se počítač nezačne nudit, když má nějakou věc dělat stále stejně a pořád dokola.

Představte si, že byste po Edisonovi chtěli, aby udělal tutéž věc 100×. Chtěli byste psát program se stovkou opakujících se bloků? Nebylo by pro vás psaní takového programu nudné? Myslíte, že byste dokázali napsat celý program a neudělat v něm chybu? Existuje přitom jednodušší a efektivnější způsob, jak počítač přimět k opakování příkazů. Opakování kódu je možné zajistit pomocí **cyklu** *(loop)*.

Slovníček pojmů

Cyklus (loop) je část kódu přikazující počítači opakovat určitý krok. Cykly jsou druhem **řídící struktury**, protože řídí jiné části kódu v programu.

Používání cyklů při psaní kódů nám umožňuje opakovat části kódu bez toho, abychom museli psát příkazy stále znovu. V EdScratchi se bloky pro cykly nacházejí v paletě bloků v kategorii Control (Řízení). Jedním z cyklických bloků v EdScratchi je repeat (opakuj):

Existují různé druhy cyklů. Blok repeat je **konečný cyklus**.

Slovníček pojmů

Konečný cyklus je cyklus s pevně stanoveným počtem opakování. Příkladem konečného cyklu je blok repeat v EdScratchi. Pomocí vstupního parametru bloku přikazuje cyklu, kolikrát se má akce opakovat.

Stejně jako všechny ostatní cyklické bloky v EdScratchi repeat obepíná ostatní bloky.

Proč je tomu tak?

Prohlédněte si dobře tvar bloku repeat. Vidíte, že se podobá ústům? Ostatní bloky je možné vložit do jeho otevřených „úst“. Kterýkoli blok vložený do bloku repeat se stává součástí cyklu. Všechny bloky v cyklu budou opakovány.

Pamatujte si, že Edison bude plnit příkazy v blocích jednotlivě, jeden po druhém. Robot nejprve uvidí cyklický blok, který jej informuje, že všechny bloky v daném cyklu je třeba zopakovat; počet opakování je určen vstupním parametrem bloku repeat. Robot pak provede jednotlivé akce definované bloky v cyklu v pořadí, v němž jsou seřazeny. Až se dostane na konec cyklu, vrátí se na její začátek a začne znovu!

Jméno_____

Zkuste pomocí bloku repeat napsat program, díky němuž Edison objede čtverec. Měli byste být schopni vytvořit pro Edisona program využívající pouze **tří bloků** po zahajovacím bloku start, včetně bloku repeat. Stáhněte program do robota a vyzkoušejte si jej na Pracovním listu U3-1. Ujistěte se, že Edison skončí jízdu na stejném místě, kde začal.

3. Jakou hodnotu musíte zadat do vstupního parametru bloku repeat, aby Edison objel čtverec?

   _____

4. Proč musí tato hodnota dosahovat právě této výše?

_____

_____

_____

Jméno_____

U3-1.1a Změňte to: Objeďte trojúhelník

I malé změny vstupů mohou vést ke zcela odlišným výstupům z programu. Skvělým příkladem je změna počtu opakování v cyklu. Představte si, že jste napsali program s cyklem čtyř opakování. Následně změníte počet opakování v tomto vstupu na pět. Co se stane, když spustíte aktualizovaný program?

Co máte dělat

Napište Edisonovi v EdScratchi program, který mu umožní objet trojúhelník. V programu musíte využít řídící strukturu s konečným cyklem, takže v něm určitě použijte blok repeat. Program by měl být co nejefektivnější, takže zkuste ke splnění úkolu použít co nejméně bloků.

Stáhněte program do robota a vyzkoušejte jej na Pracovním listu U3-2. Ujistěte se, že Edison skončí jízdu na stejném místě, kde začal.

1. Kolik bloků jste potřebovali k napsání úspěšně fungujícího programu? (Blok start nezapočítávejte.)

   _____

2. Jakou hodnotu musíte zadat do vstupního parametru bloku repeat, aby Edison objel trojúhelník?

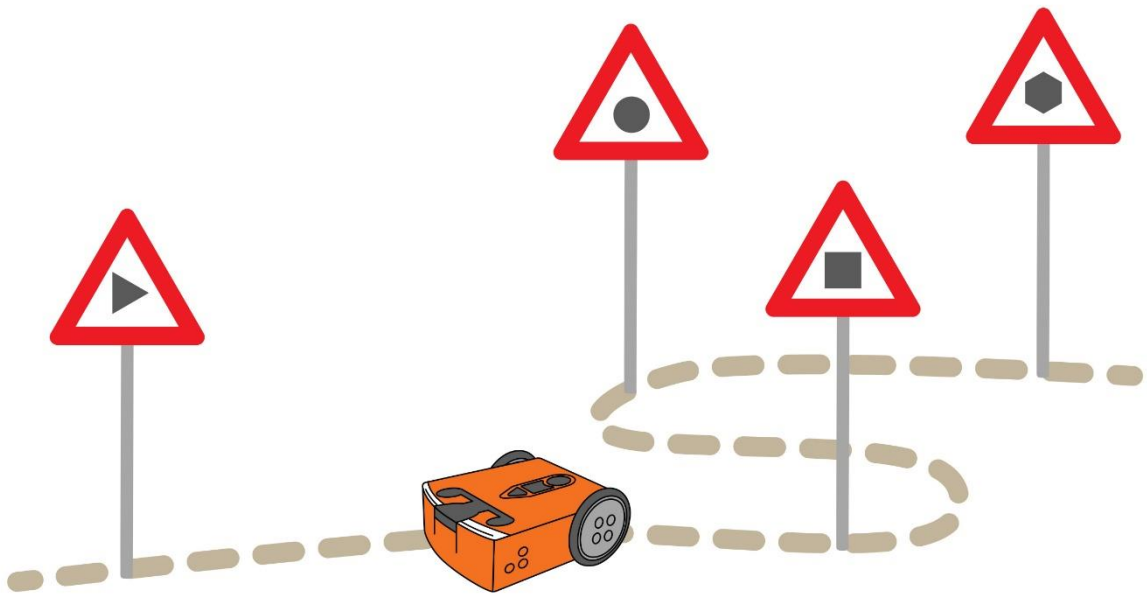
   _____

3. Proč musí tato hodnota dosahovat právě této výše?

_____

_____

_____



Jméno_____

U3-1.1b Změňte to: Objeďte šestiúhelník

I malé změny vstupů mohou vést ke zcela odlišným výstupům programu. Skvělým příkladem je změna počtu opakování v cyklu. Představte si, že jste napsali program s cyklem se čtyřmi opakováními. Následně změníte počet opakování v tomto vstupu na tři. Co se stane, když spustíte aktualizovaný program?

Co máte dělat

Napište Edisonovi v EdScratchi program, který mu umožní objet šestiúhelník. V programu musíte využít řídící strukturu s konečným cyklem, takže v něm určitě použijte blok repeat. Program by měl být co nejefektivnější, takže zkuste ke splnění úkolu použít co nejméně bloků.

Stáhněte program do robota a vyzkoušejte jej na Pracovním listu U3-3. Ujistěte se, že Edison skončí jízdu na stejném místě, kde začal.

1. Kolik bloků jste potřebovali k napsání úspěšně fungujícího programu? (Blok start nezapočítávejte.)

   _____

2. Jakou hodnotu musíte zadat do vstupního parametru bloku repeat, aby Edison objel šestiúhelník?

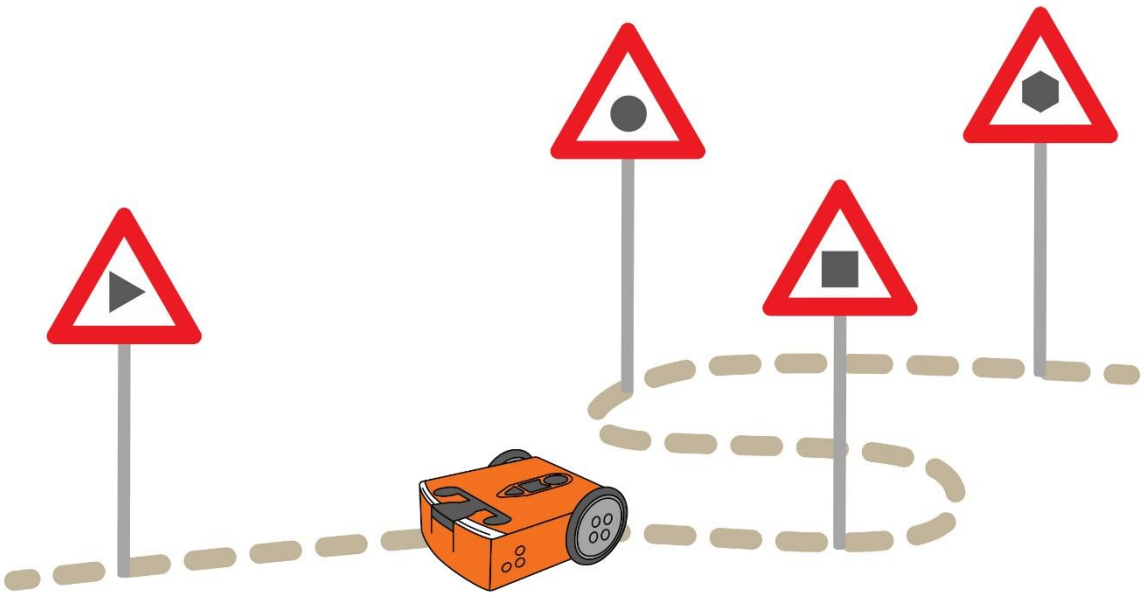
   _____

3. Proč musí tato hodnota dosahovat právě této výše?

_____

_____

_____



Jméno_____

U3-1.1f Další výzva: Kreslete různé tvary

Edisona můžete naprogramovat tak, aby jezdil v různých obrazcích či tvarech. Pokud k robotovi připevníte tužku, může tyto tvary i nakreslit!

Co máte dělat

Připevněte k Edisonovi pero. Můžete použít komponenty ze sady EdCreate nebo jakékoli jiné chcete.

Napište program v EdScratchi, s jehož pomocí robot namaluje nějaký tvar. Po připevnění pera a naprogramování spusťte program v Edisonovi a prohlédněte si nakreslený tvar!

1. Popište své zážitky z této výzvy. S jakými problémy jste se setkali? Jaké kroky jste podnikli, abyste tyto problémy překonali? Jak jste připojili pero? Jaký tvar jste zkusili nakreslit? Fungovalo to?

_____

_____

_____

_____

_____

_____

_____

_____

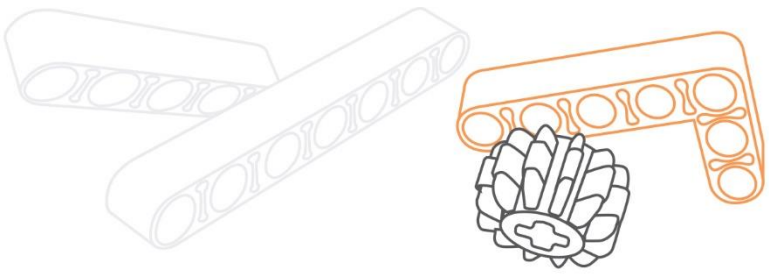
_____

_____

_____

_____

_____



Jméno_____

U3-1.2 Prozkoumejte cykly a sekvence

Cykly jsou při psaní kódů velmi užitečnou řídící strukturou. Pomocí cyklů lze zvýšit efektivitu programů, protože vám možní opakovat příkazy, aniž byste museli psát tytéž bloky znovu a znovu.

Při používání cyklů v programech však musíte věnovat velkou pozornost sekvenci, tj. posloupnosti bloků. To platí především, pokud tvoříte program, v němž je část kódu v cyklu a část mimo ni.

Zkuste napsat program, který umožní Edisonovi objet čtyřúhelník. Část kódu ve vašem programu bude v cyklu, ale část kódu bude muset být mimo cyklus.

Nezapomeňte

Sekvence znamená postup v určitém pořadí, krok za krokem.

Vyzkoušejte si to!

Čtyřúhelník je útvar se čtyřmi stranami. Jedním ze čtyřúhelníků je čtverec, ale existují i jiné čtyřúhelníky. Prohlédněte si čtyřúhelník na Pracovním listu U3-5. Tento čtyřúhelník má čtyři strany a čtyři úhly, které však nejsou identické.

Vaším úkolem je napsat Edisonovi v EdScratchi program, díky němuž dokáže objet tvar čtyřúhelníku na Pracovním listu U3-5. Váš program by měl využívat k dosažení požadovaných výstupů motorů bloky z kategorie Drive (Jízda). V programu rovněž musí být použit cyklus z kategorie Control (Řízení).

Musíte přijít na to, na kterém místě na Pracovním listu je pro Edisona nejvhodnější začít. Ujistěte se rovněž, že Edison skončí jízdu na stejném místě, kde začal.

Nápověda!

Promyslete si posloupnost kroků, které má podle vás Edison provést. Mějte na paměti, že když píšete Edisonovi v EdScratchi program, robot jej provádí postupně krok za krokem od prvního bloku k poslednímu.

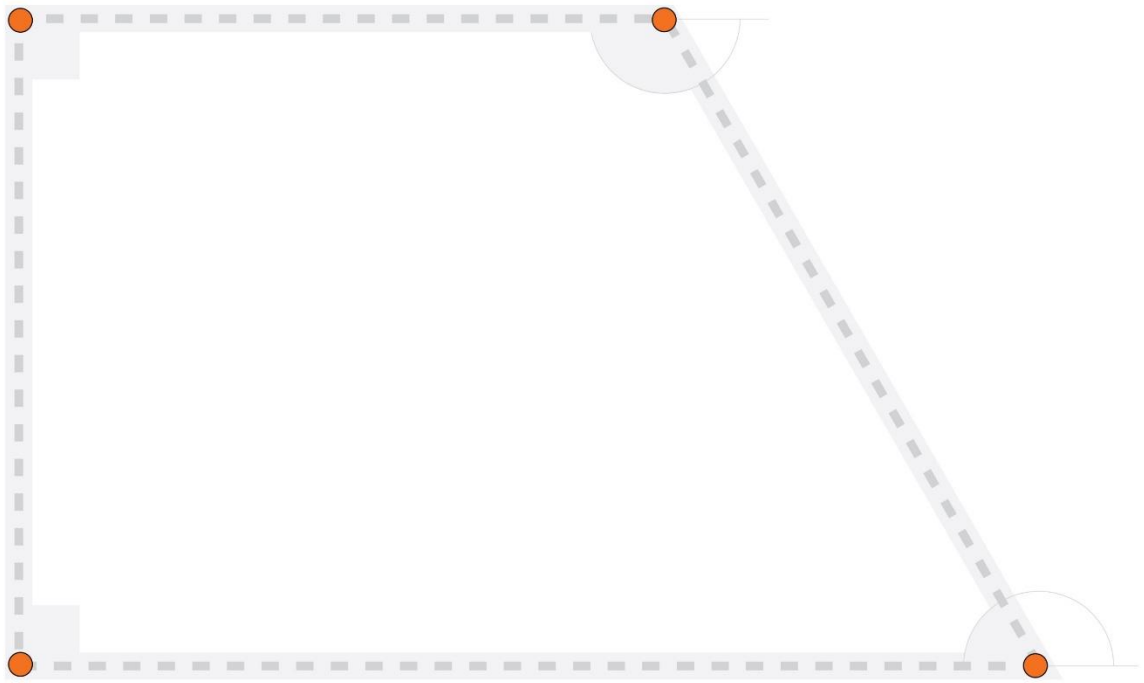
Napište program v EdScratchi. Stáhněte jej a vyzkoušejte na Pracovním listu U3-5.

Nápověda!

Vaším cílem nadále je, aby program byl co nejefektivnější, takže zkuste v kódu využít co nejméně bloků.

Jméno_____

1. Kde na čtyřúhelníku jste Edisona spustili? Vyznačte, kde jste Edisona spustili, včetně směru, jímž jste robota vyslali.



2. Prohlédněte si místo zahájení jízdy i svůj program? Jak volba místa zahájení ovlivnila sekvenci vašeho programu?

_____

_____

_____

_____

Jméno_____

U3-1.3 Prozkoumejte nekonečné cykly

Cykly umožňují opakovat kroky v programu, aniž byste museli psát tentýž kód stále znova. Pokud chcete, aby program provedl určitý krok opakovaně, je snazší použít cyklus než psát tentýž příkaz stále znovu. Kód je tak efektivnější, protože můžete dát počítači příkaz k opakování určitého kroku pomocí mnohem méně kódu.

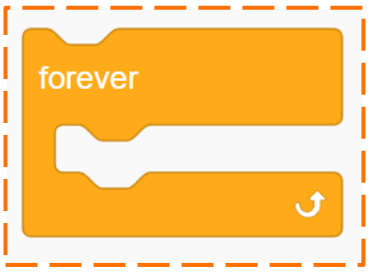
V mnoha programech s opakovanými příkazy je předem známo, kolikrát by se měl cyklus opakovat. Pokud např. chcete, aby robot jel po obvodu čtverce, musíte mu přikázat, aby jel v před a zatočil celkem čtyřikrát.
V programu tak použijete konečný cyklus se čtyřmi opakováními.

Nezapomeňte

Existují různé druhy cyklů. Konečný cyklus je cyklus s pevně daným počtem opakování. Konečným cyklem je např. blok repeat v EdScratchi.

Pokud chcete použít konečný cyklus, musíte vědět, kolikrát mají být kroky v rámci tohoto cyklu zopakovány. Co dělat, pokud to nevíte? Nebo pokud chcete napsat program, který se bude opakovat donekonečna?

Pokud chcete určitou akci v EdScratchi opakovat donekonečna, použijte speciální cyklický blok z kategorie Control - forever (nekonečný cyklus):



Blok forever je **nekonečný cyklus**.

Slovníček pojmů

Nekonečný cyklus je typem cyklu s neomezeným počtem opakování. Nekonečným cyklem je např. blok forever v EdScratchi. Tento blok přikazuje Edisonovi opakovat bloky uvnitř tohoto cyklu stále dokola.

Blok forever v EdScratchi si tak můžete představit jako blok repeat se vstupním parametrem počtu cyklů nastaveným na nekonečno!

Nápovědu ohledně použití bloku v EdScratchi v programovacím jazyku vám poskytne jeho tvar. Prohlédněte si tvar bloku forever. Stejně jako všechny ostatní cyklické bloky v EdScratchi blok forever obepíná ostatní bloky. Všechny bloky uvnitř cyklického bloku tak budou opakovány. Čeho dalšího jste si u tvaru tohoto bloku všimli?

Jméno_____

1. Když napíšete program využívající blok forever, myslíte si, že budete moci po tomto cyklu přidávat další příkazy? Proč ano nebo proč ne?

_____

_____

_____

_____

Vyzkoušejte si to!

Udělejte si z Edisona minutku na odměření času na vaření vajíček! Napište pomocí bloku forever program, díky němuž Edison po uplynutí daného množství sekund spustí navždy alarm.

Nápověda!

Program můžete kdykoli zastavit stisknutím tlačítka stop (čtvercového) na Edisonovi.

Zamyslete se nad sekvencí kroků nutných k tomu, aby minutka fungovala. Co by mělo být uvnitř cyklu? Co mimo cyklus? Stáhněte program do robota a otestujte jej.

2. Jak program vypadá? Napište program do rámečku níže. Zapište i použité vstupní parametry.

Jméno_____

U3-1.3a Další výzva: Vlezlá písnička

Vlezlá písnička je taková, která se vám dostane do hlavy, neustále vám zní v uších a nemůžete se jí zbavit. V této kapitole je vaším úkolem naprogramovat Edisona pomocí bloku forever, aby hrál zadanou melodii stále dokola!

Co máte dělat

Naprogramujte Edisona pomocí cyklického bloku forever tak, aby hrál písničku nebo melodii stále dokola. Napište program v EdScratchi, stáhněte jej do Edisona a otestujte.

Nezapomeňte

Program můžete kdykoli zastavit stisknutím tlačítka stop (čtvercového) na Edisonovi.

Jméno_____

U3-1.4 Prozkoumejte spojování a vnoření cyklů

Používání cyklů při psaní programů vám umožňuje efektivnější práci, protože můžete počítači dát příkaz k opakování kroků, aniž byste museli psát příkazy opakovaně. Použitím cyklů rovněž můžete počítači dát příkaz, aby něco dělal donekonečna, což by nebylo možné, pokud byste příkazy psali jednotlivě.

V programu můžete použít i více cyklů a můžete je používat různými způsoby: **spojováním** dohromady nebo **vnořením** do jiných cyklů *(nesting)*.

Proč byste měli bloky cyklů spojovat nebo je do sebe vnořovat? Použití více cyklů tímto způsobem vám umožní psát programy s opakovanými vzorci. Můžete dokonce napsat programy se vzorci, které opakují uvnitř jiných vzorců (jsou do nich vnořeny).

Proč je to tak?

Vzpomeňte si na budík na svém mobilním telefonu. Budík může být nastaven tak, aby zazvonil v 7 hodin ráno. Toto nastavení se může opakovat každý den. Když se budík spustí, ozve se předem nastavený počet pípnutí. Pokud budík „uspíte“ a signál odložíte, na nějakou dobu pípat přestane a po jejím uplynutí se opět ozve předem daný počet pípnutí.

Vidíte, že se jedná o opakované vzorce, vnořené do dalších opakovaných vzorců?

Budík je tak příkladem využití spojených a vnořených cyklů. Spojené a vnořené cykly umožňují opakování celých skupin příkazů v programu.

Spojování a vnoření cyklů se přitom využívá k různým účelům. Spojováním cyklů můžeme Edisonovi přikázat, aby opakovaně vykonal definované skupiny kroků a pak postoupil k další skupině opakovaných kroků. Vnoření cyklů nám umožňuje napsat Edisonovi program, při němž bude opakovat celé vzorce.

Úkol 1: Co se stane?

Programy, které využívají více cyklů, především vnořených, mohou na první pohled působit zmatečným dojmem. Abyste pochopili, co program bude dělat, musíte si představit každý jednotlivý krok, který se odehraje v dané sekvenci.

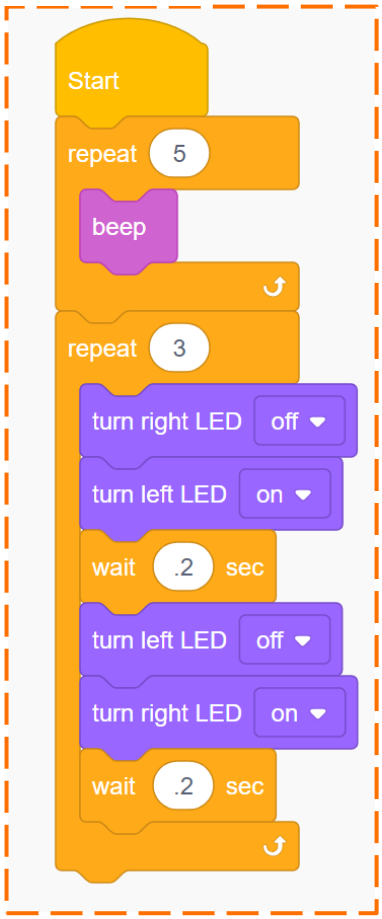
Nezapomeňte

Když píšete Edisonovi program v EdScratchi, robot vždy začne od nejvyššího bloku a další příkazy plní krok za krokem. Po dokončení jednoho bloku přejde k dalšímu. To platí pro všechny programy v EdScratchi – ať už v nich nejsou žádné cykly, obsahují jeden cyklus nebo více cyklů!

Jméno_____

Prohlédněte si následující programy a zodpovězte otázky ohledně toho, co se stane v každém programu.

Program 1:



Program 2:

Jméno_____

1. Program 1 – kolikrát se rozsvítí LEDka?

   _____

2. Program 1 – který krok skončí dříve: všechna pípnutí nebo všechna bliknutí LEDky?

   _____

3. Program 2 – kolikrát se rozsvítí LEDka?

   _____

4. Program 2 – který krok skončí dříve: všechna pípnutí nebo všechna bliknutí LEDky?

   _____

Nápověda!

Dokážete pochopit, co se stane v každém programu? Pokud si chcete své odpovědi ověřit, zkuste každý z programů napsat v EdScratchi, stáhněte je do Edisona a spusťe. Pak sledujte, co se stane.

Úkol 2: Jízda podle vzorce

Podobně jako vám cyklus umožní provést určený počet opakování vzorce, vnořené cykly vám umožní zopakovat více vzorců. Pro tuto aktivitu budete potřebovat Pracovní list U3-6.

5. Prohlédněte si vzorec na Pracovním listu U3-6. Jak byste tento vzorec popsali?

_____

_____

_____

Vaším úkolem je napsat program, který Edisonovi umožní jet podle vzorce nakresleného na Pracovním listu U3-6. Edison může přejet některou z čar více než jednou, ale musí se dotknout všech čar.

Jméno_____

6. Jak podle vás můžete využít vnořených cyklů k napsání efektivního programu, díky němuž Edison dokáže jet podle vzorce nakresleného na Pracovním listu?

_____

_____

_____

Zkuste napsat v EdScratchi program, který umožní Edisonovi jet podle vzorce nakresleného na Pracovním listu U3-6. Vyzkoušejte si použití a funkčnost vnořených cyklů.

Nápověda!

Program splňující úkol v tomto pracovním listu lze napsat pomocí pouhých pěti bloků v EdScratchi!

U3-2.1 Prozkoumejte přerušení hlavního programu

Různé programovací jazyky využívají různou syntaxi, tj. pravidla, takže se od sebe liší. Bez ohledu na syntax však jsou všechny programovací jazyky založeny na stejné logice. Všechny počítačové programy se proto chovají podobně a řídí se základní programovací logikou, např. sekvencí.

Nezapomeňte!

Logika je organizovaný způsob provádění kroků, kterému počítač rozumí. Logika určuje tok programu.

Syntaxe jsou pravidla fungování programovacího jazyka.

V EdScratchi logický tok program začíná od nejvyššího bloku s tím, že kroky definované dalšími bloky jsou vždy plněny jednotlivě, jeden po druhém. Sekvenčně postupují i programy obsahující cykly. Když program obsahuje cyklický blok, vykoná příkazy v tomto bloku ve stanovené posloupnosti (sekvenci). Když se robot či počítač dostane až na konec cyklu, vrátí se opět na začátek a začne plnit dané úkoly znova. Ačkoliv programy s cyklickými bloky vypadají odlišně, rovněž dodržují logický tok sekvence od prvního k poslednímu.

Existuje přitom způsob, jak tento sekvenční tok přerušit. Tok počítačového programu můžete přerušit použitím **přerušení** *(interrupt)*.

Slovníček pojmů

Přerušení *(interrupt)* je speciální část kódu, která zastavuje tok hlavního kódu. Přerušení se nazývá proto, že *přerušuje* hlavní kód. Přerušení se obvykle používá k zastavení hlavního kódu a spuštění **podprogramu** *(subroutine)*.

Podprogram je jedinečný kód nezávislý na hlavním programu. Jde vlastně o samostatný mini program.

Abyste pochopili, jak přerušení funguje, musíte rozumět tomu, co je přerušováno.

Co je to hlavní program?

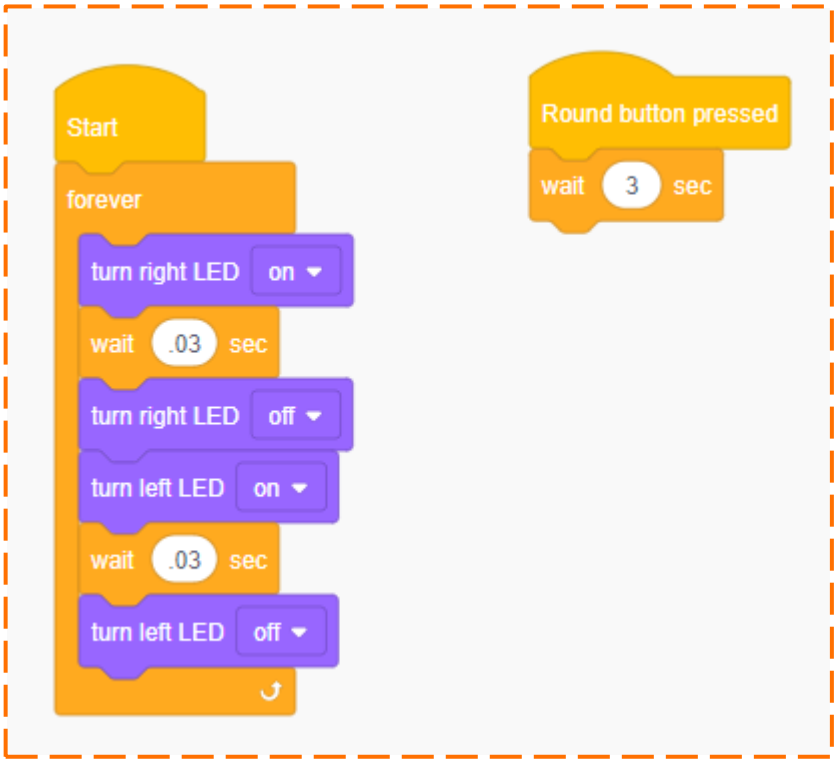
Hlavní program v EdScratchi je cokoli, co je připojeno za úvodní žlutý blok start.

Kdykoliv budete psát kód pro Edisona, v hlavním programu připojeném za blokem start musí být nejméně jeden další blok. Když spustíte program v Edisonovi stisknutím tlačítka pro

spuštění (trojúhelníkové), hlavní program poběží v postupné posloupnosti krok za krokem až do konce.

Tento tok může být přerušen tzv. přerušením.

Prohlédněte si následující program:



Tento program se skládá ze dvou částí: hlavního programu a podprogramu.

1. Co hlavní program přikazuje Edisonovi udělat? *Nápověda:* hlavní program je tvořen pouze bloky připojenými k bloku start.

_____

_____

_____

Kromě hlavního programu je součástí programu i podprogram. Kdy dojde k přerušení hlavního programu a spuštění podprogramu?

Ke spuštění podprogramů dochází pouze v případě konkrétní **události.**

Slovníček pojmů

V programování **událost** *(event)* označuje jev, který nastane mimo kód a ovlivní běh programu. Událostí může být např. stisknutí tlačítka nebo přenos informace ze senzoru.

Jméno_____

K zahájení podprogramu v EdScratchi musíte použít blok z kategorie Events (Události).

2. Prohlédněte si bloky v kategorii Events v EdScratchi. Čeho jste si všimli na jejich tvaru?

_____

_____

Bloky z kategorie Events přikazují programu vyhledávat konkrétní událost. Pokud k této události dojde, blok Event okamžitě přeruší hlavní program a spustí podprogram. Po dokončení podprogramu se program vrátí do místa, v němž byl přerušen.

Proč je to tak?

Přerušení v programování používáme proto, že umožňují programu zareagovat na jakoukoli událost, které nastane v průběhu běhu programu. Při použití přerušení nemusíte předvídat, kdy k dané události dojde. Bez možnosti přerušení byste museli přesně vědět, kdy daná událost nastane, i kdyby tato událost byla mimo vaši kontrolu!

Vyzkoušejte si to!

Napište v EdScratchi program, který obsahuje jak hlavní program, tak podprogram, přesně podle obrázku uvedeného výše v této aktivitě. Stáhněte program do Edisona. Stiskněte na robotovi tlačítko pro spuštění (trojúhelníkové). Tím se spustí hlavní program a Edisonovy LEDky začnou blikat. Nyní stiskněte kulaté tlačítko na robotovi. Tím dojde k přerušení hlavního programu a spuštění podprogramu. Podprogram přikáže Edisonovi vyčkat 3 sekundy a pak se vrátit do hlavního programu.

Tento program můžete využít k tomu, abyste z Edisona udělali rozhodovacího robota!

Vymyslete si nějakou otázku, na kterou lze odpovědět „ano“ či „ne“. Rozhodovací robot vám pomůže tuto otázku zodpovědět. Pokud při stlačení kulatého tlačítka svítí pravá LED, odpověď na vaší otázku je „ano“, pokud levá LED, odpověď je „ne“.

Proč je to tak?

Přerušení zastaví hlavní program okamžitě, takže je možné, že v okamžiku spuštění podprogramu nebude svítit žádná LEDka. Pokud hlavní program v okamžiku přerušení vypnul pravou LEDku, ale ještě nezapnul levou, budou vypnuty obě LED. Je to trošku, jako byste si házeli mincí a mince přistála na hraně – vzácná situace, ale může nastat!

Jméno_____

U3-2.2 Prozkoumejte komentáře při psaní kódů

Součástí výuky psaní kódu je výuka cizího jazyka: programovacího jazyka. Čím více kódu v programovacím jazyce napíšete, tím snadnější pro vás bude porozumět programům napsaným v tomto jazyce. Budete schopni si program přečíst příkaz po příkazu a představit si, co udělá, když jej spustíte.

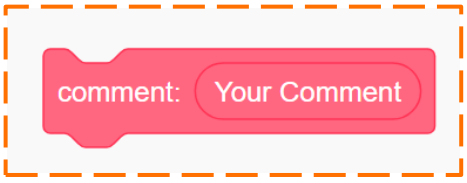
Existuje i nástroj, který nám usnadňuje čtení programů: **komentáře**.

Slovníček pojmů

V programování jsou **komentáře** poznámky, které programátor přidává ke kódu, aby v něm usnadnil orientaci. Komentáře jsou zprávy, které popisují a vysvětlují, co se v programu děje.

Vzhled komentářů závisí na používaném programovacím jazyce. Někdy mohou vypadat trochu jako kód, avšak komentáře ve skutečnosti nejsou kód. Když v počítači běží program s komentáři, počítač tyto komentáře ignoruje. Komentáře jsou určeny výhradně lidem!

V EdScratchi můžete ke svému přidat komentář pomocí speciálního bloku v kategorii Comments (Komentáře). Blok pro komentáře vypadá takto:



Blok comment vypadá podobně jako ostatní bloky v EdScratchi, ale funguje odlišně. Vidíte v bloku nápis „Your Comment“ (*váš komentář*)? Do tohoto místa můžete vložit svou poznámku. Mějte na paměti, že tento blok netvoří kód pro Edisona. Když Edison v programu narazí na blok s komentářem, jednoduše jej přeskočí a přejde na následující blok. Zprávu, kterou napíšete do bloku pro komentáře, můžete považovat za vstupní parametr daného bloku, avšak místo aby obsahoval informace pro Edisona, obsahuje poznámku pro jinou osobu.

Jak používat komentáře?

V mnoha ohledech je to, jak použijete v programu komentáře, jen na vás. Komentáře se nemusí řídit syntaxí, takže není nutné je psát konkrétním způsobem. Poznámky do komentářů můžete formulovat tak, jak se vám

Proč?

Protože jsou komentáře psány pro lidi, nemusíte mít obavy, zda vám počítač bude rozumět. Komentáře proto nemusí dodržovat syntaxi programovacího jazyka.

budou zdát nejsrozumitelnější. Komentáře můžete do kódu vkládat, kdykoli budete cítit potřebu vysvětlit, co bude následovat.

Přidávání komentářů usnadňuje ostatním lidem čtení vašeho programu, ale tím, kdo bude v budoucnu program číst, budete s největší pravděpodobností vy sami! Komentáře jsou totiž užitečným nástrojem při odlaďování programů (debugging).

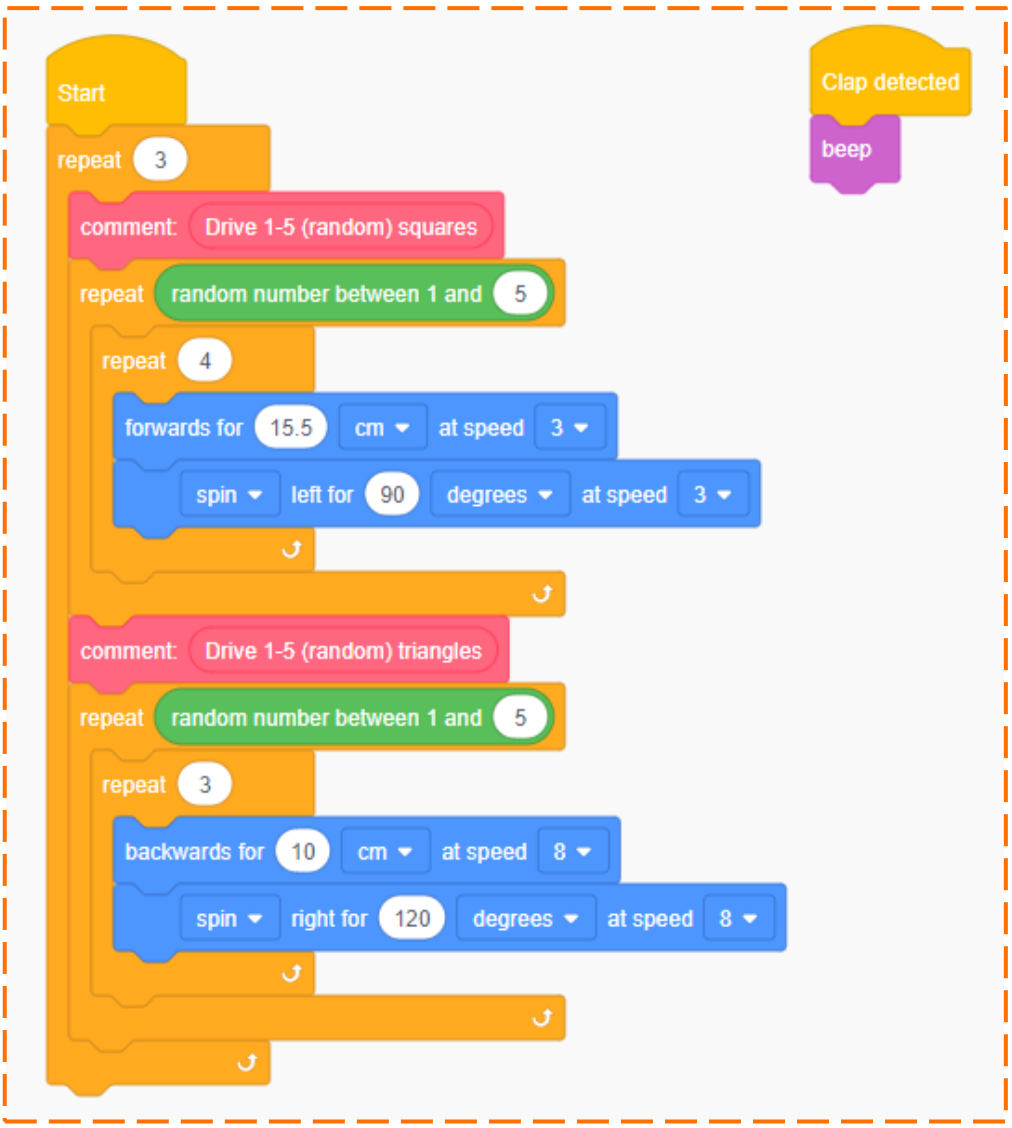
Přidáním komentáře můžete uspořádat své myšlenky a poznamenat si, co je vaším cílem. Pokud něco v programu nebude fungovat podle vašich představ, usnadní tyto poznámky nalezení zdroje problému.

Nezapomeňte

Nalézání a opravě problémů v počítačovém programu se říká ladění (debugging).

Vyzkoušejte si to!

Prohlédněte si následující program:



Jméno_____

Programátor přidal do kódu několik komentářů, které usnadňují čtení programu. Tento obrázek použijte k zodpovězení následujících otázek.

1. Co podle vás Edison udělá, pokud do něj naprogramujete kód z obrázku?

_____

_____

_____

_____

2. Chápete díky komentářům snadněji, co programátor chtěl, aby program udělal? Proč ano, nebo proč ne?

_____

_____

_____

_____

Nyní zkuste do Edisona naprogramovat program z obrázku.

3. Fungoval program tak, jak jste očekávali? Popište cokoli neočekávaného, k čemu došlo.

_____

_____

_____

_____

Když programátor tento program spustil, stalo se něco nečekaného:

„Chtěl jsem, aby Edison pípnul pouze, když tlesknu, ale robot namísto toho pípá celou dobu, kdy program běží. Proč je tomu tak?“

4. Proč Edison pípá? *Nápověda: Lze klíč k záhadě najít v prostředí EdScratch?*

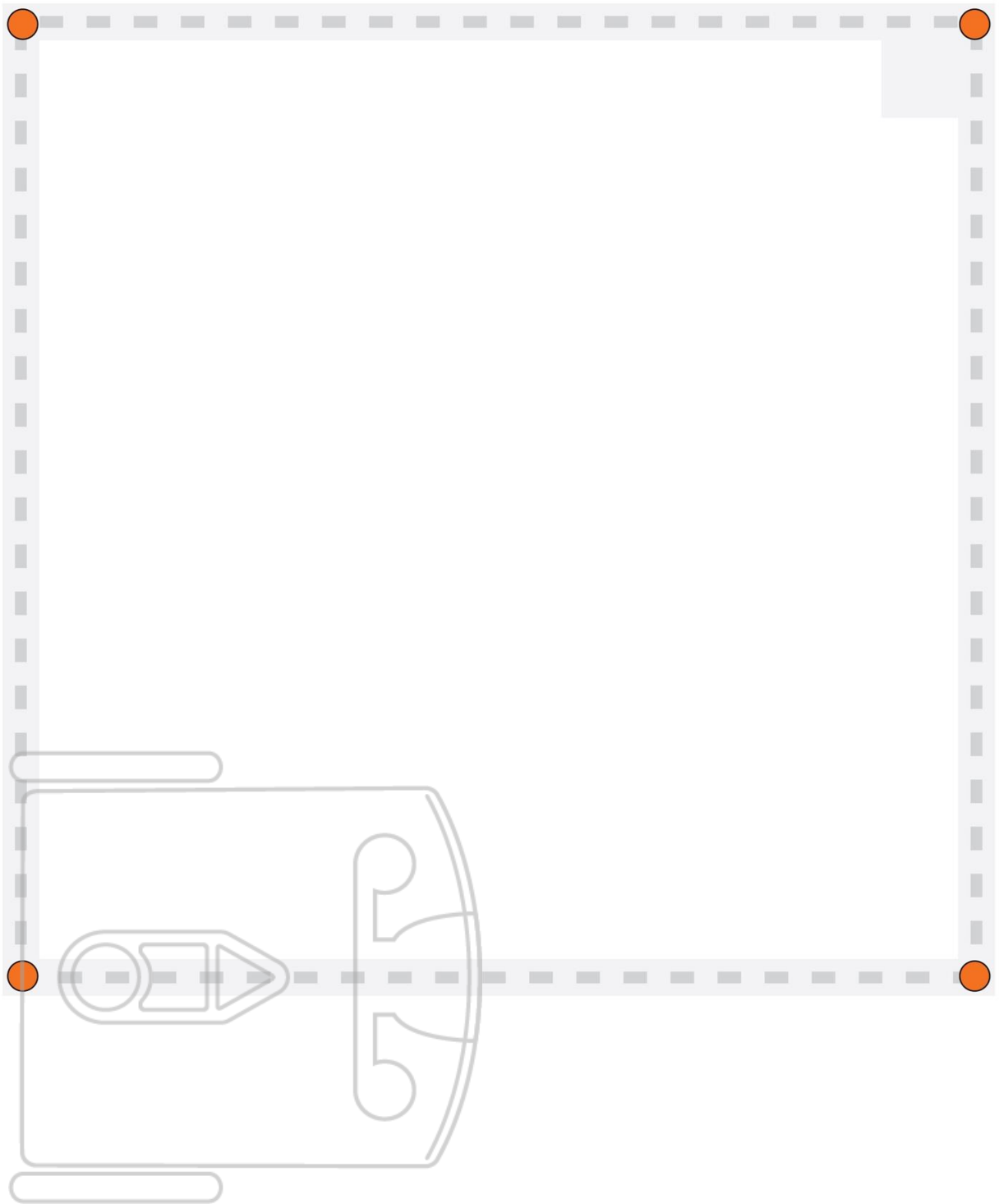
_____

_____

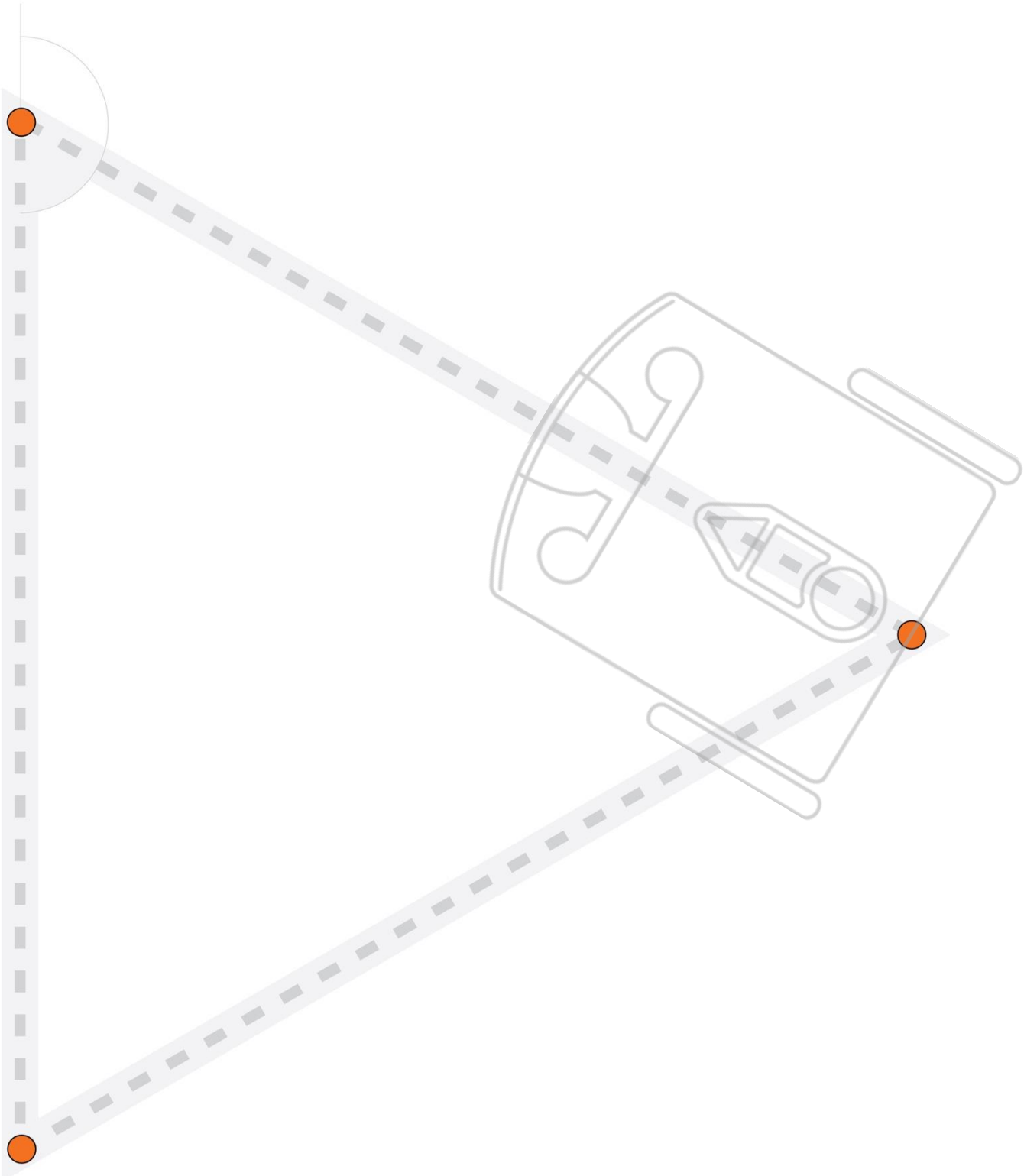
_____

Jméno_____

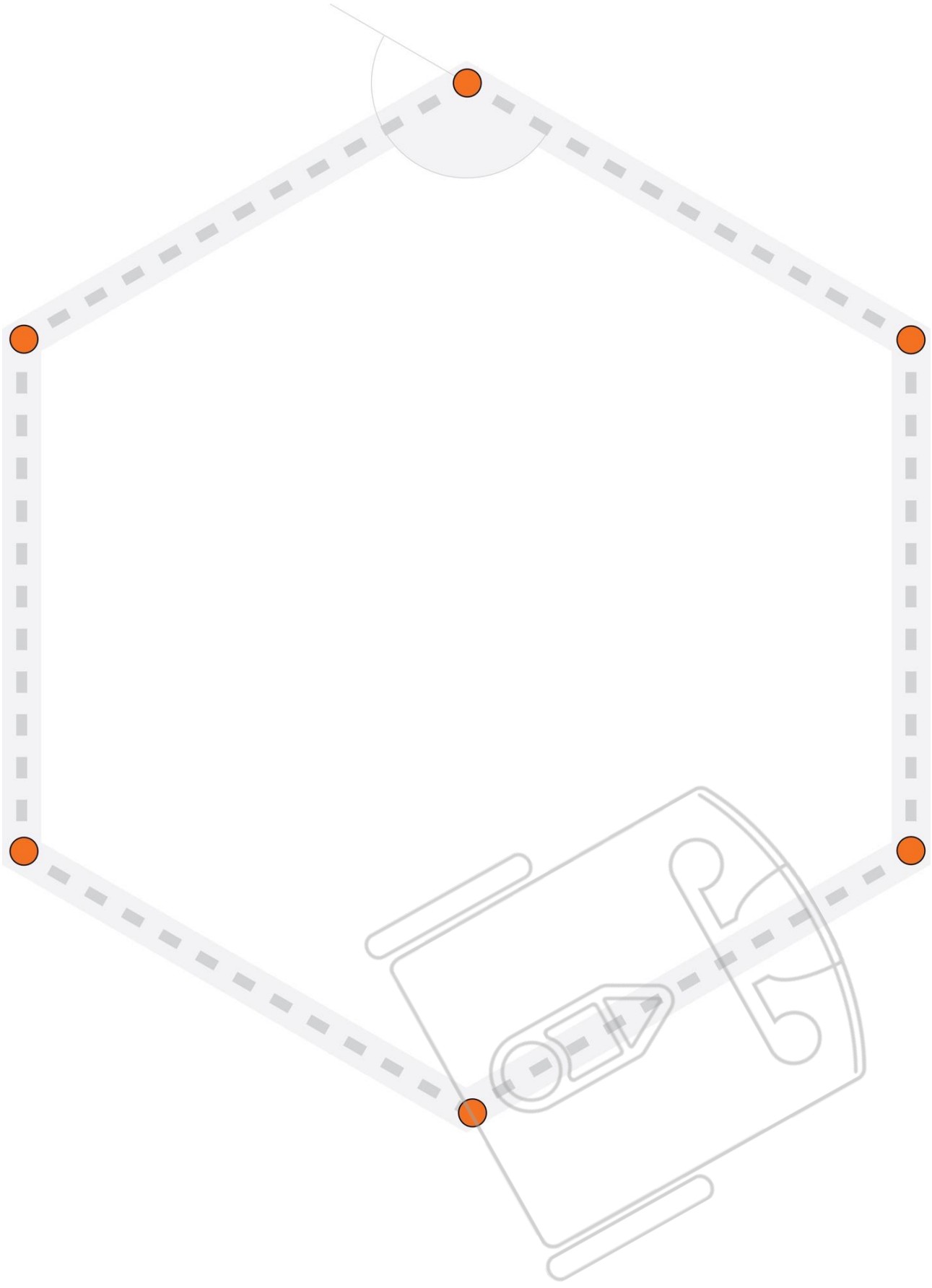
Pracovní list U3-1: Jízda podél čtverce



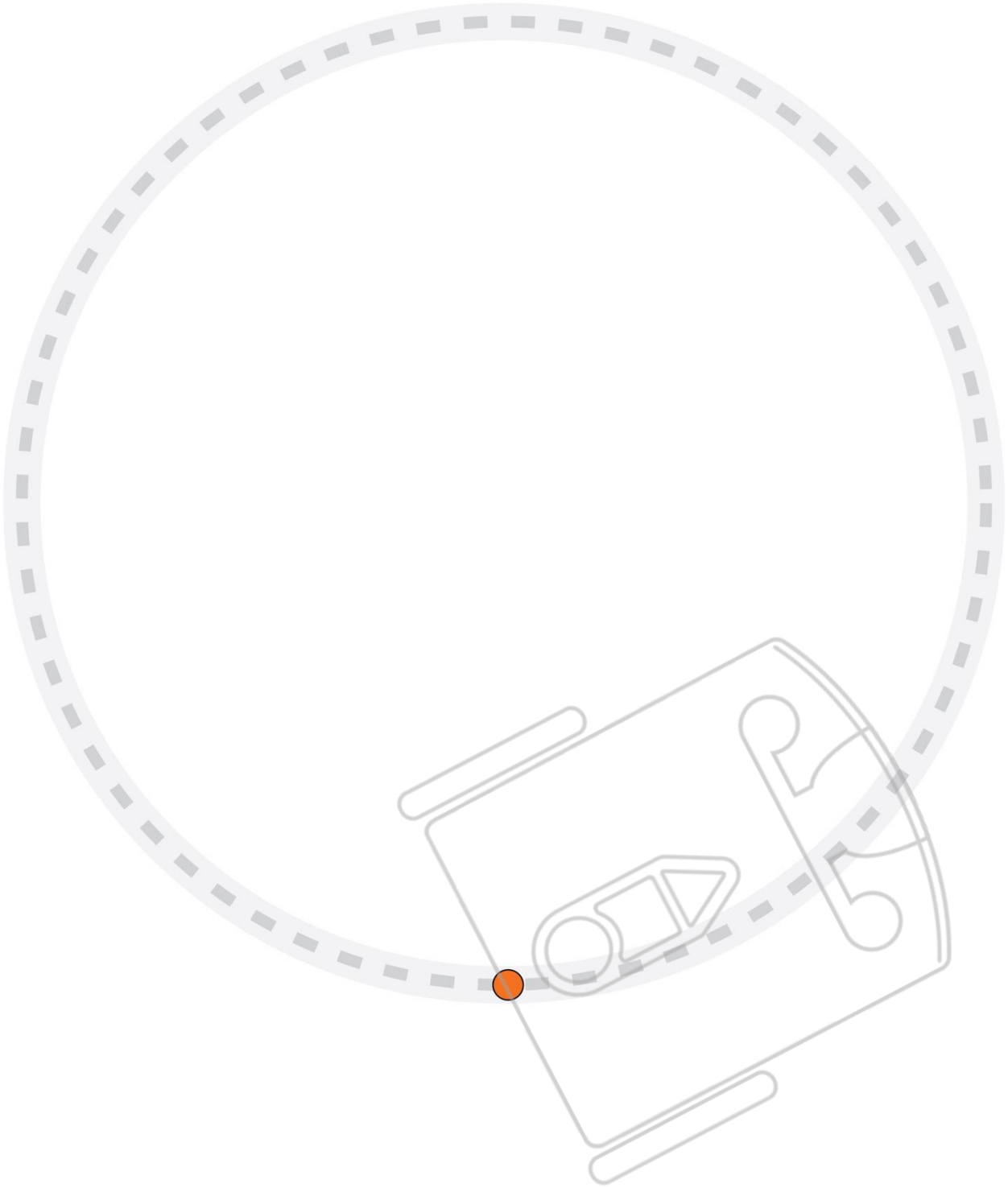
Pracovní list U3-2: Jízda podél trojúhelníku



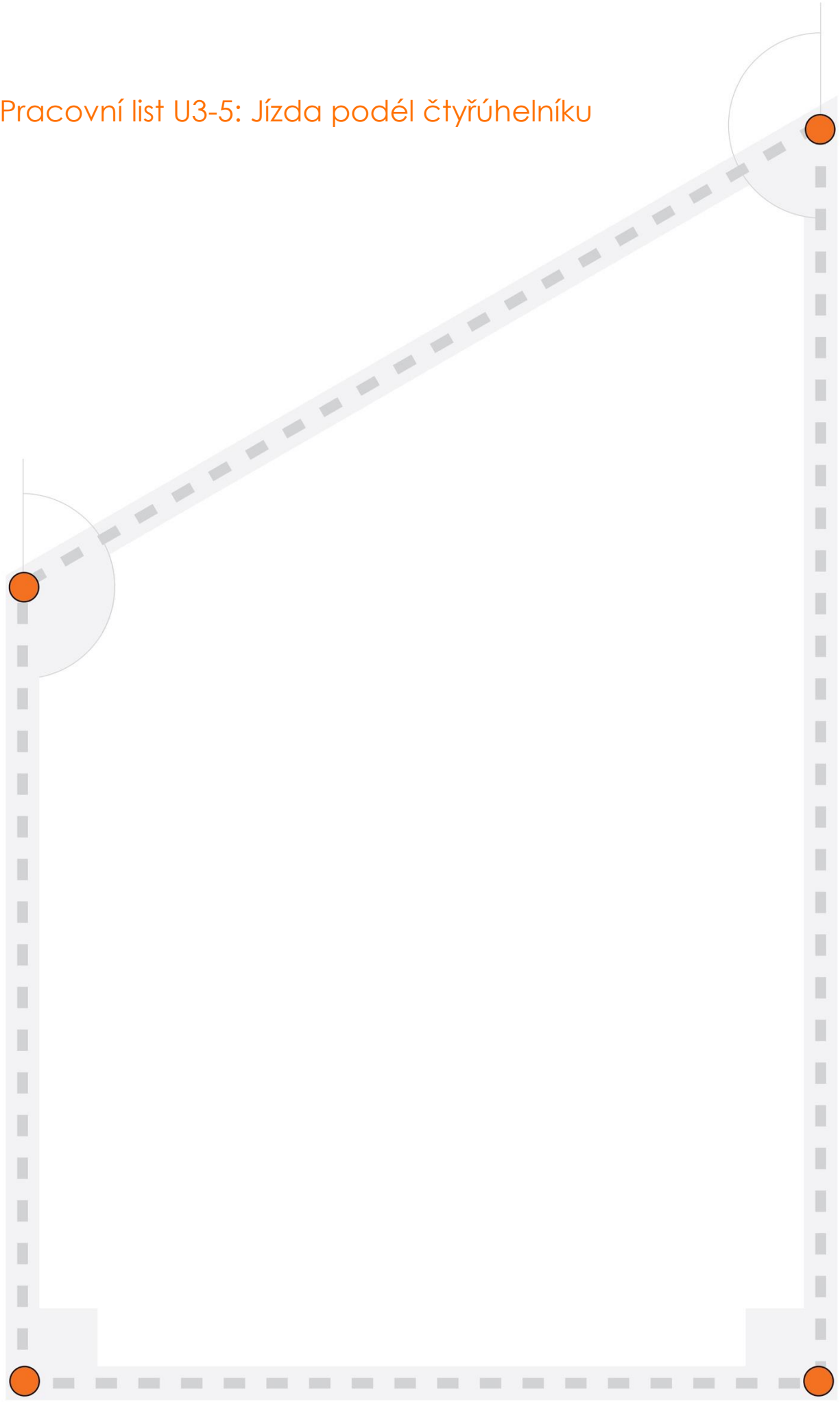
Pracovní list U3-3: Jízda podél šestiúhelníku



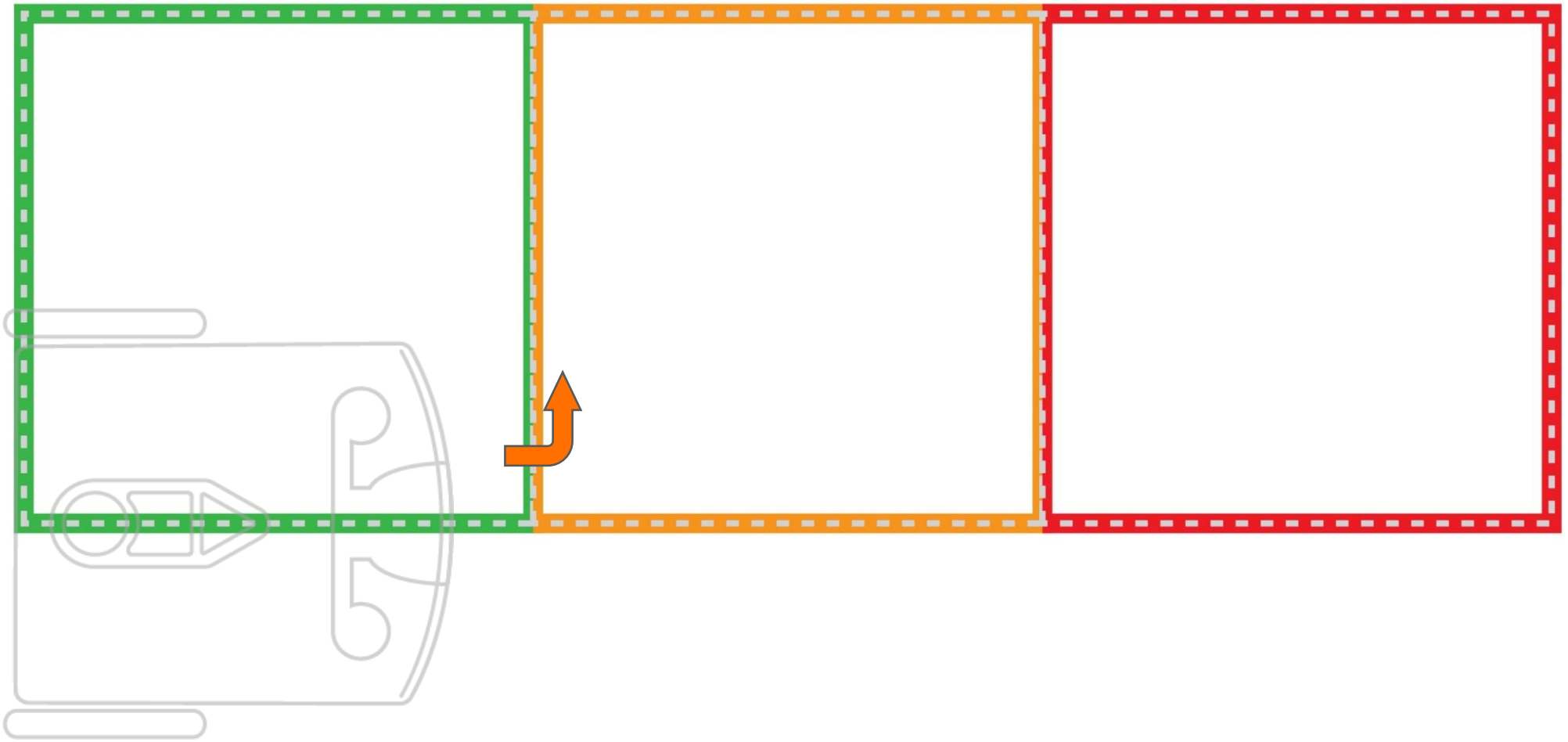
Pracovní list U3-4: Jízda podél kruhu



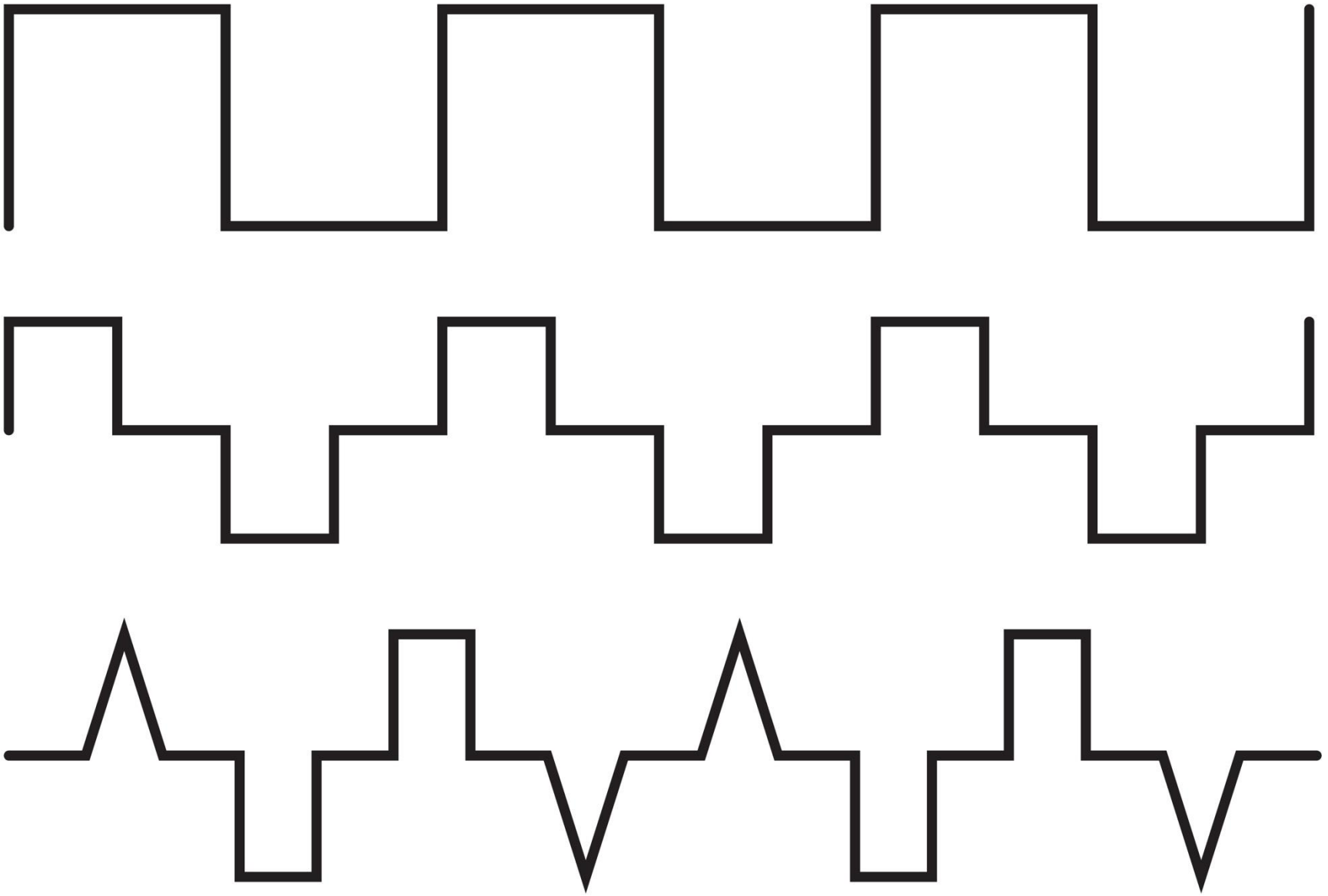
Pracovní list U3-5: Jízda podél čtyřúhelníku



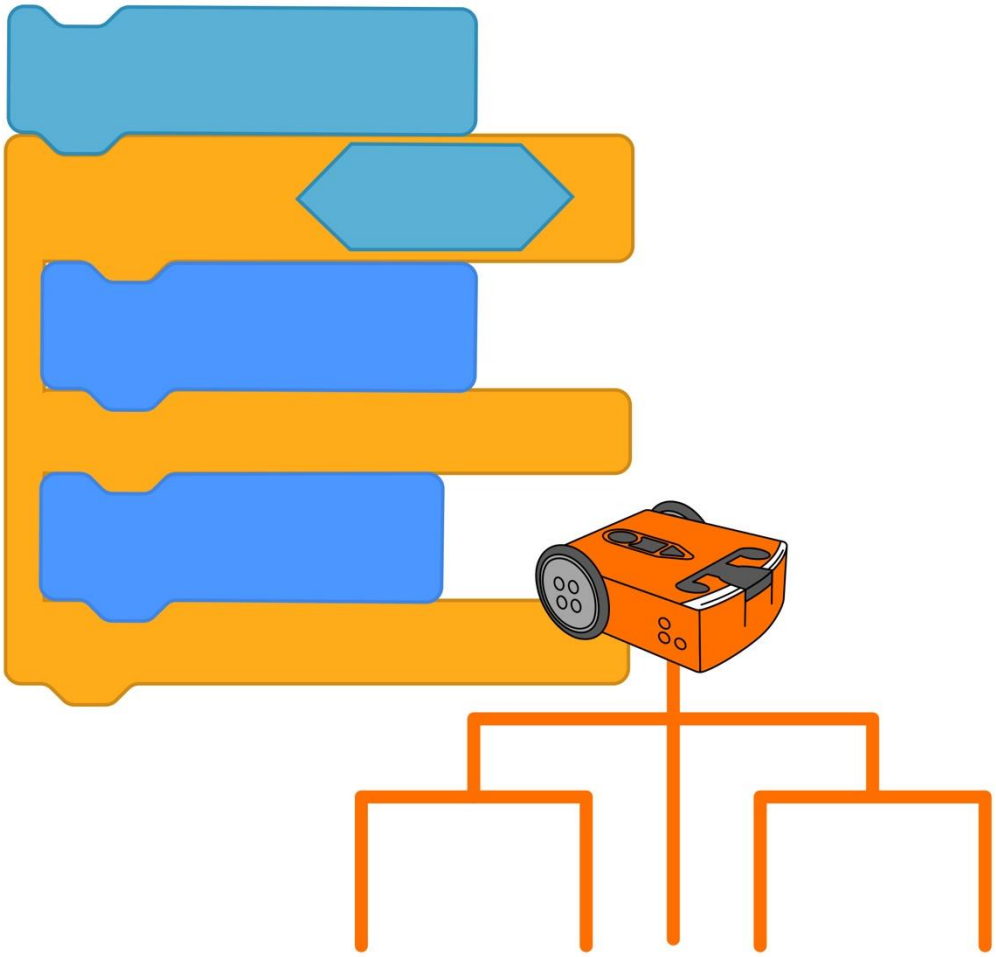
Pracovní list U3-6: Opakování čtverců



Pracovní list U3-7: Vzory na projetí



Lekce 4: Co když…



Edison

U4-1.1 Prozkoumejte používání podmíněných příkazů

Abyste přiměli počítač, jako je např. robot Edison, aby udělal to, co chcete, musíte mu dát velmi konkrétní pokyny formou počítačového programu. Počítač se pak krok za krokem řídí vašim kódem. Dělá, cokoli mu řeknete.

Ale co kdybyste chtěli, aby počítač rozhodoval sám o sobě?

Většina počítačů, včetně robota Edison, nedokáže dělat složitá rozhodnutí tak, jako člověk. Můžete však naučit Edisona přijímat jednoduchá rozhodnutí. Robot však stále potřebuje, abyste mu dali přesné pokyny, aby věděl, jaké rozhodnutí dělá. Rovněž potřebuje k těmto rozhodnutím pravidla, neboli **podmínky** *(condition)*. K napsání tohoto druhu programu musíte použít druh řídící struktury zvaný **podmíněný příkaz** *(conditional)*.

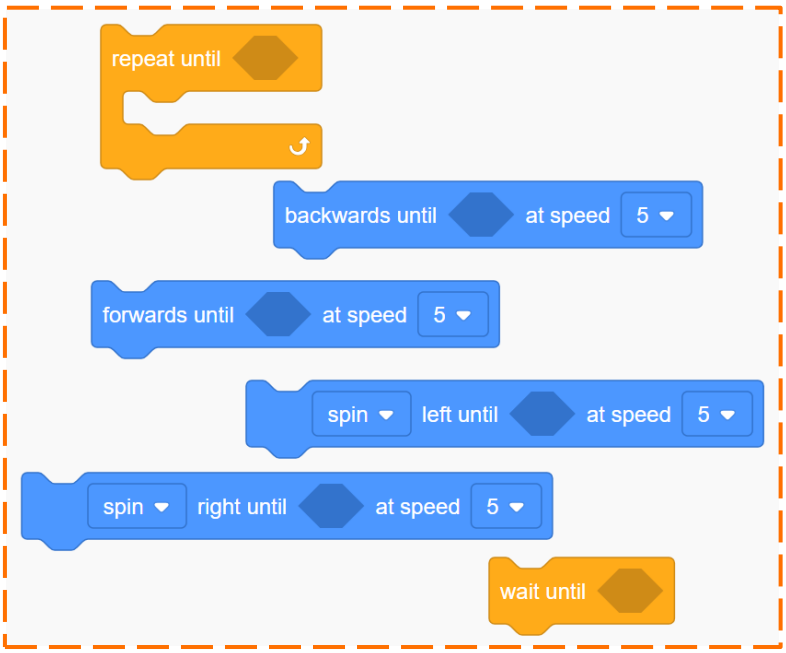
Slovníček pojmů

Podmíněný příkaz je prvek kódu závisející na jiném faktoru, tj. podmínce. Kroky určené touto částí kódu proběhnou pouze, pokud bude splněna tato **podmínka**.

I v kódu je **podmínka** předem určená okolnost nebo soubor faktorů, které je třeba splnit, aby mohl být podmíněný kód spuštěn.

Podmíněné příkazy pro Edisona můžete v EdScratchi využít např. k napsání programu, který robotovi přikáže provádět určitou činnost, dokud (until) nebude splněna stanovená podmínka.

Prohlédněte si bloky v EdScratchi na tomto obrázku:



Všechny tyto bloky jsou podmíněné příkazy využívající tentýž vzorec stanovený *podmínkou* until. Každý blok přikazuje Edisonovi provádět určitou akci, dokud není naplněna konkrétní podmínka. Jaká podmínka to přesně je?

Prohlédněte si znovu všechny bloky until na obrázku. Vidíte kosočtvercový prostor v každém bloku? Podmínku pro blok until stanovíte zadáním zvláštního vstupního parametru do tohoto prostoru.

Nezapomeňte

V EdScratchi existují tři druhy vstupních parametrů:

- čísla, která zadáváte do bloku pomocí klávesnice,
- rozbalovací menu, v nich si vybíráte možnost,
- díry ve tvaru kruhu nebo kosočtverce, do nichž umísťujete speciální bloky.

Každý vstupní parametr v bloku poskytuje jinou informaci, kterou Edison potřebuje ke splnění daného příkazu. Vstupní parametry si můžete představit jako odpovědi na otázky, které má robot ohledně toho, co od něj požadujete.

Podobně jako u všech ostatních bloků s kódem je pro správné fungování podmíněných příkazů třeba zadat přesné vstupní parametry. Když budete používat kterýkoli blok until v EdScratchi, musíte robotovi zadat podmínku pomocí vstupního parametru ve tvaru kosočtverce.

1. Otevřete programovací prostředí EdScratch a prohlédněte si různé bloky. Které kategorie bloků podle vás obsahují bloky, které můžete použít k zadání vstupního parametru pro některý z bloků until? Proč si to myslíte?

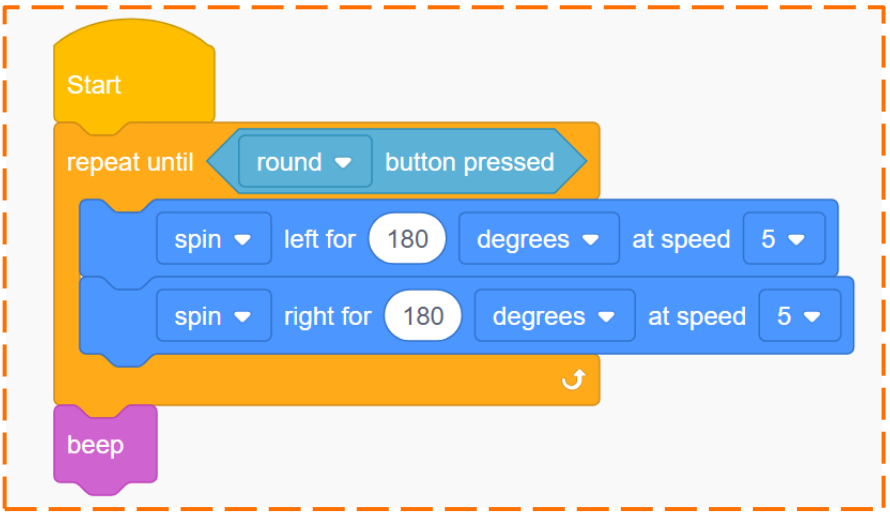
_____

_____

_____

Úkol 1: Opakuj, dokud...

Prohlédněte si následující program v EdScratchi:



Jméno_____

Tento program používá cyklus repeat until (opakuj dokud), který je nekonečným cyklem.

Proč je to tak?

Jakýkoli cyklus s nedefinovaným počtem opakování se nazývá nekonečný cyklus

Příkladem nekonečného cyklu je blok forever v EdScratchi, který je opakován do nekonečna. Dalším příkladem je blok repeat until, protože cyklus probíhá *do* splnění podmínky. Tato podmínka může být splněna po pouhém jednom cyklu, nebo může být splněna po 20 cyklech, ale taky nikdy! Protože nevíme přesně, kolikrát cyklus proběhne, jde o nekonečný cyklus.

Napište v EdScratchi program uvedený na obrázku a stáhněte jej do Edisona. Spusťte program a otestujte, jak funguje.

2. Když spustíte tento program, co musíte udělat, aby Edison zapípal? Proč je to tak?
   Nápověda: prohlédněte si program a sledujte jednotlivé příkazy v sekvenci.

_____

_____

_____

Úkol 2: Událost + podmínka = podmínky události

Jedním z nejčastějších způsobů využití podmíněných příkazů v EdScratchi je stanovení určitých událostí jako podmínek dalších kroků či akcí. Takovýmto podmínkám se říká **podmínka události** *(event condition)*.

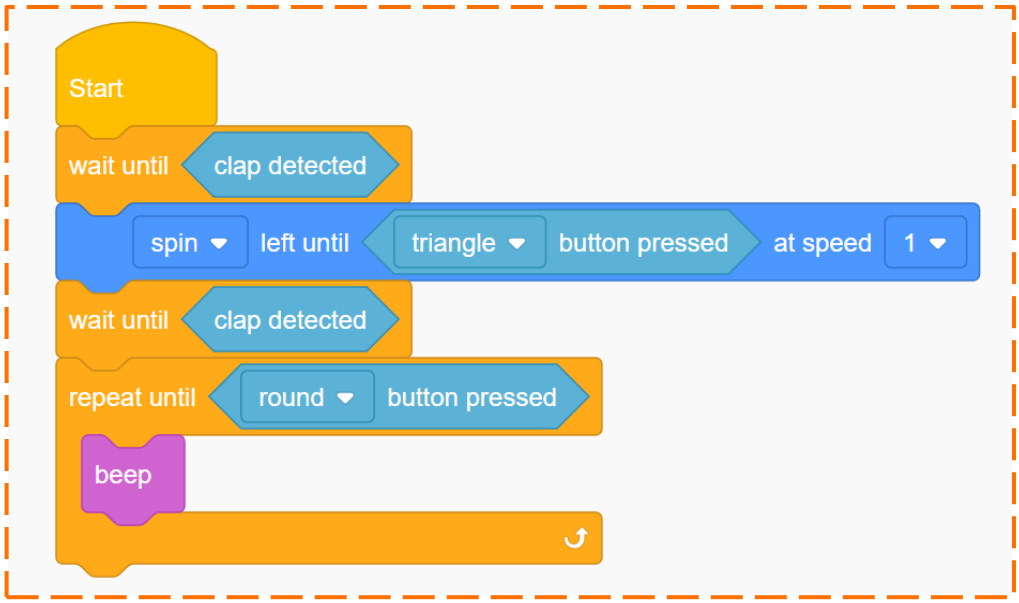
Slovníček pojmů

Pamatujte si ze v programování pojem událost (event) označuje něco, co se stane mimo programový kód ovlivňující běh programu.

Podmínka události (event condition) je podmínka vyžadující konkrétní událost, např. stlačení tlačítka, ke spuštění podmíněného kódu.

Podmínky události můžete v EdScratchi použít s bloky until. Edison bude provádět akci definovanou blokem until, dokud nedojde ke stanovené události. Po proběhnutí této události Edison přejde k dalšímu bloku v programu.

Následující program využívá mnoho bloků s *podmínkou* until s podmínkami událostí:



Napište tento program v EdScratchi a stáhněte jej do Edisona. Spusťte program. Dokážete přimět program, aby provedl každý krok až do úspěšného konce a přepnul robota do režimu standby?

3. Kolik událostí musí nastat po spuštění programu, aby se celý program úspěšně dokončil?

   _____

4. První blok v tomto programu přikazuje robotovi wait until clap detected *(čekej dokud nerozpoznáš tlesknutí)*. Napadá vás situace v reálném životě, v níž by bylo možné využít kód s podmínkou wait until? Jaké zařízení by mohlo využívat program s podmínkou wait until jako prvním krokem? Jaká podmínka by spustila podmíněný kód? Co by program přiměl zařízení udělat, když dojde ke splnění podmínky události?

_____

_____

_____

_____

_____

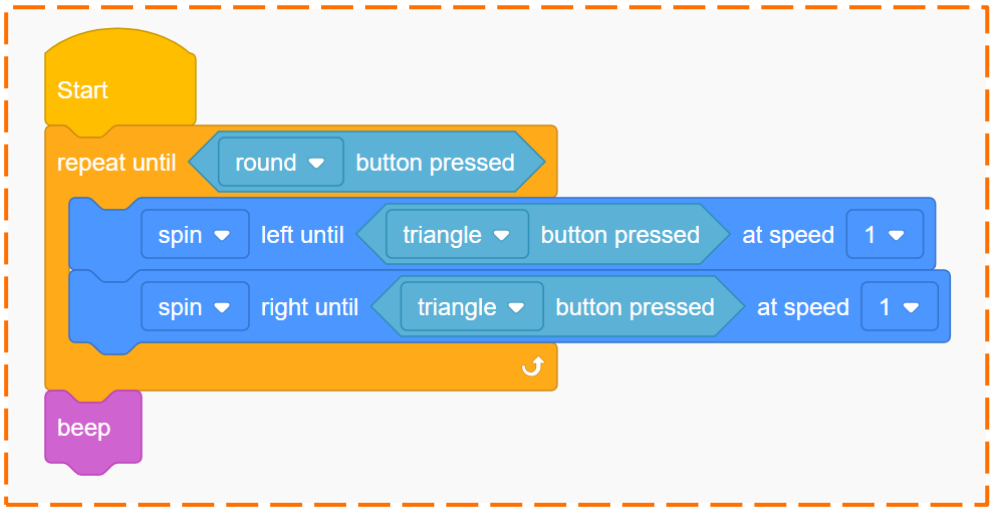
_____

Jméno_____

U4-1.1a Změňte to: Chyba robota, nebo lidská chyba?

Když píšeme program pro Edisona, někdy se zdá, že robot prostě odmítá udělat to, co chceme. Stejně jako u ostatních počítačů je rozsah toho, co Edison dokáže, omezený. Dříve, než obviníte robota z toho, že váš program nefunguje, zeptejte se sami sebe, zda je problém u robota – nebo u člověka.

Prohlédněte si následující program, který využívá tři různé podmíněné příkazy:



Program nefunguje tak, jak by si programátor přál:

„Když po spuštění programu stisknu kulaté tlačítko, robot by se podle mého názoru měl přestat pohybovat, pak zapípat a program by měl skončit. To se neděje! Zkusil jsem dvakrát stisknout trojúhelníkové tlačítko, pak stisknout kulaté tlačítko, ale robot se stále otáčí. Myslím, že robot je pokažený.“

Je robot skutečně pokažený? Jsou v programu nějaké mouchy? Udělal programátor logickou chybu? Nebo je to něčím jiným?

Nezapomeňte

Logické chyby jsou problémy s logikou, neboli způsobem myšlení, v programu. Pokud program nefunguje tak, jak jste očekávali, možná v něm je logická chyba. Edison možná provádí program přesně podle napsaného kódu, avšak ve vašem vnímání program nefunguje.

Mějte na paměti, že počítače nedokáží přemýšlet jako lidé. Proto musíte používat algoritmické myšlení a plánovat, řešit problémy a analyzovat informace stejně jako počítač.

Jméno_____

Co máte dělat

Podívejme se, jestli můžeme programátorovi pomoci zjistit, co se s programem děje, a vysvětlit mu to.

Prvním krokem, který musíte udělat, je spustit si sami program. Tak můžete zjistit, jak funguje, a začít kontrolovat případné chyby. Spuštění programu vám také umožní zkontrolovat programátorovo myšlení a zjistit, zda neudělal nějakou logickou chybu.

Napište program v EdScratchi a stáhněte jej do Edisona. Tento program využívá podmínky událostí, takže je nutné, abyste jej spustili a následně otestovali, zda se robot chová tak, jak byste při každé události očekávali.

Nezapomeňte

Ke spuštění programu stiskněte jednou tlačítko ke spuštění (trojúhelníkové). Tím dojde ke spuštění program. Edison nevnímá tento stisk jako součást programu.

1. Spusťte robota stiskem tlačítka pro spuštění (trojúhelníkového). Co robot provede?

_____

2. Je jeho chování v souladu s vaším očekáváním? Proč ano nebo proč ne?

_____

_____

3. Nyní stiskněte trojúhelníkové tlačítko. Robot se začne otáčet doprava. Proč je tomu tak?

_____

_____

Programátor uvádí dva problémy s funkcí programu:

Problém č. 1:

„Když po spuštění programu stisknu kulaté tlačítko, robot by se podle mého názoru měl přestat pohybovat, pak zapípat a program by měl skončit. To se neděje!“

Problém č. 2:

„Zkusil jsem dvakrát stisknout trojúhelníkové tlačítko, pak stisknout kulaté tlačítko, ale robot se stále otáčí. “

Spusťte opět program v Edisonovi. Sledujte kroky, které popisuje programátor a podívejte se, zda dokážete navodit problémy, se kterými se potýkal programátor. Máte stejné problémy jako on? Myslíte si, že jde o lidskou chybu, nebo chybu robota?

Jméno_____

Co se děje?

Podívejme se na oba problémy jednotlivě.

Problém č. 1:

„Když po spuštění programu stisknu kulaté tlačítko, robot by se podle mého názoru měl přestat pohybovat, pak zapípat a program by měl skončit. To se neděje!"

Jde o lidskou chybu. Programátor udělal logickou chybu při úvaze nad tím, jakou akci by měl program v Edisonovi vyvolat.

Abyste pochopili, v čem chyba spočívá, zkuste znovu spustit program v Edisonovi. Tentokrát stiskněte během programu jednou kulaté tlačítko. Následně stiskněte dvakrát trojúhelníkové tlačítko.

Robot se otočí doleva, pak se otočí doprava, pak zapípá a následně program skončí.

Proč je to tak?

Pamatujte si, že blok *podmínka* repeat until je nekonečný cyklus, který skončí, když je splněna podmínka.

Tento cyklus přikazuje Edisonovi provést každý kód v rámci cyklu v daném pořadí a pak se vrátit na začátek cyklu. Pokud podmínka cyklu **není** splněna, cyklus přikáže Edisonovi opakovat kód uvnitř cyklu. Můžete si to představit tak, že se cyklus Edisona ptá: „bylo stisknuto kulaté tlačítko?" Pokud zní odpověď „ne", odešle cyklus Edisona opět na začátek cyklu. Pokud je odpověď „ano", cyklus odešle Edisona na kód následující po cyklu.

Program kontroluje splnění podmínky cyklu pouze na začátku cyklu, nikoliv když probíhá kód uvnitř cyklu. Robot nejprve musí dokončit všechen kód uvnitř cyklu, pak přejde na začátek cyklu a zkontroluje podmínku.

Náš programátor stiskl kulaté tlačítko, ale nesplnil podmínky uvedené v blocích kódu uvnitř cyklu. Tím došlo k chybě!

4. Vysvětlete vlastními slovy logickou chybu, kterou udělal programátor, který tento program napsal.

_____

_____

_____

_____

_____

Jméno_____

Podívejte se rovněž na druhý problém:

Problém č. 2:

„Zkusil jsem dvakrát stisknout trojúhelníkové tlačítko, pak stisknout kulaté tlačítko, ale robot se stále otáčí."

Jde rovněž o lidskou chybu, ale nikoliv logickou. Problémem je, že v porovnání s roboty jsou lidé trochu pomalí. Je to proto, že roboti procházejí kód opravdu, ale *opravdu* rychle!

Proč je to tak?

Nezapomeňte, že program kontroluje plnění podmínky cyklu na **začátku** cyklu.

Jakmile v tomto programu dojde ke druhému stisknutí trojúhelníkového tlačítka, je ukončen blok spin right until triangle button pressed (otáčet doprava do stisku trojúhelníkového tlačítka) a kód se přesune na začátek, aby zkontroloval, zda nedošlo ke stisku kulatého tlačítka.

To vše se odehraje velmi rychle. Zkontrolování stisknutí kulatého tlačítka trvá kódu méně než 10 milisekund. To je méně než setina sekundy!

Náš programátor stiskl kulaté tlačítko, ale kód již v tu dobu měl zkontrolováno splnění podmínky cyklu a odeslal robota zpět do cyklu. Programátor byl pomalý!

5. Při programování věci dost často nefungují tak, jak bychom chtěli. Někdy to může být velmi frustrující! Zamyslete se nad tím, co byste mohli udělat příště jinak, až napíšete program, který nebude fungovat podle vašich představ. Napište zprávu svému budoucímu já. Navrhněte, jak by šlo problém řešit.

_____

_____

_____

_____

_____

_____

_____

_____

Jméno_____

U4-1.2 Prozkoumejte příkazy if

Pomocí podmíněných příkazů můžete psát programy, které umožní počítači dělat rozhodnutí. Podmíněnými příkazy počítači dáváte přesné pokyny, díky nimž ví, jaké rozhodnutí dělá, a jaké jsou podmínky pro toto rozhodnutí.

Nejčastějším způsobem stanovení podmínek při programování je použití **příkazu if**.

Slovníček pojmů

Příkaz if *(if statement)* je podmíněný příkaz. Jde o část kódu, která závisí na jiném faktoru. Tato část kódu poběží pouze, pokud *(if)* bude splněna daná podmínka. Proto se jí říká příkaz „if"!

Příkazy „if" pravděpodobně používáte v reálném životě k rozhodnutím velmi často, možná aniž byste si toho byli vědomi. Podívejte se na následující příklady:

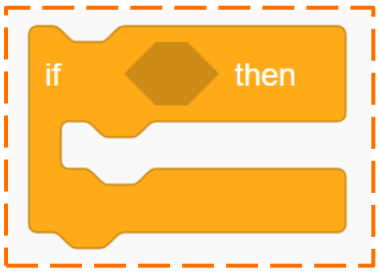
- POKUD je venku zima, TAK si nasadím před odchodem z domova bundu.
- POKUD mám po škole hlad, TAK si dám svačinu.

1. Zamyslete se nad podmíněným příkazem z každodenního života. Zapište jej pomocí vzorce 'pokud__, tak___'.

   POKUD (IF) _____

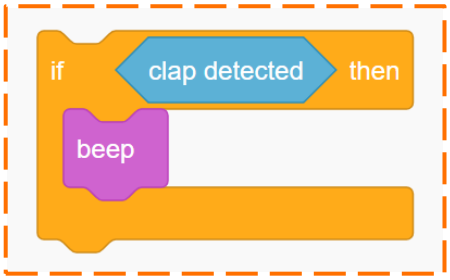
   TAK (THEN) _____

Při psaní kódu se příkazy 'if' řídí stejným vzorcem. Prohlédněte si v EdScratchi blok if:



Vidíte v bloku vzorec „if__, then___" (pokud__, tak___)? Pokud v kódu použijete příkaz „if", přikazujete počítači, že POKUD dojde k naplnění *podmínky,* TAK má provést podmíněnou akci.

Např. IF *clap detected* (detekováno tlesknutí) THEN beep (pípnout):



Musíte počítači rovněž říct, co má dělat, pokud daná podmínka NENÍ splněna. K tomu potřebujete podmíněný příkaz pokud-jinak (**if-else statement**).

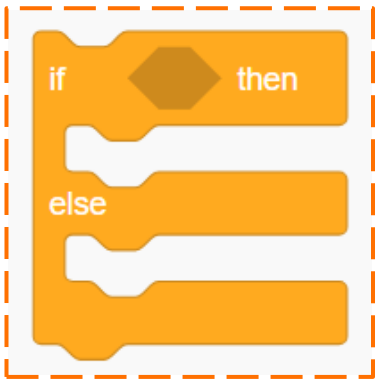
Slovníček pojmů

Příkaz if-else (pokud-jinak) je podmíněný příkaz. Stejně jako u všech jiných podmíněných příkazů tato část kódu závisí na jiném faktoru. Příkaz if-else říká programu, co má dělat, pokud je podmínka splněna, i co má dělat, pokud podmínka NENÍ splněna.

Část „elseL v příkazu if-else říká programu, co dělat, pokud podmínka není splněna.

Příkaz if-else si můžete představit jako místo rozhodnutí v programu. Příkaz if-else programu říká: „pokud je podmínka splněna, udělej věc A. Pokud podmínka není splněna, udělej věc B."

Blok if-else v EdScratchi vypadá takto:



Podobně jako blok if i blok if-else využívá základní vzorec „pokud__, tak___" avšak na rozdíl od bloku if uvádí i to, co má program udělat, když podmínka není splněna. Příkaz if-else umožňuje uskutečňovat podmíněnou volbu:

- POKUD je venku zima, TAK si před odchodem z domu nasadím bundu. JINAK jdu ven v tričku.
- POKUD mám po škole hlad, TAK si dám svačinu. JINAK počkám až na večeři.

Použitím příkazu if-else dochází k rozvětvení programu. Program bude pokračovat buď jedním směrem, nebo jiným.

Jméno_____

Vyzkoušejte si to!

Procvičujte si používání vzorce „if__, then___ else____“ (pokud__, tak___ jinak____) a sledujte, jak se program větví. Při této činnosti používejte Pracovní list U4-1. Tento pracovní list nabízí speciální mapu pokladů, kterou lze vyřešit pouze pomocí příkazů if-else.

Nápověda!

Víte, že se při psaní kódů používají i některé matematické symboly? Při této aktivitě použijete následující znaky:

A = B znamená „A se rovná B“

A > B znamená „A je větší než B“

A < B znamená „A je menší než B“

Řiďte se jednotlivými sadami pokynů a nalezněte všechny poklady.

2. Maršmelounová koule

Začni na startu.

Opakujte třikrát:

IF číslo < 7, THEN jdi doleva. ELSE jdi doprava.

Kde se nachází maršmelounová koule? _____

3. Palačinkový plášť

Začni na startu.

Opakujte třikrát:

IF číslo > 2, THEN jdi doleva. ELSE jdi doprava.

Kde se nachází palačinkový plášť? _____

4. Omeleta z vesmírných vajec

Začni na startu.

Opakujte dvakrát:

IF číslo = 9, THEN jdi doleva. ELSE jdi doprava.

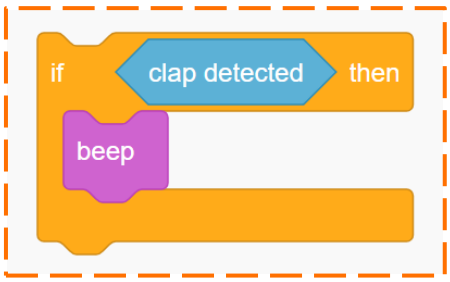
IF číslo < 7, THEN jdi doleva. ELSE jdi doprava.

Kde se nachází Omeleta z vesmírných vajec? _____

Jméno_____

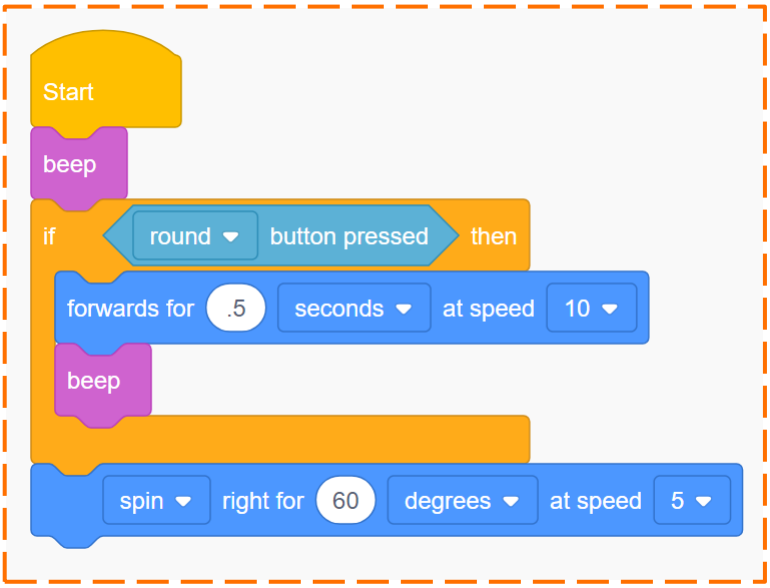
U4-1.3 Prozkoumejte příkazy if a sekvenci

Když v kódu použijete příkaz „if“, říkáte počítači, co má dělat, když nastane daná podmínka. V EdScratchi musíte v bloku if zadat podmínku pomocí vstupního parametru v kosočtvercovém vstupním parametru. Vložením bloku či bloků do „úst“ bloku if můžete robotovi rovněž říct, jakou má podniknout podmíněnou akci:



Co se stane v programu, který využívá blok if, když podmínka **není** splněna?

Prohlédněte si následující program:



1. Co se musí v tomto programu stát, aby podmínka stanovená v bloku if byla splněna?

_____

_____

Napište program v EdScratchi. Stáhněte jej a spusťte v Edisonovi.

2. Co se stalo, když jste program spustili? Běžel podmíněný kód (kód uvnitř bloku if)?

_____

_____

3. Co se na základě vašich zjištění a podle vašeho názoru stane v programu, který používá blok if, když podmínka NEBUDE splněna?

_____

_____

_____

Proč je to tak?

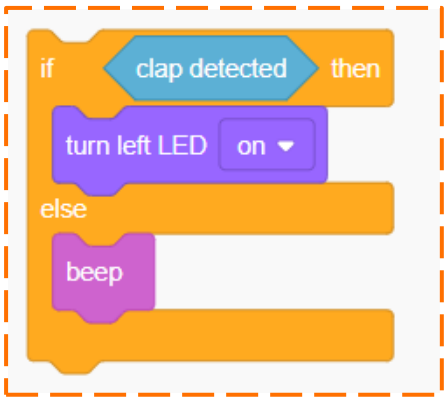
Pamatujte si, že všechny programy procházejí kód krok za krokem v sekvenci, tj. posloupnosti. Když program narazí na if blok, zkontroluje, zda daná podmínka byla splněna. Pokud byla, program spustí kód uvnitř bloku. Pokud podmínka není splněna, program kód v bloku if přeskočí a přejde na další řádek kódu.

Edison postupuje od bloku k dalšímu bloku v kódu velmi rychle. Za méně než 10 milisekund robot dorazí k bloku if a kontroluje, zda je stisknuto kulaté tlačítko. Za méně než setinu sekundy! Je téměř nemožné stisknout kulaté tlačítko včas.

Pokud chcete, aby proběhl podmíněný kód, jaký další blok byste měli do programu přidat, abyste získali čas ke stisknutí kulatého tlačítka?

Podobně jako je tomu u bloku if, když program napsaný v EdScratch narazí na blok if-else, zkontroluje, zda byla daná podmínka splněna. Blok if-else říká robotovi, co dělat, když je daná podmínka splněna, a co dělat, pokud není splněna, takže robot za všech okolností podnikne nějakou akci.

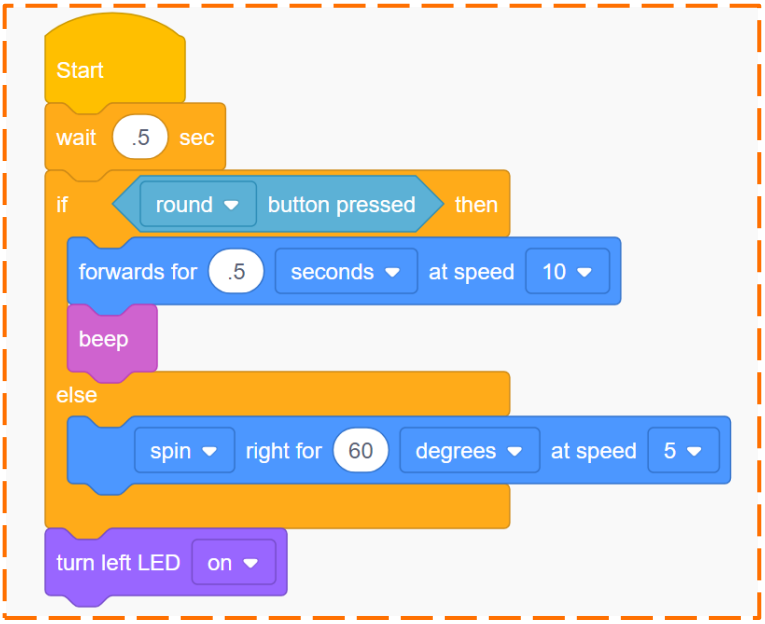
Pokud je podmínka splněna, robot spustí kód uvnitř „if“ části bloku. Pokud podmínka není splněna, robot spustí kód v části ‘else’:



Příkaz if-else způsobuje rozvětvení programu, protože robot může provést pouze jednu z částí kódu – buď „if“ nebo „else“ – ale nikoliv obě. Jakmile robot dokončí „if“ kód nebo „else“ kód, přejde na další řádek programu.

Jméno_____

Prohlédněte si následující program v EdScratchi:



4. Pokud byste tento program spustili v Edisonovi a robot by NEROZPOZNAL stisknutí kulatého tlačítka, pípl by robot? Proč ano nebo proč ne?

_____

_____

_____

_____

5. Když program běží, které kroky proběhnou vždy, bez ohledu na to, zda robot detekuje stisknutí kulatého tlačítka? *Nápověda: Projděte program v sekvenci. Které tři akce se odehrají vždy, bez ohledu na okolnosti?*

_____

_____

_____

Jméno_____

U4-1.4 Prozkoumejte spojené a vnořené if příkazy

Podmíněné příkazy jsou důležitou součástí kódu ve všech programovacích jazycích, včetně EdScratch. Použití podmíněných příkazů, jako jsou „if“ příkazy v EdScratchi, vám umožňuje napsat Edisonovi nejrůznější zajímavé programy.

Všechny podmíněné bloky, včetně bloků if a if-else, se v EdScratchi nacházejí v kategorii Control (Řízení). Je to proto, že jak podmíněné příkazy, tak cykly, vám umožňují řídit tok programu. Blok if a if-else mají s cykly společného ještě něco dalšího: je možné je v programech spojovat nebo vnořovat.

Proč je to tak?

Podobně, stejně jako můžete vnořit cyklický blok do jiného cyklického bloku, můžete vnořovat bloky if a if-else, jak vzájemně, tak do cyklů!

Úkol 1: Co se stane tentokrát?

Když program obsahuje více cyklů nebo podmíněných bloků, ať již spojených nebo vnořených, může být sledování toku programu náročné. Abyste pochopili, co program přesně udělá, musíte vnímat každou akci jednotlivě tak, jak nastane v posloupnosti.

Nezapomeňte

Všechny programy v EdScratchi fungují v zásadě stejně. Program říká robotovi, aby začal od horního bloku a pak plnil každou akci krok za krokem. Jakmile je daný blok splněn, program se přesune na následující blok.

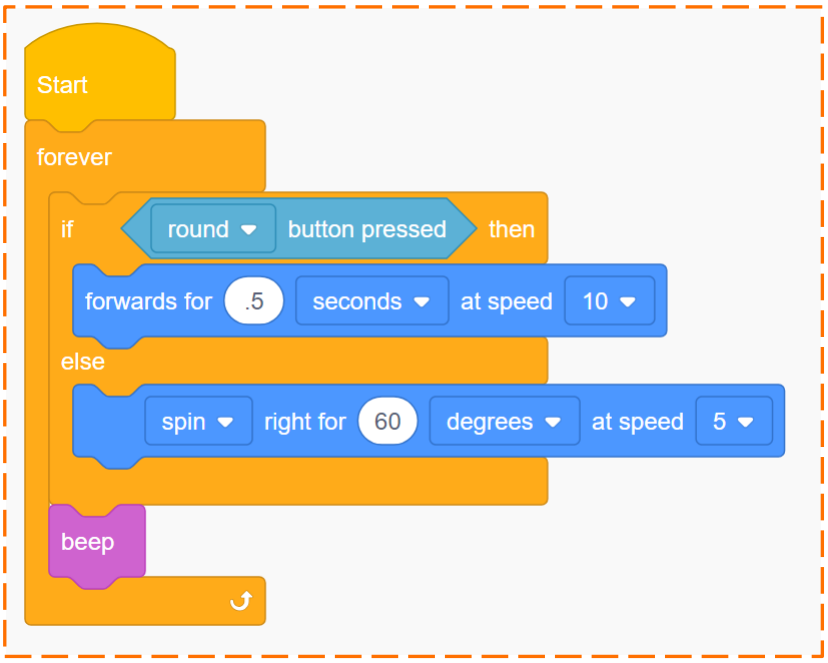
Pokud blok obsahuje podmíněný kód, program nejprve ověří splnění dané podmínky. Výsledek kontroly plnění podmínky určuje, co program provede v dalším kroku.

Ačkoliv cykly a podmíněné příkazy řídí tok programu, všechny programy stále postupují v sekvenci. Když se díváte na program s vnořenými cykly a podmíněnými příkazy, mějte na paměti, že i tento program probíhá krok za krokem. To vám pomůže lépe sledovat, co se v programu odehrává.

Jméno_____

Prohlédněte si následující programy odpovězte na otázky.

Program 1:



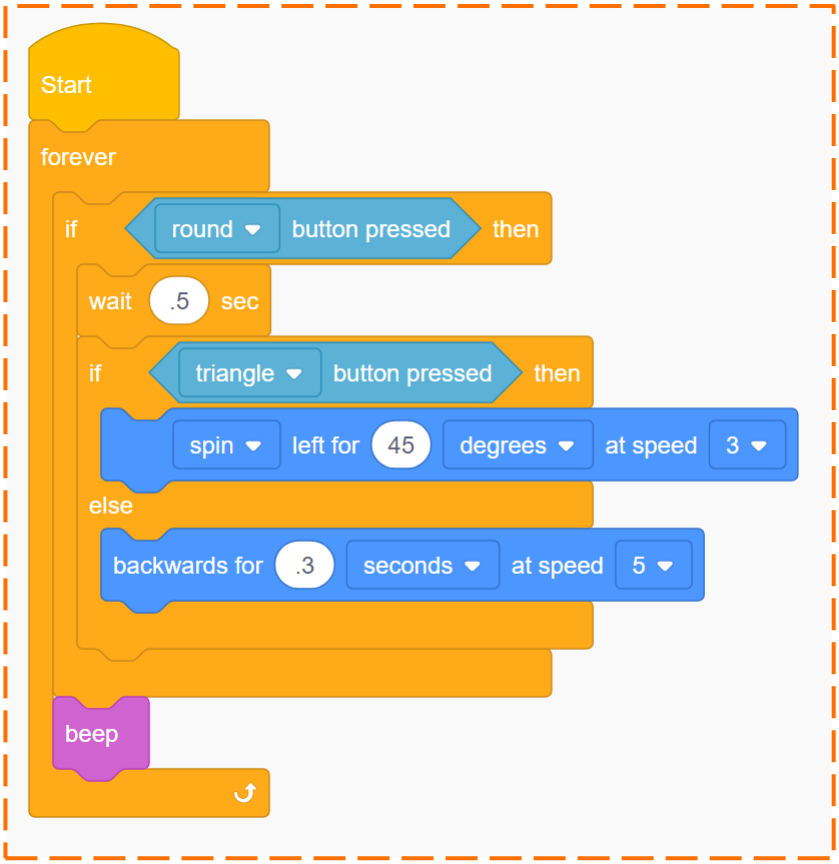
1. Pokud spustíte program 1, ale nikdy nestisknete kulaté tlačítko, co se stane? Proč?

_____
_____
_____
_____
_____
_____
_____

2. Pokud spustíte program 2, ale nikdy nestisknete kulaté tlačítko, co se stane? Proč?

_____
_____
_____
_____
_____
_____
_____

Program 2:



3. Pokud spustíte program 2, co musíte udělat, aby robot jel dozadu? Proč?

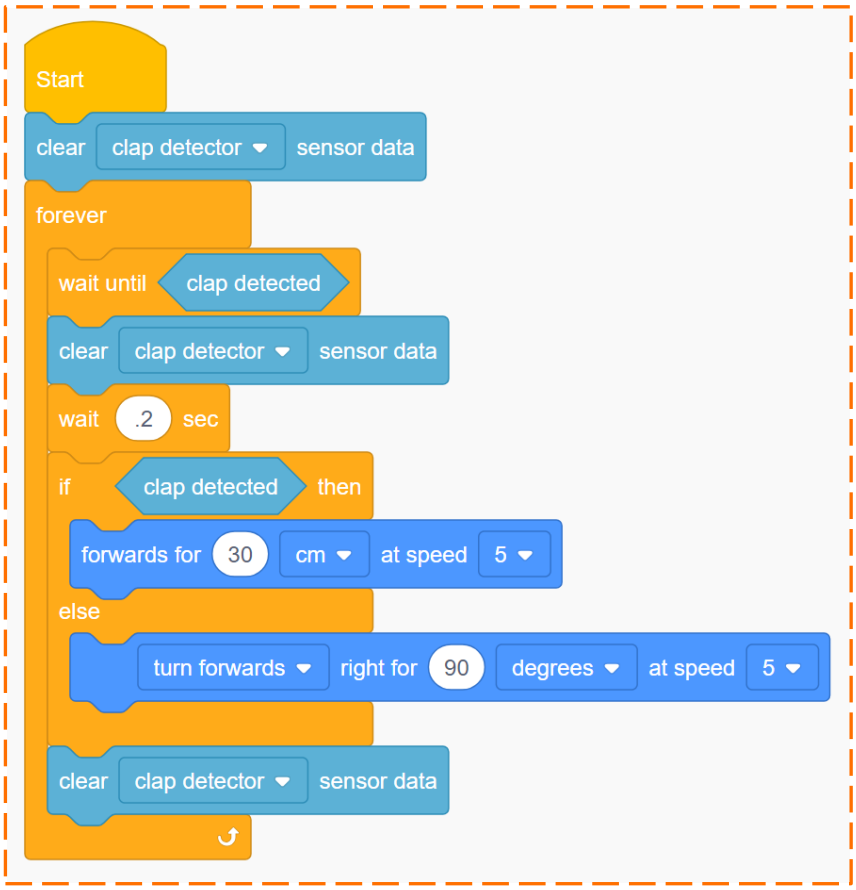
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

Vyberte si jeden z těchto programů a napište jej v EdScratchi. Stáhněte jej a otestujte v Edisonovi. Experimentujte a podívejte se, jak programy s vnořenými „if“ příkazy fungují.

Úkol 2: Tleskáním ovládaný pohyb

Vnořením bloku if-else do bloku forever (navždy) můžeme vytvořit pro Edisona program založený na tleskáním ovládaném pohybu. V tomto programu Edison čeká, až ropozná tlesknutí, a pak buď jede dopředu, nebo zatočí o 90°, podle toho, zda detekuje jedno nebo dvě tlesknutí.

Prohlédněte si tento program pro tleskáním ovládaný pohyb:



Tento program využívá speciální blok z kategorie Sensing (Senzory) zvaný clear sensor data (vymaž data senzorů).

Prohlédněte si program pro tleskáním ovládaný pohyb. Dokážete sledovat tok kódu?

Napište program pro tleskáním ovládaný pohyb v EdScratchi.

Nápověda!

Nezapomeňte použít komentáře! Přidání kvalitního komentáře vám pomáhá sledovat, co se má v programu odehrávat. To se hodí především u programů s vnořenými cykly a podmíněnými příkazy!

Stáhněte program do Edisona a spusťte jej. Experimentujte a zjistěte, jak Edison díky programu reaguje na tlesknutí.

4. V programu pro tleskáním ovládaný pohyb je vnořen blok if-else do bloku forever. Proč tomu podle vás tak je? Co by se stalo, kdybyste nepoužili blok forever?

Proč je to tak?

Vzpomeňte si, že Edison je vybaven různými senzory, včetně zařízení, které umožňuje robotovi detekovat zvuky jako je tleskání. Tyto senzory při detekci konkrétních událostí generují data. Některá z těchto dat ze senzorů jsou uložena v Edisonově paměti. Tato uložená data někdy mohou způsobit problémy, protože robot reaguje na starou událost, kterou si stále „pamatuje".

Když Edison kontroluje, zda byla podmínka splněna, tak pokud existují uložená data, bude si robot myslet, že podmínka byla splněna, i když tomu tak není! Z toho důvodu je při psaní kódu dobré vymazat data ze senzoru. Obzvláště důležité to je, když používáte události ze senzoru v podmíněných příkazech vnořených do cyklů. Potřebujete zamezit tomu, aby data z předchozího cyklu ovlivnila následující cyklus!

Rovněž je vhodné vymazat data na začátku programu pro případ, že by měl robot uložena stará data z předchozího programu.

U4-2.1 Prozkoumejte pseudokód

Vytvoření kvalitních počítačových programů vyžaduje více než jen napsání kódu. Musíte rovněž umět řešit problémy případě, že něco v programu nefunguje. Další důležitou a užitečnou dovedností při programování je schopnost programy naplánovat.

Podobně, jako můžete naplánovat příběh na storyboardu nebo si udělat osnovu při psaní eseje, mohou programátoři využít **pseudokódu** jakožto nástroje k plánování programů.

Slovníček pojmů

Pseudokód je způsob, jak napsat program v jednoduchém, snadno čitelném formátu. Namísto toho, aby se staral o syntaxi, popisuje pseudokód běžnými slovy, co program udělá.

Pseudokód vypadá trochu jako zjednodušený programovací jazyk, ale není založen na žádném konkrétním programovacím jazyce. Proto lze pseudokód použít k plánování programů v jakémkoli kódovacím jazyce.

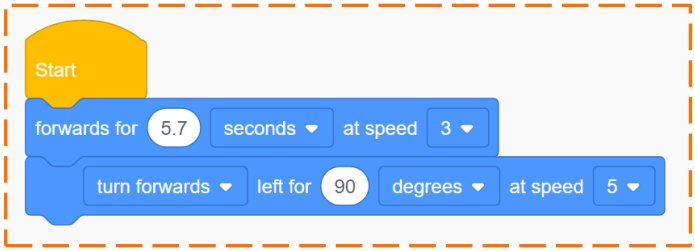
Když budete plánovat program pomocí pseudokódu, nemusíte si dělat starosti s vypisováním všech podrobností nebo s tím, jak přesně bude program po dokončení vypadat. Vašim cílem je napsat jednoduchý plán, který vám usnadní sledování toku programu.

Takto například vypadá pseudokód programu pro jízdu Edisona:

```
jeď dopředu
zatoč doleva
```

Pseudokód nastíní základní plán, ale nezahrnuje všechny podrobnosti. Ty se vypracují později při programování kódu.

Pseudokód je tak rozepsán do následujícího programu:



Vidíte, jak byl základní plán nastíněný v pseudokódu převeden do programu?

Vytvoření plánu v pseudokódu usnadní vypracování struktury vašeho programu. Poté jej můžete upravit a při psaní kódu doplnit podrobnosti.

Aby byl pseudokód snadno čitelný, měli byste jej udržovat čistý a přehledný. Pseudokód pište tak, aby byl jeho tok stejný jako u kódu programovaného v EdScratchi. To znamená, že byste měli psát každý krok jeden po druhém, řádek po řádku.

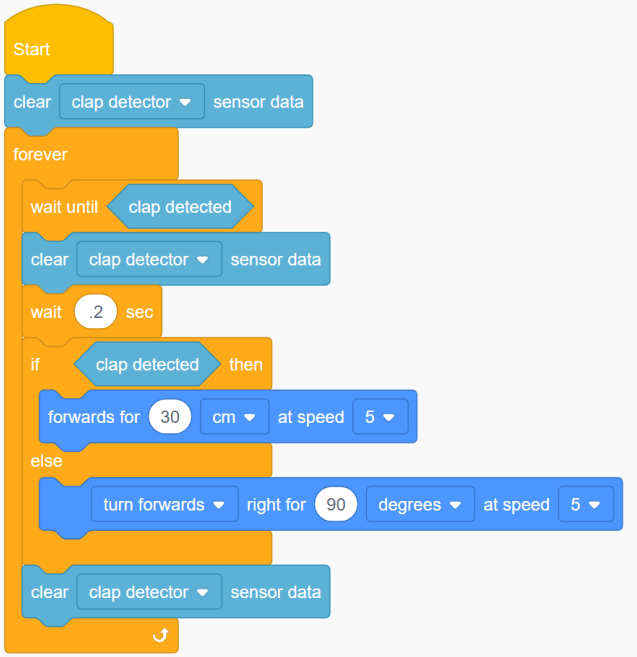
Použití pseudokódu je zvláště užitečné při plánování programů, které používají řídicí struktury, jako jsou cykly nebo podmíněné příkazy. Měli byste odsazovat akce, které jsou uvnitř cyklů nebo podmíněných příkazů. Tím naznačíte, že tento kód je uvnitř cyklu nebo „if“ příkazu. Toto rozčlenění pseudokódu usnadňuje jeho pochopení.

Následuje příklad pseudokódu popisujícího program pro pohyb ovládaný tleskáním a příslušný program v EdScratchi:

Pseudokód:

```
vymaž data o rozpoznání tlesknutí
nekonečný cyklus
        čekej do tlesknutí
        vymaž data o rozpoznání tlesknutí
        čekej
        pokud tlesknutí
                jeď dopředu
        jinak
                zatoč doprava
        vymaž data o rozpoznání tlesknutí
```

EdScratch



Vidíte, jak jsou jednotlivé akce v pseudokódu a programu řazeny?

Nezapomeňte

Pseudokód by měl vždy nastínit základní plán, ale nemusí zahrnovat všechny podrobnosti. Kolik podrobností uvedete, záleží na vás a může se to lišit v závislosti na programu.

Vyzkoušejte si to!

Vyzkoušejte si čtení pokynů z pseudokódu. Otevřete si Pracovní list U4-2, řiďte se pokyny z pseudokódu a odpovězte na otázky.

Jméno_____

1. Přečtěte si pseudokód. Kde program skončí? _____

```
začni na C směrem na východ
vpřed až po jídlo
doleva o 90 stupňů
opakuj 6krát
        vpřed 1
        pokud zvíře
                doleva o 90 stupňů
```

2. Přečtěte si pseudokód. Kde program skončí? _____

```
Začni na H směrem na východ
opakuj 3krát
        vpřed 1
        pokud žijící věc
                doprava o 90 stupňů
        jinak
        doleva o 90 stupňů
zpět 1
```

3. Přečtěte si pseudokód. Kde program skončí? _____

```
Začni na D směrem na západ
opakuj 3krát
        dopředu 3
        pokud zvíře
                doleva o 90 stupňů
        jinak
                pokud jídlo
                        doprava o 180
                stupňů
opakuj 3krát
        dopředu do písmene
        doprava o 90 stupňů
zpět do čísla
```

Nápověda!

Zamyslete se nad tím, jak fungují podmíněné příkazy until, if a if-else.

Spustí program podmíněný kód v bloku if, pokud podmínka **není** splněna? Co se místo toho stane? A jak je tomu u bloku if-else?

Ujistěte se, že čtete pseudokód stejně, jako by jej četl počítač!

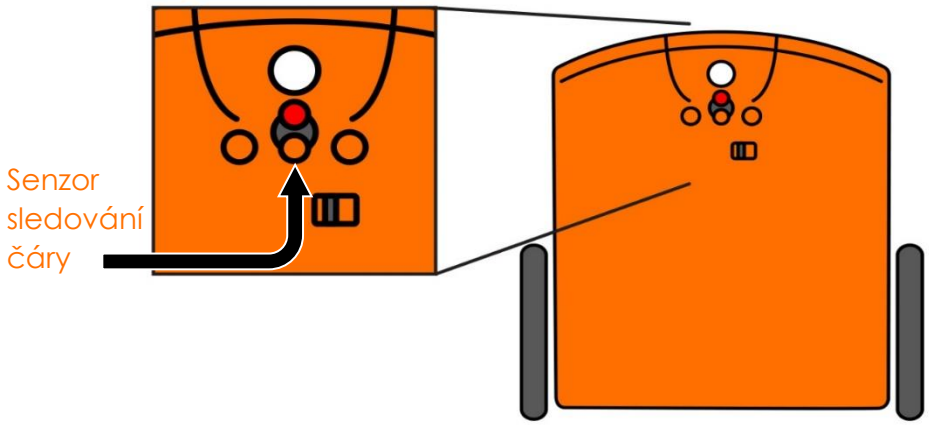
Jméno_____

U4-2.2 Prozkoumejte Edisonův senzor pro sledování čáry

Robot Edison je vybaven různými senzory, které dokáží sledovat a detekovat různé věci. Jedním z těchto senzorů je senzor pro sledování čáry.

Úkol 1: Seznamte se s Edisonovým senzorem pro sledování čáry

Senzor pro sledování čáry umožňuje Edisonovi rozpoznat rozdíl mezi tmavým a světlým povrchem. Senzor je umístěn ve spodní části Edisona, blízko vypínače.



Senzor pro sledování čáry se skládá ze dvou částí: červené LEDky a světelného senzoru. Prohlédněte si senzor pro sledování čáry na Edisonovi. Vidíte obě části senzoru?

Senzor pro sledování čáry funguje tak, že osvětluje světlem z červené LEDky povrch pod robotem. Světelný senzor pak měří, kolik z tohoto světla se odráží od povrchu. Edison ukládá hodnotu odraženého světla jako odečet světla. Čím více světla se odráží zpět k Edisonovi, tím vyšší je hodnota světla.

Odráží více světla zpět k Edisonovi bílý povrch nebo černý povrch? Použijte Pracovní list U4-3 a otestujte, jaký povrch odráží více světla zpět k Edisonovi – zda bílý nebo černý.

Zapněte Edisona a stiskněte dvakrát kulaté tlačítko; tím zapnete senzor pro sledování čáry. Lehce nadzvedněte Edisona nad papír a prohlédněte si kruh osvětlený světlem z LEDky. Srovnejte, jak světle tento bod vypadá na černém povrchu a jak na bílém povrchu.

Nápověda!

Čím více světla se odráží, tím jasnější bude bod na povrchu níže.

1. Který povrch podle vás odráží více světla zpět směrem k Edisonovi, bílý nebo černý? Proč si to myslíte?

_____

_____

_____

Měřením toho, kolik odraženého světla se odráží od povrchu pod robotem, umožňuje senzor pro sledování čáry robotovi „vidět“ rozdíl mezi tmavými a světlými povrchy. Edison však nevidí barvy jako člověk.

Robot je schopen pouze rozeznat, zda je povrch reflexní či nereflexní. Reflexní povrch odrazí zpět hodně světla z červené LED, zatímco nereflexní povrch odrazí zpět velmi málo světla.

Edison vnímá bílé povrchy jako reflexní a černé povrchy jako nereflexní. Co ostatní barvy?

2. Bude Edison považovat červený povrch za reflexní nebo nereflexní? A co modrý povrch? Nebo zelený? Použijte Pracovní list U4-3 a vyzkoušejte všechny tři barvy pomocí červené LED pro sledování čáry. *Nápověda:* pokud vidíte světlý bod podobně, jako je tomu u bílého povrchu, odráží se hodně světla, a robot bude danou barvu vnímat jako „reflexní".

Barva	Reflexní či nereflexní?
Červená	
Modrá	
Zelená	

Úkol 2: Jeď až k černé čáře

Edisonovy senzory můžeme použít k vytváření vstupů v programech v EdScratchi, abychom Edisonovi řekli, aby hledal různé typy událostí a instruovali robota, co má dělat, když tyto události nastanou.

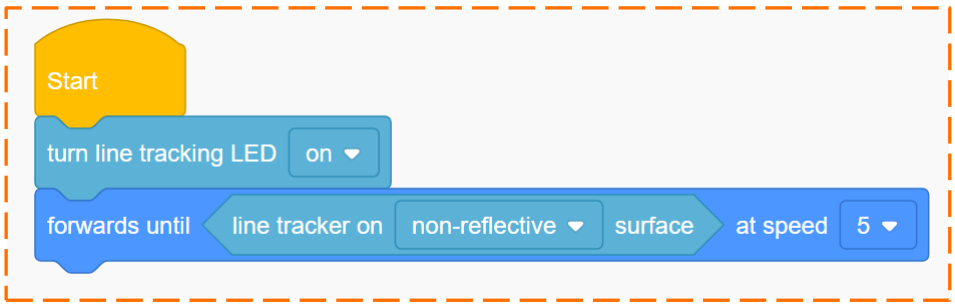
Nezapomeňte

Vstupy jsou informace a pokyny, které dáváte počítači. Když píšete program pro Edisona, pomocí vstupů říkáte robotovi, co chcete. Edisonův mikročip poté zpracovává informace a řekne robotovi, jaký má být výstup v rámci cyklu vstup-zpracování-výstup.

Událost je věc, která nastane mimo kód programu a ovlivňuje běh programu. Událostí může být např. stisknutí tlačítka nebo přenos informace ze senzoru.

Zkuste použít senzor pro sledování čáry v programu, který řekne Edisonovi, aby jel, dokud nenarazí na černou čáru. K napsání tohoto programu budete potřebovat použít bloky z kategorie Sensing (Senzory) v EdScratchi.

Prohlédněte si následující program:



První blok kódu v programu zapíná sledovací LED. Kdykoli chcete použít v programu senzor pro sledování čáry, musíte ji zapnout.

Proč je to tak?

Některé z Edisonových senzorů jsou vždy zapnuté a průběžně kontrolují události. Příkladem takového „vždy zapnutého“ senzoru je zvukový senzor, který dokáže detekovat tleskání.

Jiné senzory, jako je např. Edisonův senzor pro sledování čáry, jsou ve výchozím stavu vypnuty. Abyste je zapnuli, musíte do svého programu přidat příslušný kód. Nestačí však pouze zapnout LED pro sledování čáry. Potřebujete rovněž kód, kterým Edisonovi řeknete, jakou událost má hledat (reflexní povrch nebo nereflexní povrch) a co dělat, pokud tato událost nastane.

Napište program v EdScratchi a otestujete jej v Edisonovi s použitím Pracovního listu U4-3. Zarovnejte svého robota na obrys na Pracovním listu směřující k černé čáře a spusťte program. Zastaví se Edison u černé čáry?

Mini výzva!

Zajistí váš program zastavení Edisona na barevných čárách i a černé čáře a Pracovním listu U4-3? Proč ano, nebo proč ne?

Zamyslete se nad tím, zda program způsobí zastavení Edisona na každé z jednotlivých barev, a pak ověřte, zda byla vaše předpověď správná!

Jméno_____

U4-2.3 Prozkoumejte algoritmy

Edisonův senzor pro sledování čáry dokáže zjistit, zda je povrch pod robotem reflexní nebo nereflexní. Tento senzor můžete použít k tomu, aby se Edison choval různými způsoby, například jel, dokud nezjistí černou čáru, a pak se zastavil. Pomocí tohoto senzoru můžete také naprogramovat Edisona, aby sledoval černou čáru, i když nevíte, jak tato černá čára vypadá. Nejprve musíte vytvořit **algoritmus**.

Slovníček pojmů

Algoritmus je široká sada instrukcí k řešení souboru problémů. Algoritmus stanoví proces nebo soubor pravidel, která mají být dodržena k vyřešení jakéhokoli problému v sadě.

Počítačové programy často používají algoritmy, ale programy a algoritmy nejsou totéž. Pamatujte, že počítačový program je soubor pokynů, které říkají počítači, aby provedl konkrétní úkol. Algoritmus stanoví logiku pro řešení celé sady problémů, nejen pro jeden konkrétní úkol. Můžete napsat počítačový program, který používá algoritmus, ale ne všechny počítačové programy jsou algoritmy.

Algoritmy jsou opravdu užitečné, protože použití algoritmu umožňuje člověku nebo počítači vyřešit celou sadu problémů, i když neznáte všechny podrobnosti.

Proč je to tak?

Řekněme, že chcete naučit své kamarády, jak upéct ovocný koláč. Když víte, že všichni mají po ruce jablka, stačí napsat jeden recept na jablečný koláč.

Všichni ale nemusí mít jablka po ruce. Co když má jeden kamarád borůvky, jiný třeba třešně a třetí jablka? Pak se všichni nemohou řídit receptem na jablečný koláč. Budete muset napsat samostatný recept pro každé jednotlivé ovoce.

A co když nevíte, jaké ovoce kdo zrovna má? Jak byste mohli své kamarády naučit péct ovocný koláč?

Bez ohledu na to, jaké ovoce zrovna mají, musí všichni vaši kamarádi dodržovat stejné základní pokyny: připravit těsto, naplnit koláč ovocem a upéct koláč.

Tato nová sada instrukcí je příkladem algoritmu.

Jméno_____

V počítačovém programování často chceme vytvořit pokyny, které by mohl počítač používat k vyřešení celé řady problémů. Pomocí algoritmu můžeme napsat program, který umožní počítači vyřešit jakýkoli problém v dané sadě. Bez algoritmu bychom museli pro každý jednotlivý problém psát nový program.

Úkol 1: Sledování černé čáry

Už víte, že můžete použít Edisonův senzor pro sledování čáry k detekci černých (nereflexních) a bílých (reflexních) povrchů. Můžeme vytvořit algoritmus, který pomocí tohoto senzoru přiměje Edisona sledovat jakoukoli černou čáru.

Proč je to tak?

Řekněme, že nakreslíte černou čáru, kterou má Edison sledovat. Chcete-li, aby Edison sledoval vaši čáru, můžete napsat program, díky němuž Edison pojede přesně po čáře. Pokud však nakreslíte novou čáru, budete muset napsat zcela nový program.

Místo toho můžete vytvořit algoritmus.

Tento algoritmus vyřeší sadu problémů: „sleduj jakoukoli černou čáru“. Jakákoli konkrétní čára, kterou Edisonovi nakreslíte, bude novým problémem v rámci této sady.

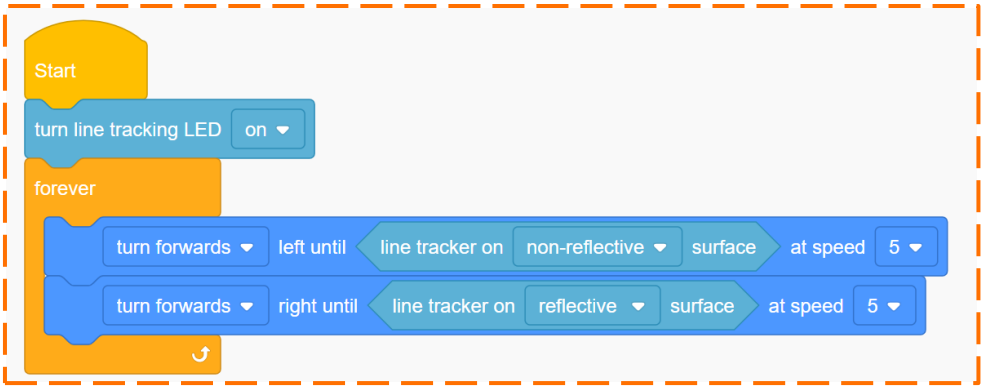
Pomocí algoritmu pro vedení logiky pak můžete napsat program, který bude fungovat pro všechny problémy v sadě. Tímto způsobem není potřeba tvořit nový program pro každý nový problém.

Algoritmus můžete naplánovat pomocí pseudokódu, stejně jako při plánování programu. Zde je pseudokód algoritmu, který umožní Edisonovi sledovat jakoukoli černou čáru:

```
zapni senzor sledování čáry
nekonečný cyklus
        jeď dopředu, dokud robot nenarazí na černou plochu
        jeď dopředu, dokud robot nenarazí na bílou plochu
```

Tento algoritmus říká, že by robot měl jet dopředu doleva, dokud senzor pro sledování čáry nenarazí na nereflexní (černý) povrch. Poté by měl robot jet dopředu doprava, dokud senzor pro sledování čáry nenarazí na reflexní (bílý) povrch. Tyto kroky by měl robot opakovat stále dokola.

Algoritmus můžete použít k napsání programu v EdScratchi:



Vidíte, jak je logika algoritmu zapsána v programu?

Napište program pro sledování čáry v EdScratchi a stáhněte jej do svého Edisona. Otestujte jej na Pracovním listu U4-4. Nezapomeňte spustit Edisona se senzorem pro sledování čáry umístěným na bílém povrchu, nikoliv přímo na černé čáře.

1. Jak se robot po spuštění programu pohybuje? Prohlédněte si program a zamyslete se nad logikou algoritmu. Proč se robot pohybuje právě takto?

_____

_____

_____

_____

Úkol 2: Sleduj jinou černou čáru

Program, který jste napsali, využívá algoritmus, jehož cílem je vyřešit jakýkoli problém v sadě problémů „sleduj jakoukoli černou čáru“. To znamená, že tento program by měl Edisonovi umožnit jet podle jakékoli černé čáry!

Proveďte vlastní test. Použijte černý fix na bílém papíře nebo vytvořte čáru pomocí černé pásky na podlaze nebo na stole. Spusťte program v Edisonovi a vyzkoušejte jej na své čáře. Dokáže Edison sledovat vaši čáru?

2. Dokázal Edison sledovat vaši čáru? Pokud jste se setkali s problémy, popište je. Co podle vás tyto problémy způsobilo?

_____

_____

_____

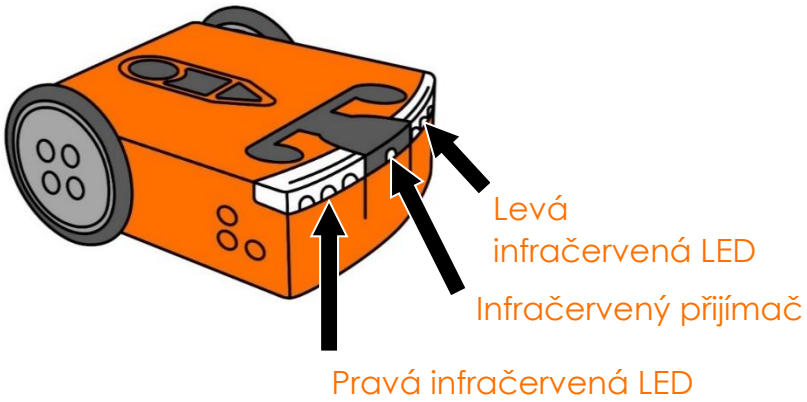
Jméno_____

U4-2.4 Prozkoumejte rozpoznávání překážek

Roboti Edison jsou vybaveny různými senzory, které mohou detekovat různé věci. Jedním z těchto senzorů je infračervený světelný senzor, které můžeme použít k detekci překážek.

Úkol 1: Seznamte se s Edisonovým infračerveným senzorem

Edisonův senzor infračerveného světla je snímač, který umožňuje Edisonovi vyzařovat a detekovat infračervené světlo. Senzor se skládá ze tří částí, které jsou umístěny přes přední část Edisona: dvě infračervené LED (jedna vpravo a jedna vlevo) a infračervený přijímač uprostřed. Prohlédněte si svého Edisona. Vidíte různé části senzoru?



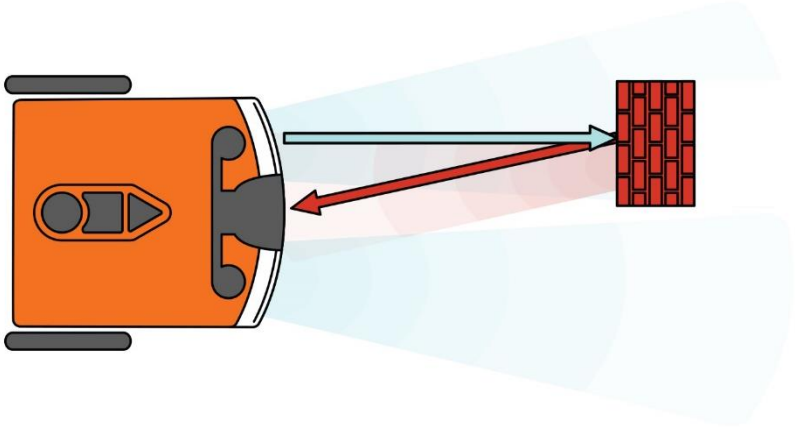
Podobně jako Edisonovy dvě červené LEDky i dvě infračervené LED mohou vyzařovat světlo. Protože však jde o infračervené světlo, neuvidíte jej.

Proč je to tak?

Existuje široká škála (spektrum) světla. Lidé mohou vidět pouze část spektra, ne celé. Infračervené světlo (které se také nazývá IR světlo) není pro člověka viditelné.

I když jej nevidíme, lidé používají infračervené záření velmi často. Infračervené světlo se používá např. v dálkovém ovládání televizoru. Pomocí něj dálkový ovladač řekne televizoru, aby změnil kanál nebo zvýšil hlasitost!

Na rozdíl od lidí dokáže Edison detekovat infračervené světlo pomocí infračerveného přijímače na přední straně. Edisonův infračervený senzor lze použít např. k detekci překážek. Edison může vyzařovat infračervené světlo ze dvou infračervených LED. Pokud infračervené světlo narazí na překážku, jako je např. zeď, odrazí se světlo zpět k Edisonovi. Odražené světlo bude detekováno Edisonovým infračerveným přijímačem, který robota informuje, že se před ním nachází překážka. V závislosti na tom, kde se překážka nachází,



se infračervené světlo odrazí od levé LED, pravé LED nebo od obou. Odražené světlo říká robotovi, kde je překážka umístěna.

Úkol 2: Dopředu až k překážce

Edisonovy senzory můžeme použít k vytváření vstupů v programech v EdScratchi. Pomocí nich Edisonovi řekneme, že má hledat různé typy událostí, a dáme mu pokyny, co má dělat, když tyto události nastanou.

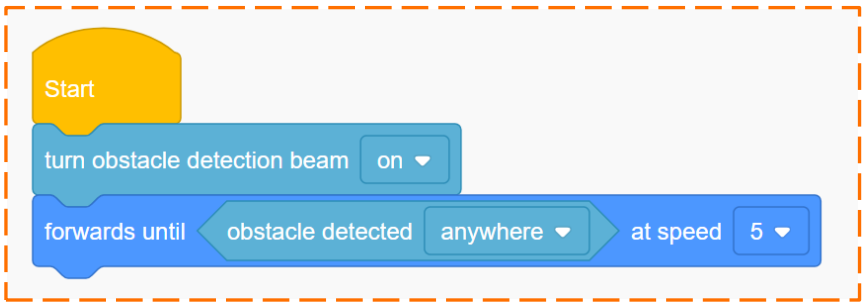
Nezapomeňte

Vstupy jsou informace a pokyny, které dáváte počítači. Když píšete program pro Edisona, říkáte robotovi, co chcete, pomocí vstupů. Edisonův mikročip poté zpracovává informace a řekne robotovi, jaký má být výstup v rámci cyklu vstup-zpracování-výstup.

Událost je věc, která nastane mimo kód programu a ovlivňuje běh programu. Událostí může být např. stisknutí tlačítka nebo přenos informace ze senzoru.

Zkuste použít infračervený senzor v programu, který řekne Edisonovi, aby jel, dokud nezjistí překážku. Chcete-li napsat tento program, budete muset použít bloky z kategorie Sensing v EdScratchi.

Prohlédněte si následující program:



První blok kódu v tomto programu zapne paprsek detekce překážek.

Napište program v EdScratchi. Stáhněte program do Edisona a vyzkoušejte jej. K uskutečnění testu budete potřebovat nějakou překážku.

Některé překážky Edison dokáže rozpoznat lépe než jiné. Pokud je překážka příliš malá nebo neodráží dostatek infračerveného světla, Edison ji nedokáže detekovat. Zvolte objekt, který je neprůhledný, ale není příliš

Nápověda!

Pokud robot nedokáže rozpoznat překážku, zkuste jiný předmět. Pokud stále překážku nerozezná, bude pravděpodobně nutné provést kalibraci detekce překážky. Požádejte svého učitele o speciální čárový kód pro kalibraci detekce překážek.

tmavý (nevybírejte černé předměty) a je alespoň tak vysoký jako Edison. Pro tento program je dobrou překážkou např. stěna v místnosti.

Otestujte program a podívejte se, jak funguje.

Proč je to tak?

Některé z Edisonových senzorů jsou vždy zapnuté a kontrolují události. Infračervený přijímač je zapnutý vždy, ale infračervené LED diody nejsou.

Aby Edison detekoval překážky, musíte do svého programu vložit kód, který zapne paprsek pro detekci překážek. Tím se rozsvítí infračervené LED diody a infračervený přijímač dostane pokyn, aby hledal odražené infračervené světlo odrážející se zpět k robotovi. Pouhé zapnutí paprsku pro detekci překážek však nestačí. Potřebujete také kód, který sdělí senzoru, jakou událost má kontrolovat (jinými slovy, kde by měl hledat překážku) a co dělat, pokud je tato událost detekována.

Úkol 3: Detekuj a vyhni se

Místo toho, aby pouze zastavil, když detekuje překážku, můžete Edisona přimět, aby se detekované překážce vyhnul a pokračoval v jízdě. K tomu musíte vytvořit algoritmus řešící sadu problémů: „detekuj jakoukoli překážku a vyhni se jí“.

1. Pomocí pseudokódu napište algoritmus k detekci a vyhnutí se překážce.

_____

_____

_____

_____

_____

_____

Jméno_____

Až budete mít algoritmus vytvořený, napište v EdScratchi pro Edisona program s využitím logiky z tohoto algoritmu. Stáhněte si program a vyzkoušejte jej v Edisonovi pomocí několika překážek.

2. S problémy se při psaní programů setkávají i profesionální programátoři. Popište, na jaký problém jste narazili při tvorbě algoritmu k detekci a vyhnutí se překážce nebo při psaní programu. Jak jste tento problém vyřešili?

_____

_____

_____

_____

_____

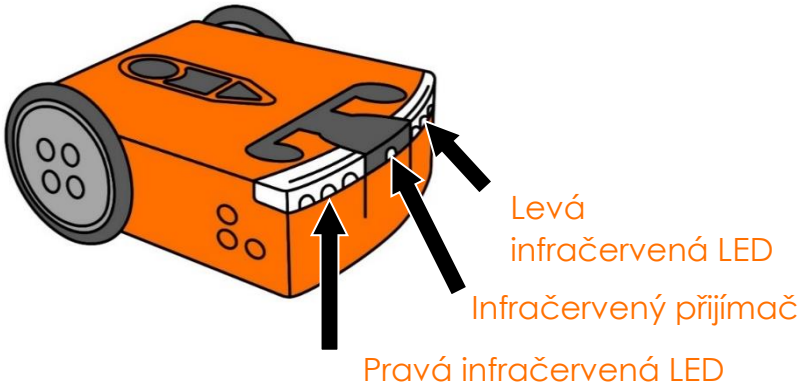
Jméno_____

U4-2.5 Prozkoumejte odesílání zpráv pomocí Edisona

Roboti Edison mají infračervený světelný senzor, který lze použít k detekci překážek. Tento senzor můžeme také použít jiným způsobem: k odesílání a přijímání infračervených zpráv.

Úkol 1: Edison a infračervené zprávy

Edisonův senzor infračerveného světla (IR) je snímač, který umožňuje Edisonovi vyzařovat a detekovat infračervené světlo. Senzor se skládá ze tří částí, které jsou umístěny přes přední část Edisona: dvě infračervené LED (jedna vpravo a jedna vlevo) a infračervený přijímač uprostřed.



Při detekci překážek používáme infračervené LEDky k vyzařování infračerveného světla a infračervený přijímač ke kontrole, zda se část tohoto světla neodrazil od objektů zpět k Edisonovi. Infračervený přijímač můžeme využít i k detekci infračerveného světla z jiných zdrojů, např. z dálkového ovladače TV či DVD přehrávače nebo z jiného Edisona. Pomocí infračerveného přijímače tímto způsobem můžete odesílat nebo přijímat zprávy pomocí robota.

Proč je to tak?

Dálkový ovládač televizoru využívá infračerveného světla k zasílání „zpráv“ televizoru. Tyto zprávy se však nepodobají zprávám, které posíláte např. přátelům. Jde o přednastavený signál, který aktivuje konkrétní funkci. Jedna zpráva např. říká televizi, aby hlasitost zvýšila, a jiná zpráva sděluje, aby hlasitost snížila.

Každé tlačítko na dálkovém ovladači odešle jinou zprávu pomocí infračerveného světla!

Nezapomeňte

Pro infračervené světlo se někdy používá zkratka IR. Bloky IR message (IR zpráva) v EdScratchi souvisejí s používáním infračervených zpráv pomocí infračervených LED a infračerveného přijímače.

Jméno_____

Infračervené světlo můžeme rovněž použít k odesílání a přijímání zpráv pomocí robotů Edison. Jeden robot může vyslat infračervenou zprávu pomocí svých dvou infračervených LED, které může detekovat IR přijímač jiného robota.

Úkol 2: Zpráva přijata

Chcete-li používat zasílání zpráv mezi roboty Edison, potřebujete vždy alespoň dva roboty, jednoho k odeslání IR zpráv a jednoho k detekci a reakci na IR zprávy.

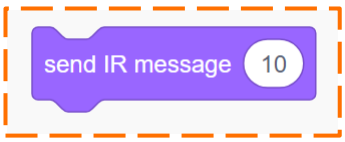
Pro tuto aktivitu budete potřebovat partnera nebo skupinu. Jeden robot pošle zprávu. Všichni ostatní roboti musí čekat, dokud nedetekují zprávu. Jakmile přijímající roboti detekují zprávu, měl by každý z robotů zareagovat tak, že se dá do tance!

Proč je to tak?

Edisonův infračervený senzor můžete použít v programu, který Edisonovi řekne, aby detekoval událost IR zprávy. Ke spuštění programu, který vysílá IR zprávu, potřebujete rovněž dalšího Edisona, aby váš měl první robot co detekovat!

Program pro ‘odeslání IR zprávy’

K odeslání infračervené zprávy pomocí Edisona budete potřebovat tento blok:



1. Blok send IR message se nachází v EdScratchi v kategorii bloků LEDs. Proč je to tak?

_____

_____

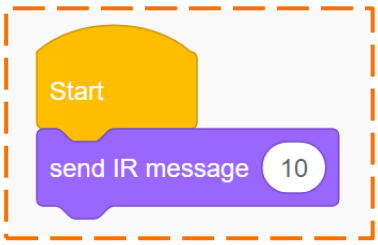
Blok send IR message má vstupní parametr pro odeslání konkrétní zprávy.

Proč je to tak?

Stejně jako dálkové ovládání může odesílat různé zprávy do televize, můžete odesílat různé zprávy ze svého Edisona. V EdScratch můžete zprávu upravit změnou hodnoty vstupního parametru v bloku send IR message (pošli IR zprávu). Vstupní hodnota bloku send IR message se může pohybovat v rozmezí 0 až 255. Edison může odeslat či přijmout 256 různých „zpráv“. Představte si, že by tolik zpráv uměl odeslat dálkový ovladač. Musel by mít skutečně hodně tlačítek!

Jméno_____

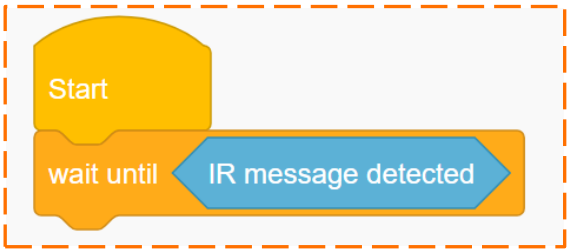
Někdy chceme vytvořit programy, které řeknou Edisonovi, aby na konkrétní zprávu reagoval konkrétním způsobem. Jindy, jako např. při této aktivitě, chceme, aby Edison reagoval, pokud bude detekována jakákoli IR zpráva. V robotovi, který vysílá ostatním robotům zprávy k detekci, musí běžet tento základní program:



Pro tuto aktivitu můžete nastavit vstupní parametr v bloku send IR message na libovolnou hodnotu. Musíte také napsat další kód pro robota, určující, co má dělat po odeslání zprávy IR. Použijte alespoň dva různé typy výstupů, včetně bloků z kategorie Drive, aby váš robot po odeslání zprávy tancoval.

Program „příjem IR zprávy“

Všichni roboti, kteří budou detekovat IR zprávy, musí mít stejný základní program:



Tento blok využívá blok z kategorie Sensing (Senzory) spolu s řídícím blokem wait until. K zapnutí detekce IR zpráv není třeba žádný blok.

Proč je to tak?

Některé z Edisonových senzorů jsou vždy zapnuté a kontrolují události. Jedním ze „vždy zapnutých“ senzorů je infračervený přijímač, takže k jeho zapnutí nepotřebujete kód. Musíte mu však sdělit, jaký druh IR události se má detekovat, a co dělat, pokud je tato událost detekována.

V příkladu základního programu kód řekne Edisonovi, aby hledal událost IR message detected (přijata IR zpráva). Tato událost bude spuštěna, pokud robot detekuje IR zprávu, bez ohledu na to, o jakou zprávu půjde. Proto nezáleží na tom, jakou hodnotu program „odesílání IR zprávy“ používá!

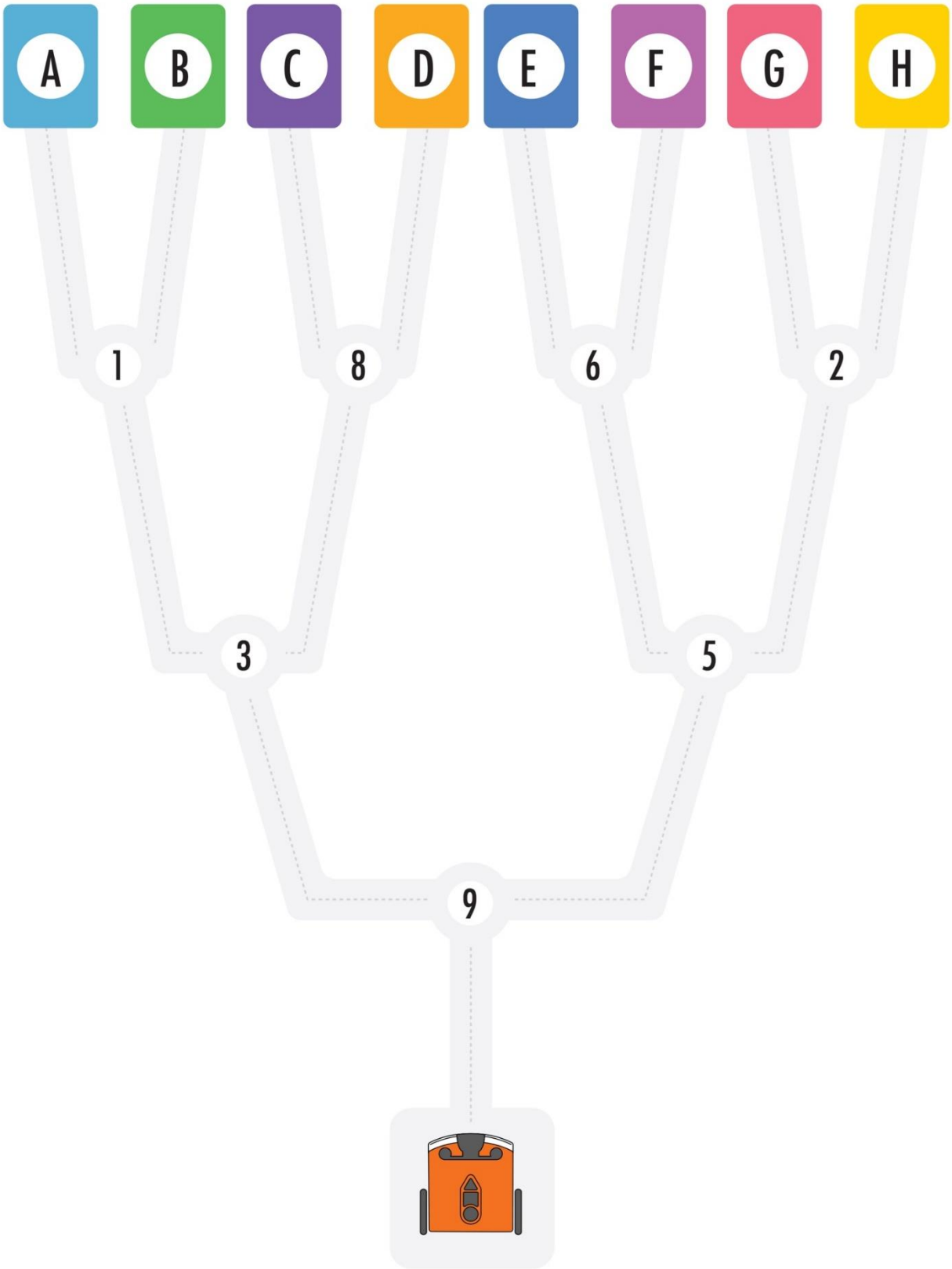
Použijte základní program a napište další kód, který robotovi řekne, co dělat, jakmile obdrží IR zprávu. Použijte nejméně dva druhy výstupů, včetně bloků z kategorie Drive, aby se robot dal po obdržení IR zprávy do tance.

Jakmile budou mít všichni roboty staženy programy, může začít večírek!

Nápověda!

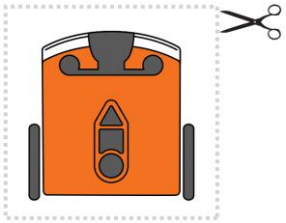
Nejprve stiskněte tlačítko pro spuštění (trojúhelníkové) na všech přijímajících robotech a poté na odesílajícím robotu. Pokud spustíte program odesílajícího robota dříve, než budou připraveni přijímající roboti, přijímající roboti zprávu nezaregistrují!

Pracovní list U4-1: Bludiště If-else

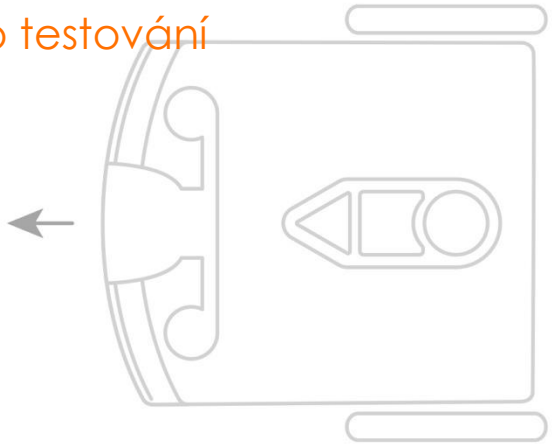


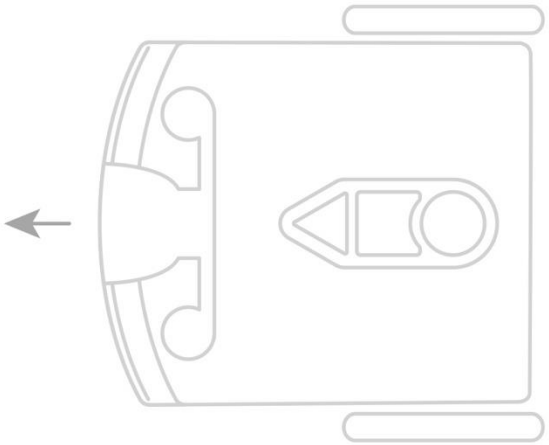
Pracovní list U4-2: Presudokód krok za krokem

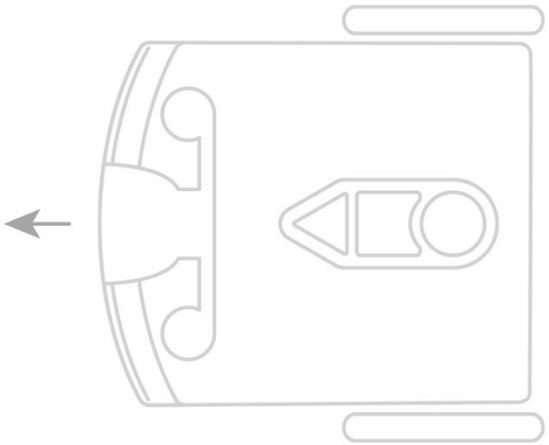
				J		D
	6		I			
F		C	7		9	
		H		4	G	2
3		1				E
B			A	8	5	

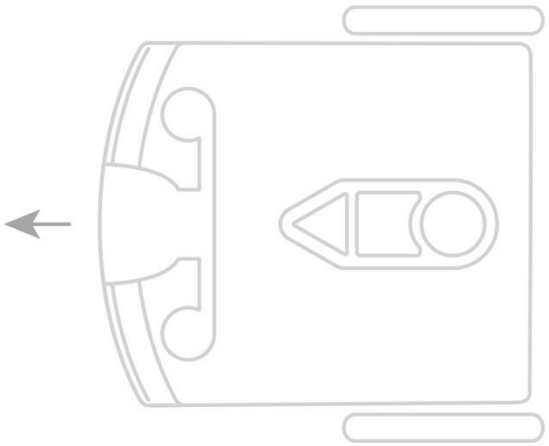


Pracovní list U4-3: Zóna pro testování sledování čáry

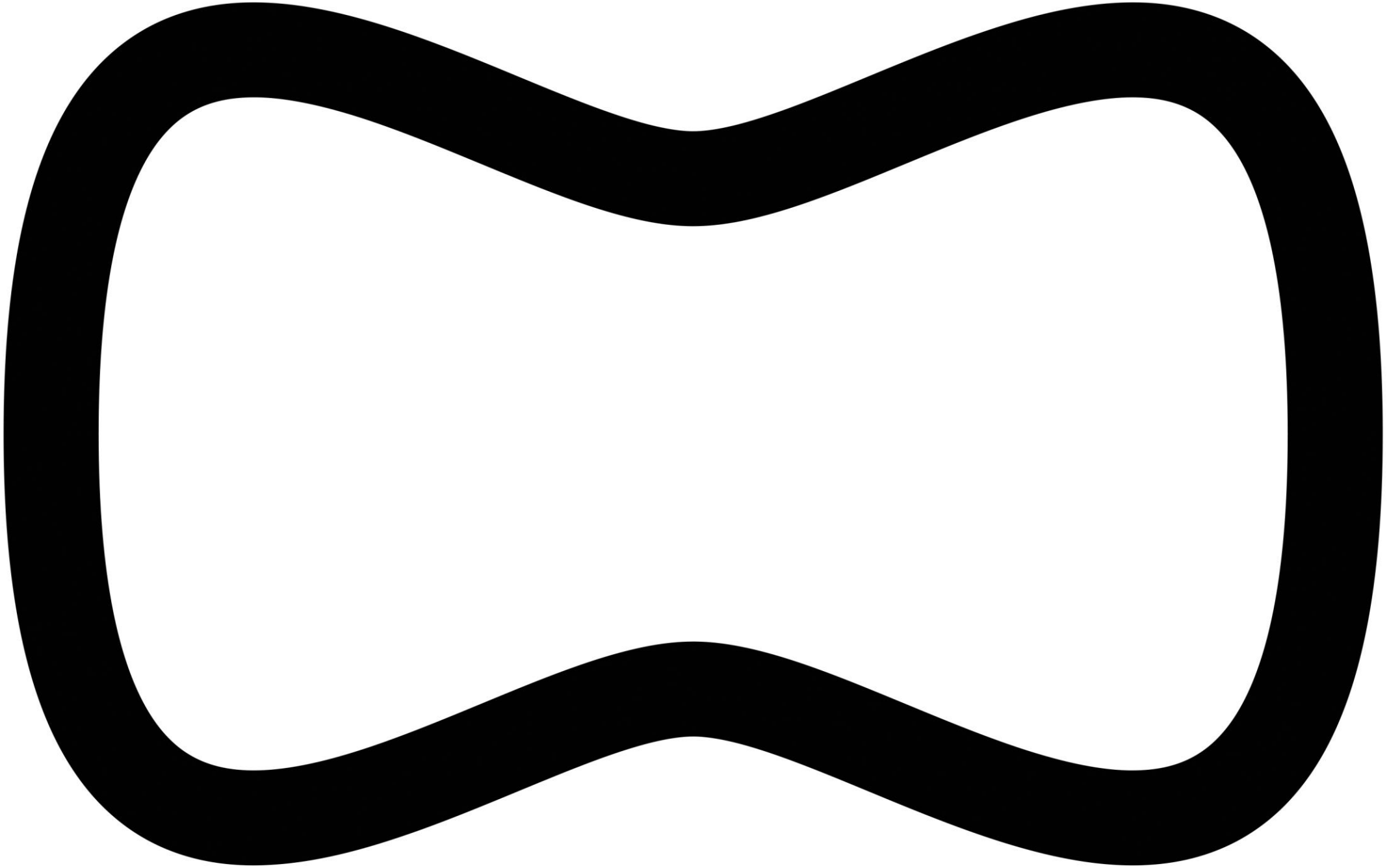






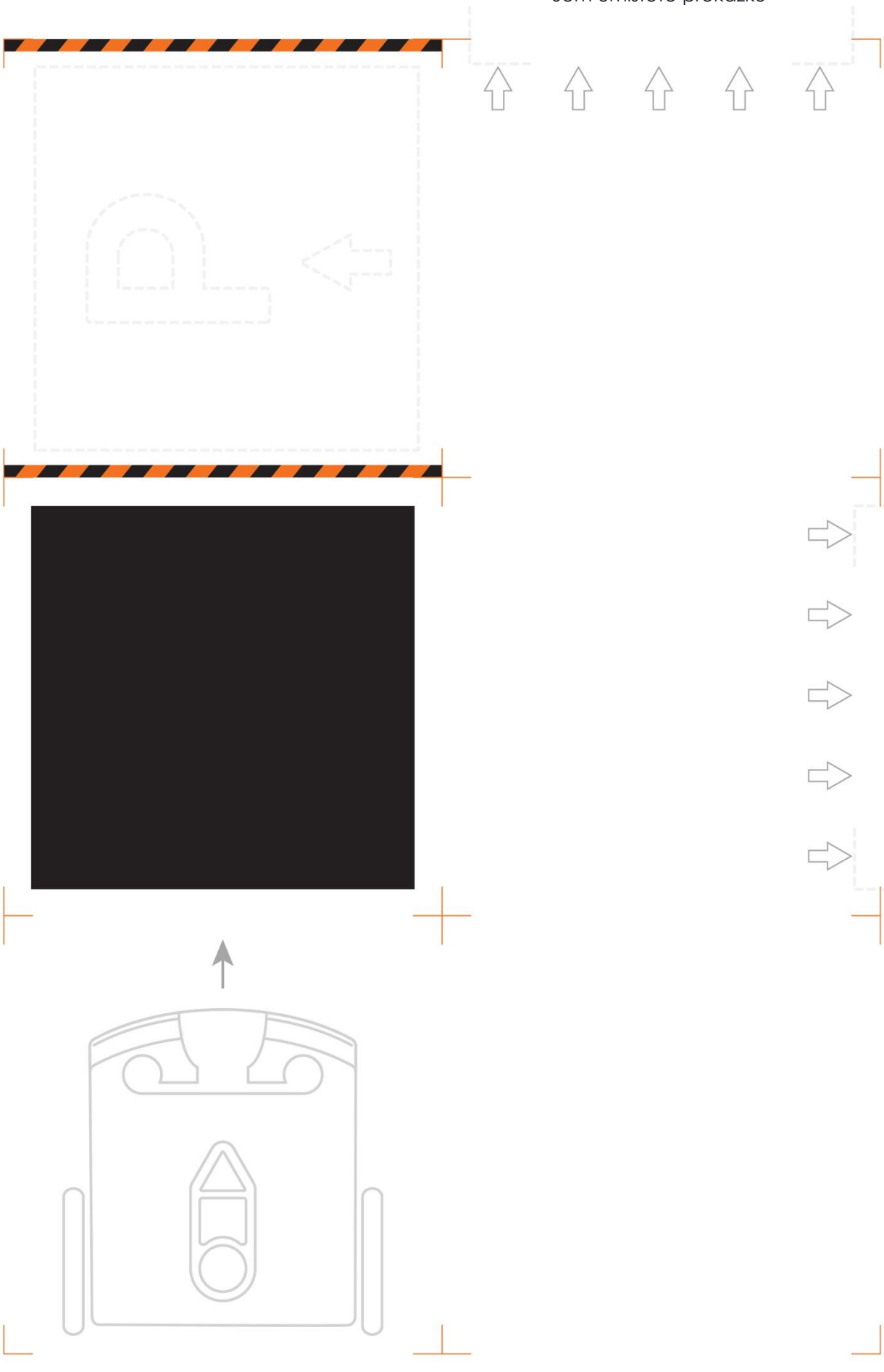


Pracovní list U4-4: Nereflexní ohraničení



Pracovní list U4-5: Pokud narazíš na čáru, jeď doprava

Sem umístěte překážku



Place obstacle over

Pracovní list U4-6: Programovatelné kódy dálkového ovladače TV

TV/DVD remote control code

TV/DVD remote control code

Kód dálkového ovladače TV/DVD

Kód dálkového ovladače TV/DVD

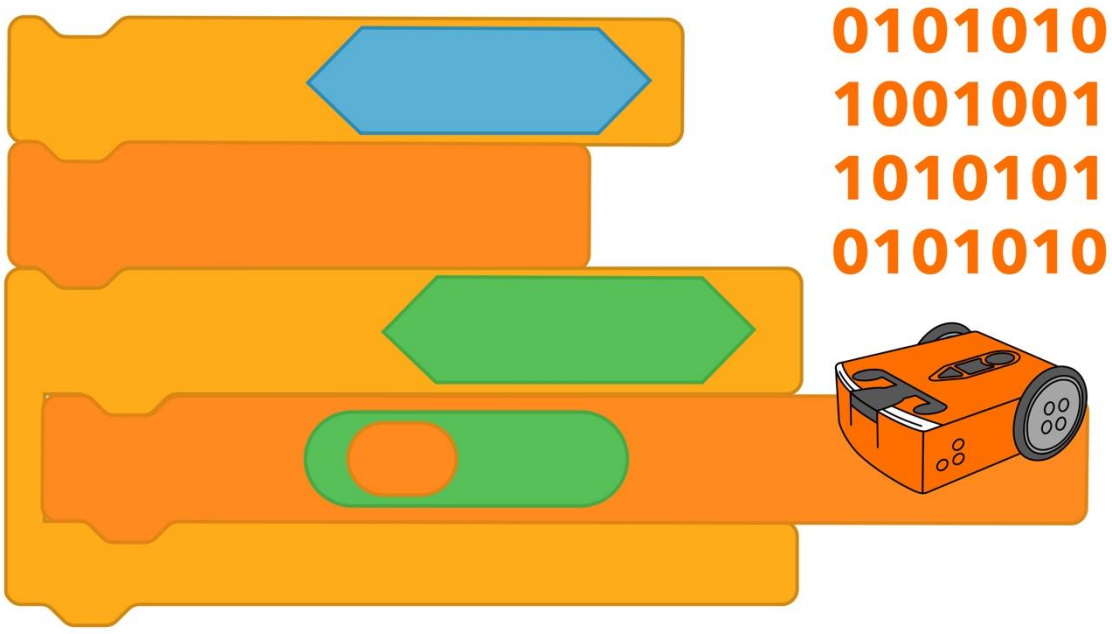
Kód dálkového ovladače TV/DVD

Kód dálkového ovladače TV/DVD

Kód dálkového ovladače TV/DVD

Kód dálkového ovladače TV/DVD

Lekce 5: Univerzální proměnné



Edison

Jméno_____

U5-1.1 Prozkoumejte výrazy

Věřte nebo ne, počítačové jazyky jsou většinou určeny pro lidi, nikoliv pro počítače.

Proč je to tak?

Počítače dokáží ve skutečnosti „myslet“ pouze v číslech, nikoliv ve slovech či blocích. Např. Edison nechápe bloky v EdScratchi tak, jak je vidíte na obrazovce svého počítače. Bloky je před stažením do robota nutné změnit do formátu, kterému Edison dokáže porozumět. Tento proces, který může nějakou dobu trvat, je důvodem, proč občas musíte chvíli počkat, než se při stahování programu z EdScratche do Edisona objeví ve vyskakovacím okně tlačítko **Program Edison**.

Počítačové jazyky, jako je EdScratch, nám usnadňují vytváření programů. Bez počítačového jazyka byste museli psát každý příkaz výhradně pomocí jedniček a nul!

Bloky v EdScratchi nám usnadňují zadávání vstupů do Edisona a psaní programů, které Edisonovi řeknou, co má udělat.

Je však důležité si uvědomit, že Edison vidí celý kód, který naprogramujeme, jako čísla. Podobně, když Edison ukládá data do své paměti ze senzoru, jsou tato data uložena ve formě čísel.

Nezapomeňte

Vstupy jsou informace a pokyny, které dáváte počítači.

V programech, které píšeme, můžeme čísla používat různými způsoby. Schopnost používat čísla a trochu matematiky v počítačových programech nám umožňuje přimět robota dělat mnoho různých věcí. Čísla a matematiku můžeme v EdScratchi využít např. ve **výrazech**.

Slovníček pojmů

V programování znamená výraz otázku, kterou je možné zodpovědět a vyřešit jako „pravdivou“ (true) nebo „nepravdivou“ (false). Výrazy mohou být použity spolu s jiným kódem (především s podmíněnými kódy, jako jsou bloky until nebo if) k řízení toku programu.

Výrazy nám umožňují vzájemně porovnat dvě čísla nebo části dat. Potom můžeme Edisonovi říct, co dělat v závislosti na výsledku.

Abyste mohli používat v EdScratchi výrazy, musíte vědět, jak číst, vyhodnocovat a řešit výrazy způsobem, jakým pracuje počítač.

Řešení výrazů

Kategorie bloků Operators (Operátory) v EdScratchi obsahuje speciální bloky, které potřebujete k napsání výrazu. Tyto bloky používají matematické značky (jinými slovy symboly) ke srovnání levé a pravé strany výrazu.

Např. výraz „Je A menší než B?“ je v EdScratchi zapsán jako „A < B“ a výraz „Je A rovno B?“ je zapsán jako „A = B“.

Podívejte se na seznam výrazů, které můžete v EdScratchi napsat:

Výraz	Význam
A = B	Je A shodné s B?
A ≠ B	Není A rovno B?
A > B	Je A větší než B?
A >= B	Je A větší nebo rovno B?
A < B	Je A menší než B?
A <= B	Je A menší nebo rovno B?

„A“ a „B“ ve výrazech můžete nahradit jakýmikoli hodnotami. Pomocí těchto hodnot můžete rovněž provádět výpočty. Např. „(A + 2) > B“ znamená „Je A plus 2 větší než B?“

Při psaní kódu výrazy fungují v konkrétním pořadí. Pokud váš výraz zahrnuje výpočty, výraz dokončí nejprve výpočty. Poté porovná levou stranu výrazu s pravou stranou a vyhodnotí výraz buď jako „pravdivý“ nebo „nepravdivý“.

Protože výrazy obsahují čísla, a někdy i výpočty, je lákavé o nich přemýšlet jako o matematických problémech, které mají odpověď – číslo. Musíte však o výrazech přemýšlet stejně jako počítač. Nejprve **vyhodnoťte** výraz dokončením všech výpočtů na obou stranách výrazu. Potom **vyřešte** výraz porovnáním levé strany zápisu s pravou stranou. Jinými slovy, odpovězte, zda je výraz „pravdivý“ nebo „nepravdivý“.

Proč je to tak?

Přestože výrazy vyhodnocují čísla a mohou obsahovat matematiku, jejich řešením je vždy jedna ze dvou odpovědí: „true“ (pravdivý) nebo „false“ (nepravdivý). Proto jsou při použití s podmíněným kódem tak užitečné. Než abychom se museli obávat, jaká bude přesná hodnota výrazu, můžeme pracovat v rámci možností „pravdivý“ a „nepravdivý“.

Řekněme například, že chcete, aby program něco dělal, pokud je hodnota vyšší než 10. Namísto psaní sady vnořených if bloků pro kontrolu přesné hodnoty můžete použít pouze jeden výraz, který X > 10. Program zkontroluje tuto hodnotu a bez ohledu na nastavenou hodnotu X, dokud bude X převyšovat 10, bude podmínka pravdivá!

V informatice existuje speciální název pro data, která mají pouze jednu ze dvou možných hodnot: tento druh dat nazýváme booleovská data (čti „bůleovská”).

Jméno_____

Schopnost porozumět tomu, co výrazy znamenají a co řeší, vám pomůže sledovat, co se v programu děje, jeho pouhým „přečtením“. Jde o důležitou dovednost při programování, známou jako trasování (**tracing**).

Slovníček pojmů

Trasování (Tracing) kódu znamená zpracování programu řádek po řádku a zaznamenávání důležitých hodnot. Schopnost trasovat program a porozumět tomu, co se děje, umožňuje programátorovi zjistit, jak by měl kód běžet, i když je uvnitř tohoto programu mnoho různých hodnot. Trasování kódu může pomoci najít logické chyby nebo chyby v kódu, ale rovněž se hodí, i když potřebujete jen pochopit, co se v programu děje.

Když používáte programy obsahující hodnoty a výpočty prováděné pomocí těchto hodnot, zaměřte se na trasování programu, abyste si byli jisti, že chápete, co se v programu děje. Trasování vám pomůže porozumět, co se má v programu dít, a umožní vám program odladit, pokud nefunguje tak, jak má.

Vyzkoušejte si to!

Zkuste vyřešit následující výrazy. Nejprve si napište, co jednotlivé výrazy znamenají, a poté vyřešte, zda jsou pravdivé (true) nebo nepravdivé (false).

Pokud u následujících otázek platí, že A = 2 a B = 4, jaký má každý z následujících výrazů význam (jinými slovy, jakou otázku klade) a jaké je jeho řešení (pravdivý či nepravdivý)?

1. (A*2) = B

   Význam: _____

   Řešení: _____

2. A >= B

   Význam: _____

   Řešení: _____

3. (A+A) ≠ B

   Význam: _____

   Řešení: _____

4. (A-1) < (B-3)

   Význam: _____

   Řešení: _____

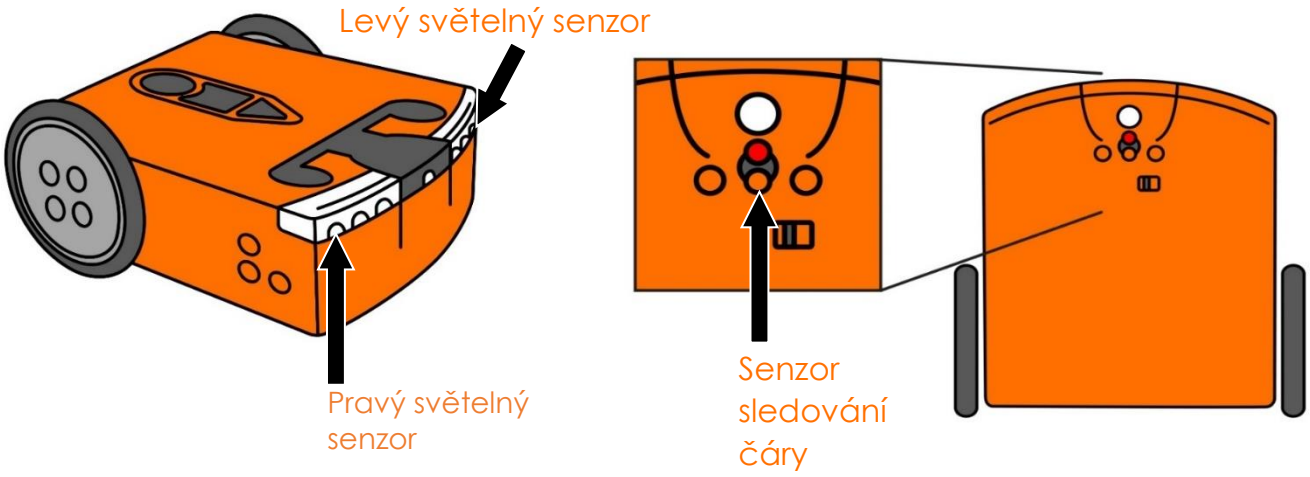
Jméno_____

U5-1.2 Prozkoumejte Edisonovy světelné senzory

Roboti Edison jsou vybaveni různými senzory, které mohou detekovat různé věci. Jedním z druhů senzorů, jimiž je Edison vybaven, jsou světelné senzory.

Úkol 1: Seznamte se s Edisonovými světelnými senzory

Edisonovy světelné senzory robotovi umožňují detekovat a měřit viditelné světlo. Hlavní světelné senzory jsou umístěny na horní části robota – jeden vpravo a druhý na levé části robota. Edison má ještě třetí světelný senzor, který je umístěn na spodní části robot a slouží jako senzor pro sledování čáry. Prohlédněte si svého Edisona. Vidíte všechny světelné senzory?



Podobně jako Edisonův infračervený přijímač dokáže detekovat infračervené světlo, lze světelné senzory použít k detekci viditelného světla, které tvoří lidským okem viditelnou část světelného spektra. Všechny senzory viditelného světla v Edisonovi fungují v zásadě podobně jako senzor pro sledování čáry, který detekuje reflexní a nereflexní povrchy pod robotem.

Nezapomeňte

Senzor pro sledování čáry funguje tak, že osvětluje světlem z červené LEDky povrch pod robotem. Světelný senzor pak měří, kolik z tohoto světla se odráží od povrchu. Edison ukládá hodnotu odraženého světla jako odečet světla. Čím více světla se odráží zpět k Edisonovi, tím vyšší je hodnota světla.

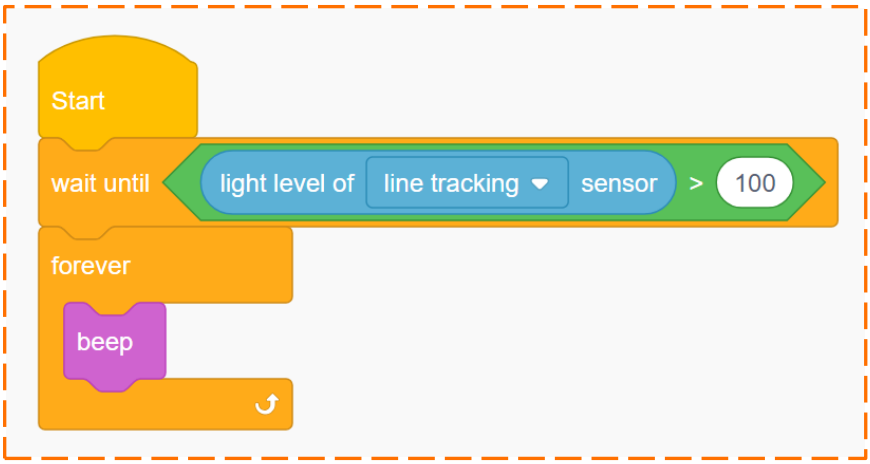
Edisonovy světelné senzory se nemusejí spoléhat pouze na infračervené světlo z Edisonových LEDek. Tyto senzory můžeme rovněž použít k měření detekovaného světla a hodnotu naměřeného světla uložit. Čím více Edison detekuje světla, tím vyšší hodnotu světla uloží.

Naměřenou hodnotu z některého ze světelných senzorů můžeme použít jako hodnotu v programu v EdScratchi.

Úkol 1: Světelný poplach

Použijte jeden z Edisonových světelných senzorů ke spuštění poplachu v okamžiku, kdy Edison detekuje dostatek světla.

Prohlédněte si následující program v EdScratchi:



Tento program říká Edisonovi, aby počkal, než úroveň světla zaznamenaná senzorem pro sledování čáry přesáhne 100, a pak začal pípat navždy (forever).

Proč je to tak?

Senzory světla měří zjištěné viditelné světlo a ukládají jeho hodnotu jako hodnotu světla. Stejně jako všechna data senzorů i Edison ukládá tuto hodnotu jako číslo. Výraz v tomto programu říká Edisonu, aby porovnal hodnotu odečtu světla ze senzoru sledování čáry s číslem 100.

Proč 100? Pamatujte, že čím více světla je detekováno, tím vyšší je hodnota světla. Světelné senzory Edison mohou poskytnout hodnoty světla s hodnotami od 0 do něco přes 1000. Hodnota 100 je tak dobrým výchozím bodem pro testování s tímto programem.

Napište program pro světelný poplach a stáhněte jej do svého Edisona. Před stisknutím tlačítka pro spuštění (trojúhelníkového) ke spuštění programu senzor pro sledování čáry něčím zakryjte, např. palcem. Ujistěte se, že je světelný senzor zcela zakrytý. Držte robota tak, aby senzor pro sledování čáry nebyl natočený k žádnému jasnému světlu, např. ke stropnímu osvětlení nebo slunečnímu jasu z okna.

Až zakryjete robotův světelný senzor, stiskněte tlačítko pro spuštění. Jakmile budete připraveni, odsuňte prst pryč ze světelného senzoru a zamiřte robota směrem ke zdroji světla. Jakmile robot detekuje dostatek světla, podmínka bloku wait until (čekej dokud) bude splněna a kód se přesune k následujícímu bloku a spustí 'poplach' pomocí bloku beep.

Jméno_____

Úkol 2: Automatická pouliční lampa

Už jste někdy viděli pouliční lampu, která se rozsvítí, když je venku tma? Pravděpodobně se tak děje pomocí světelného senzoru! Můžete přimět Edisona, aby se choval stejným způsobem a rozsvítil červené LEDky, když je úroveň světlá nízká, a vypnul je při dostatku světla.

Napište program, díky kterému se váš Edison bude chovat jako pouliční lampa citlivá na světlo. Program by měl zapnout Edisonovy červené LEDky, kdykoli robot detekuje velmi málo viditelného světla, zatímco když bude světla dostatek, měly by LEDky být vypnuté.

Nápověda

- K měření hodnoty světla pro tento program potřebujete pouze jeden z horních světelných senzorů na Edisonovi. Vyberte si levý nebo pravý senzor.
- Vyzkoušejte různé hodnoty světla a zjistěte, která funguje nejlépe...
- Zasekli jste se při psaní? Podívejte se na program z úkolu 1. Jak ho můžete upravit tak, abyste z něj vytvořili program pro automatickou pouliční lampu?

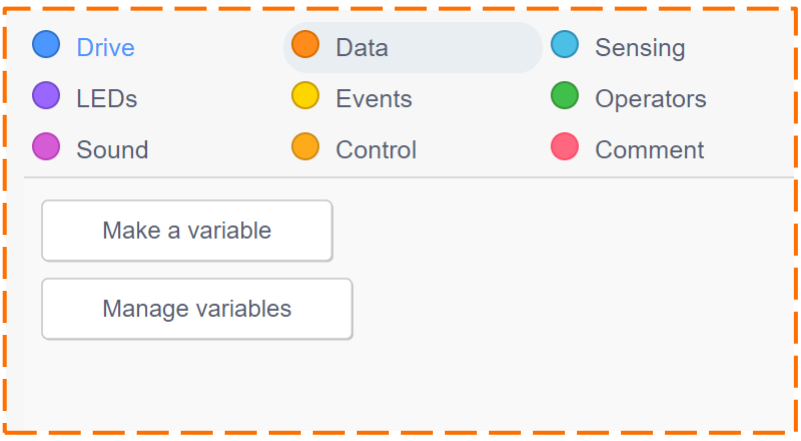
Stáhněte program do svého robota a vyzkoušejte jej. Dejte Edisona nejprve na místo, kde je hodně světla, poté na špatně osvětlené místo. Funguje Edison díky vašemu programu jako automatická pouliční lampa?

1. Jak vypadá váš program pro automatickou pouliční lampu? Napište jej sem.

Jméno_____

U5-1.3 Prozkoumejte proměnné

V EdScratchi je jedna kategorie bloků, která při prvním otevření vypadá úplně jinak, než ostatní: kategorie Data.



Když poprvé kliknete na kategorii Data, nezobrazí se vám žádné bloky, které byste mohli použít k napsání programu v EdScratchi. Namísto toho tato kategorie obsahuje dvě tlačítka: Make a variable (Vytvořit proměnnou) a Manage variables (Spravovat proměnné) Pomocí kategorie Data můžeme vytvářet **proměnné** (variables) do programů.

Slovníček pojmů

Proměnná (variable) je část paměti používaná k ukládání hodnoty v programu. Můžete si ji představit jako zásobník, do nějž uložíte část informace ve tvaru, který bude srozumitelný pro počítač.

V programování často používáme tutéž část informace v jednom programu několikrát. Proměnné toto opakování výrazně usnadňují. Použití proměnných v programu nám umožní říct počítači, aby do proměnné uložil konkrétní informaci. Tuto proměnnou můžeme použít na různých místech v programu. Kdykoliv počítač uvidí tuto proměnu, vyvolá informace uložené uvnitř proměnné.

Když v EdScratchi kliknete na tlačítko Make a variable (Vytvořit proměnnou), otevře se vyskakovací okno s požadavkem, abyste proměnnou pojmenovali. Je důležité dát proměnným správná jména: jde o to, aby vám název proměnné dával smysl a sdělil vám, jaké informace jsou uvnitř uloženy.

Proč je to tak?

Názvy proměnných musí dávat smysl vám i počítači. Počítačové jazyky obsahují často pravidla o tom, jaké znaky lze použít v názvech proměnných. Počítač dokáže porozumět pouze proměnným s názvy, které se řídí těmito pravidly.

Názvy proměnných v EdScratchi mohou obsahovat pouze malá a velká písmena anglické abecedy, čísla a podtržítka (_). Jiné znaky, jako např. vykřičník (!) nebo mezera, nejsou v názvech přípustné.

Dejte proměnné název, který vám pomůže určit, jaký typ informací bude uvnitř proměnné uložen. Pokud chcete změnit název proměnné poté, co jste ji vytvořili, můžete to udělat pomocí tlačítka Manage variables (Spravovat proměnné).

Jakmile vytvoříte a pojmenujete proměnnou, objeví se v kategorii Data spolu s dalšími speciálními bloky, včetně set (nastavení), increment (snížení) a decrement (zvýšení). Tyto bloky můžete použít v programu v EdScratchi spolu s proměnnou.

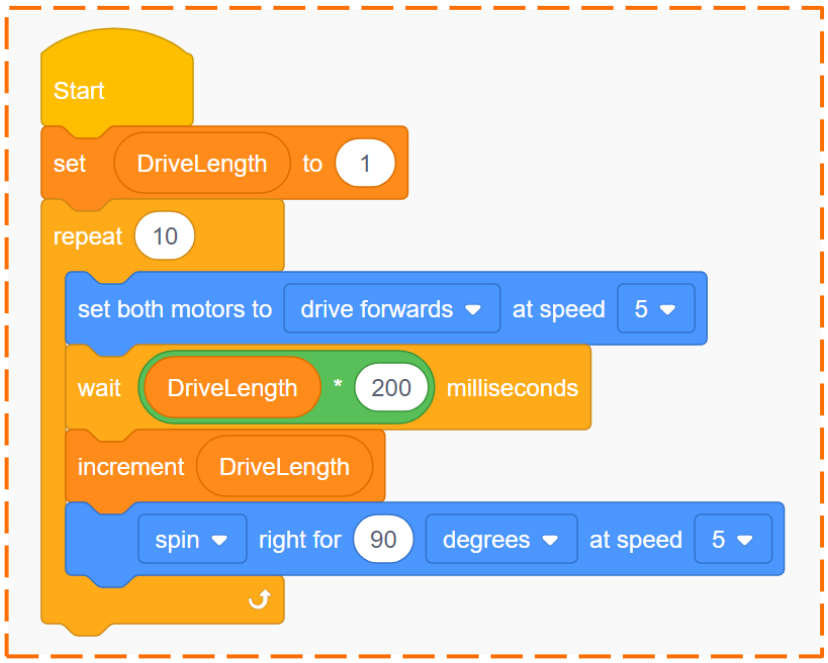
Slovníček pojmů

Increment (inkrementace) znamená v programování zvýšení o 1 a **decrement** (dekrementace) znamená snížení o 1.

Úkol 1: Sledujte kód

V EdScratchi můžeme pomocí proměnných ukládat různé hodnoty. Protože tyto hodnoty jsou čísla, můžeme s těmito čísly provádět různé operace, například je porovnávat s jinými hodnotami nebo s nimi dělat výpočty.

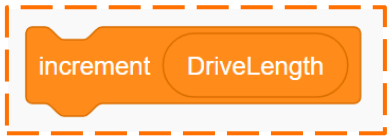
Prohlédněte si následující program:



Tento program používá proměnnou nazvanou DriveLength (délka jízdy).

Je důležité si uvědomit, že proměnná představuje hodnotu nastavenou někde v rogramu. Proto musíte ve svém programu vždy použít kód, abyste počítači řekli, jak nastavit proměnnou.

Blok set (nastavit) provádí následující kroky. Na začátku programu blok set nastaví hodnotu proměnné DriveLength na 1. Hodnota proměnné DriveLength však nezůstane navždy nezměněná, protože program používá jiný blok z kategorie Data – blok increment (zvýšit):



Blok increment je vložen dovnitř cyklu repeat (opakuj) a mění hodnotu proměnné DriveLength na novou. Na jakou hodnotu mění tento blok proměnnou DriveLength?

To bude záležet na tom, kde v programu se nacházíte. Jinými slovy, bude záležet na tom, kolikrát proběhl cyklus repeat.

Proč je to tak?

Jedním z důvodů, proč používáme v programech proměnné, je skutečnost, že proměnná může uchovávat informaci, i když se hodnota této informace změní. Může se vám to zdát nesrozumitelné, ale zkuste si to představit takto:

Představte si, že máte proměnnou nazvanou **YearsOld** (věk). Když jste oslavili první narozeniny, proměnná **YearsOld** byla nastavena na 1. Poté se každý rok o vašich narozeninách proměnná **YearsOld** zvýší a její hodnota se změní na novou hodnotu: (**YearsOld** + 1).

Rok po vašich prvních narozeninách je hodnota proměnné **YearsOld** zvýšena. Protože úvodní hodnota **YearsOld** byla 1 a protože 1 + 1 = 2, nová hodnota **YearsOld** činí 2. Vše se opakuje na vaše další narozeniny, kdy se hodnota **YearsOld** opět zvýší na číslo 3.

Hodnota **YearsOld** se na každé narozeniny mění, ale typ informace nikoli. Proměnná **YearsOld** označuje váš věk, ať již máte za sebou jakkoli mnoho narozenin!

Na začátku našeho programu bude hodnota proměnné DriveLength nastavena na 1, ale protože ve do kódu s cyklem vložen blok increment (zvýšení), hodnota se s každým proběhlým cyklem inkrementuje.

Proměnná DriveLength je použita v kódu s cyklem i na jiném místě:

Vstupní hodnota tohoto bloku wait (čekej) závisí na hodnotě DriveLength a bude se rovněž v každém cyklu měnit.

Dejte si ale pozor!

Zatímco hodnota bloku wait se bude měnit v závislosti na hodnotě proměnné DriveLength, blok wait nebude měnit hodnotu proměnné DriveLength. Hodnotu této proměnné dokáže změnit pouze blok z kategorie Data.

Rovněž si všimněte, že hodnota bloku wait je uvedena v milisekundách, nikoli sekundách.

Proč je to tak?

Kdykoli budete chtít použít proměnou nebo blok z kategorie Operators (Operátory) jako vstup v bloku wait, budete muset použít tento speciální blok wait () milliseconds. Edison ve skutečnosti „myslí“ v milisekundách, nikoliv sekundách. Pomocí tohoto bloku tak je možné provádět výpočty, kterým Edison rozumí.

Nezapomeňte, že 1 sekunda = 1000 milisekund.

Nezapomeňte

Sledování kódu znamená procházení programu řádek pořádku a zaznamenávání důležitých hodnot.

Sledujte program a zkuste přijít na to, jakou hodnotu bude mít proměnná DriveLength a jaká bude vstupní hodnota bloku wait Na různých místech kódu v programu.

1. Sledujte program a vypočítejte hodnoty. Do tabulky uveďte, jaká bude úvodní hodnota proměnné DriveLength na začátku každého opakování cyklu, jaká bude v daném cyklu vstupní hodnota bloku wait a jaká bude nová hodnota proměnné DriveLength po proběhnutí bloku increment v daném cyklu. První dva řádky tabulky jsou již předvyplněny.

V cyklu č.	Úvodní hodnota DriveLength	Vstupní hodnota bloku Wait (v milisekundách)	Nová hodnota DriveLength
1	1	200	2
2	2	400	3
3			
5			
7			
10			

Jméno_____

Úkol 2: Napište a spusťte program

Co Edison po spuštění programu udělá?

Napište program v EdScratchi. Stáhněte jej do Edisona a spusťte. Sledujte, co se bude dít. Poté zodpovězte následující otázky.

2. Co Edison po spuštění programu dělá? Jaký „tvar“ robot opisuje při jízdě?

_____

_____

3. Podívejte se na kód v programu. Vysvětlete, proč při spuštění programu robot jezdí podle pozorovaného vzorce. Co v kódu způsobuje, že se robot takto pohybuje?

_____

_____

_____

_____

Jméno_____

U5-1.3a Další výzva: pavouk ve spirále

Někteří pavouci staví pavučiny, které se spirálovitě stáčejí do bodu uprostřed pavučiny. Dokážete naprogramovat Edisona tak, aby jel po spirále do středu, podobně jako když pavouk klade síť?

Co máte dělat

Napište v EdScratchi program, který umožní Edisonovi jet po spirále směrem do středu. Edison by měl jet, pak zatočit, pak opět jet a tak stále dokola, dokud nedojede do středu spirály.

Váš program by měl pomocí proměnné řídit, jakou vzdálenost Edison pokaždé ujede.

Nápověda!

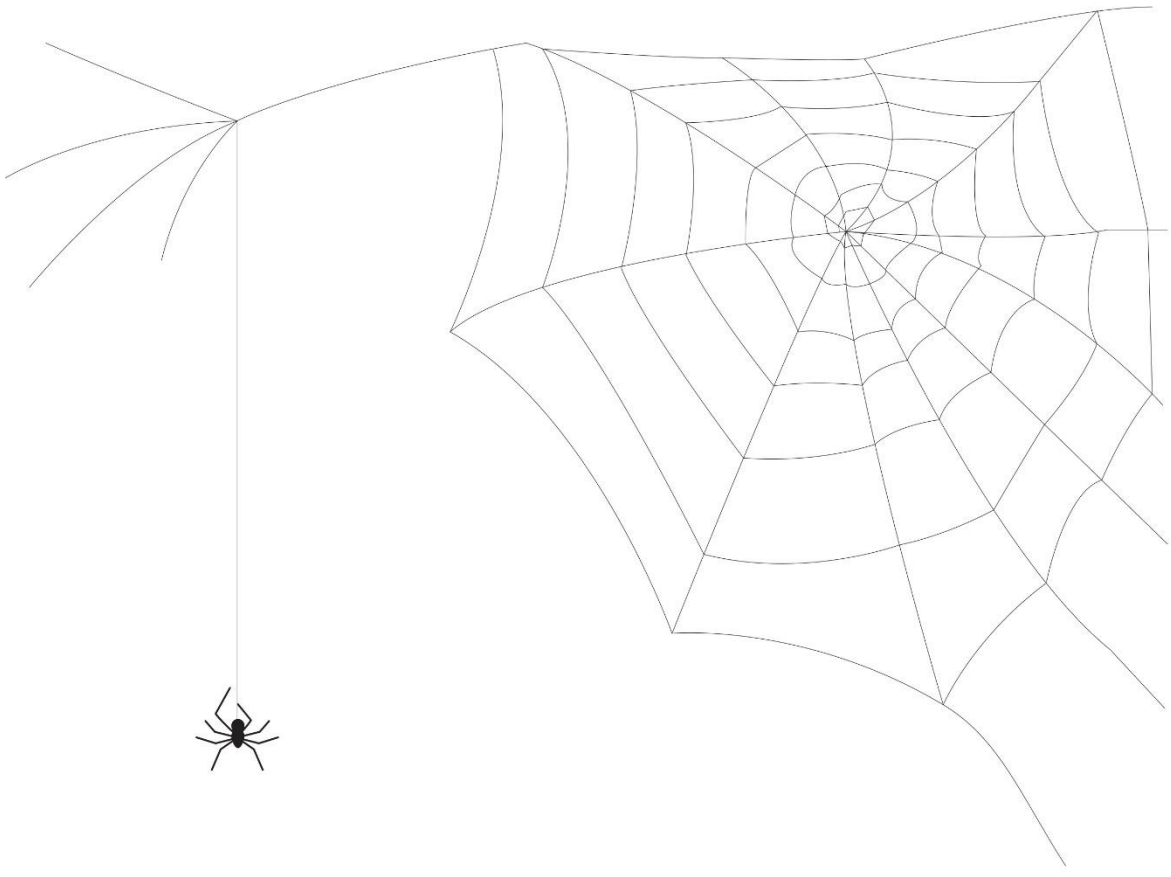
Inspiraci pro psaní tohoto programu můžete najít v aktivitě U5-1.3.

Zamyslete se nad tím, jak daleko má Edison v každém úseku jet v porovnání s předchozím úsekem. Rovněž budete muset rozhodnout, jak moc má Edison mezi jednotlivými úseky zatočit.

Stáhněte svůj program do Edisona a vyzkoušejte jej. Experimentujte s různými hodnotami vstupů a zjistěte, která funguje nejlépe.

Mini výzva!

Dokážete přimět Edisona, aby za sebou zanechával při jízdě „lepkavou pavoučí síť"? Síť vyrobte např. z vlákna nebo niti. Zkuste vymyslet řešení tak, aby za sebou Edison při jízdě zanechával pavoučí síť.



Jméno_____

U5-1.4 Prozkoumejte používání proměnných s daty ze senzorů

Jedním z nejzajímavějších způsobů použití proměnných v EdScratchi je uložení a využití dat z Edisonových senzorů. Protože všechny Edisonovy senzory odesílají Edisonovi zpět data jako hodnoty, můžete uložit hodnotu ze senzoru do proměnné. Data ze senzoru můžete také použít k ovlivnění proměnné, například ke zvýšení proměnné pokaždé, když senzor něco detekuje. Společným použitím proměnných a senzorů v programu můžeme přimět Edisona, aby různými způsoby reagoval na různá data ze senzorů.

Nezapomeňte

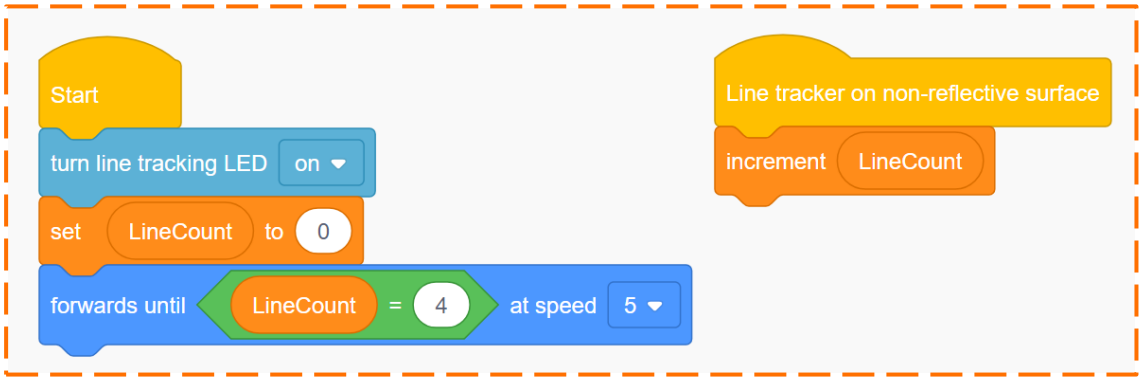
Proměnná (variable) je část paměti používaná k uložení informace v programu.

Vyzkoušejme si použití proměnné z Edisonova senzoru pro sledování čáry – naprogramujeme Edisona tak, aby jel, dokud nedetekuje čtyři černé čáry.

Co máte dělat

Pomocí Edisonova senzoru pro sledování čáry můžete vytvořit program, který Edisonovi řekne, aby jel, dokud nenarazí na černou čáru. V této aktivitě musíte přimět robota, aby přejel přes několik černých čar a zastavil teprve tehdy, až detekuje čtvrtou černou čáru.

Prohlédněte si následující program:



1. Co se v tomto programu děje? Když program stáhnete do Edisona, co robotovi řekne, aby dělal?

_____

_____

_____

_____

2. Tento program obsahuje proměnnou LineCount (počet čar). Proč se podle vás programátor rozhodl pojmenovat proměnnou právě takto?

_____

_____

Jméno_____

Vyzkoušejte to

Napište program v EdScratchi a stáhněte jej do svého robota. Vyzkoušejte program na Pracovním listu U5-1 nebo si vytvořte vlastní zkušební prostor ze čtyř souběžných černých čar na bílém povrchu např. na velkém plakátu, stole či podlaze. Čáry můžete rozmístit libovolně daleko od sebe.

3. Použití některého z Edisonových senzorů s proměnnou nám umožňuje vytvářet nejrůznější programy a používat Edisona různými způsoby. K čemu dalšímu byste mohli přimět Edisona pomocí proměnných a dat ze senzorů? Představte alespoň jeden nápad na program využívající proměnné v kombinaci s údaji ze senzorů. Popište svůj nápad.

_____

_____

_____

_____

_____

4. Lze váš nápad přepsat do kódu? Zkuste napsat program v EdScratchi. Jak to šlo? Podařilo se vám úspěšně napsat program a otestovat jej pomocí Edisona? Pokud ano, fungoval? Popište celý proces proměny myšlenky na program.

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Jméno_____

Pracovní list U5-1: Čtyři čáry