Eaton virtuální závod autonomních aut
Zadání

Vítej zpět mladý navigátore – programátore. V tuto chvíli bys již měl mít zprovozněné vše, co potřebuješ k řešení následujících úloh. Pokud by sis s něčím nevěděl rady, budeme Ti k dispozici na Discordu: https://discord.gg/9PaJT8SY .

Pokud jsi nečetl úvodní instrukce, tak se nejprve podívej na tuto stránku: https://olympiada.ddmp6.cz/Eaton_2021.htm

Pro řešení úloh doporučujeme využít některý z mnoha programovacích editorů (Notepad++, VisualStudioCode, PyCharm, nebo jakýkoliv jiný), které Ti usnadní psaní a čitelnost Tvého programu.

Prvním krokem je stažení nové verze podkladů (soutěžní verze) z následujícího odkazu:

https://github.com/DDMP6/Eaton-virtualni-zavod/tree/task

nebo rovnou *.zip archiv
https://github.com/DDMP6/Eaton-virtualni-zavod/archive/refs/tags/v1.1.zip

Nyní se pustíme do jednotlivých úkolů a k hodnotícím kritériím dodáme jen to, že i částečné zdolání tratě může přinést body (na pamách jsou “checkpointy”) vyznačené vždy červenou čarou.

Powering Business Worldwide

BÍLÁ HORA DDM PRAHA 6

Zahřívací úloha – ovládání autíčka pomocí jednoduchého programu

K projetí mapy (mapa č. 1 v sekci Přílohy) musíš sestavit správnou posloupnost příkazů, podle kterých bude autíčko vědět, kdy má zrychlit, zpomalit nebo zatočit a o kolik.

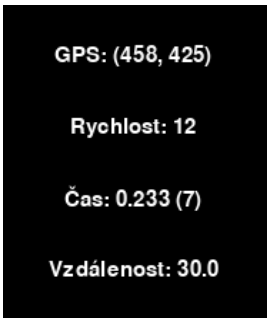
Příkazy budeš vyplňovat do souboru **input.txt**.

Řádky začánající symbolem # jsou jen komentáře, které slouží jako vysvětlivky nebo nápověda.

Pokud chceš, aby autíčko v okamžiku timestamp = 7 (na obrázku v závorce), zatočilo o 5 stupnů doprava, tak použiješ tento příkaz:

```
7 – right 5
```

Timestamp:



V jednom příkazu můžes zrychlit nebo zpomalit nejvýše o 5 bodů a zatočit o 10 stupňů. Autíčko může dosáhnout nejvýše rychlosti 15 bodů.

Svůj algoritmus otestuješ spuštěním programu **task1.bat**.

Až dovedeš autíčko úspěšně do cíle, tak si pečlivě uschovej soubor input.txt s Tvými úpravami a později ho spolu s ostatními soubory odevzdáš.

Druhá část první úlohy Tě zavede do světa programovacího jazyka Python. Pokud Python používáš poprvé, podívej se na úvod do Pythonu dále v sekci Přílohy. Tvým úkolem bude příkazy tak, jak jsi je napsal do souboru input.txt, přenést do souboru **Brain.py** a upravit je tak, aby vznikl správný algoritmus / program v Pythonu.

Úpravy budeš dělat jen a pouze v této části souboru:

```
########################################################################
        # ZAČÁTEK BLOKU
        # Zde implementujte váš algoritmus

########################################################################

        # Nahradit vaším programem
        self.faster(2)


########################################################################
        # KONEC BLOKU

########################################################################
```

Aby Tvůj Python algoritmus fungoval správně, tak řádky ze souboru input.txt zapíšeš do souboru Brain.py následovně:

```
1 - faster 5
2 - right 10
```

```
if self.getTimestamp() == 1:
    self.faster(5)
if self.getTimestamp() == 2:
    self.right(10)
```

Pro spuštění této části úlohy využij soubor **task2.bat**.

V dalších úlohách využijeme stejný soubor Brain.py, proto si **po dokončení** této části udělej **kopii** svého řešení a pojmenuj ho **Brain_1.py**, abys nám ho mohl(a) později odeslat.

Pokročilejší úlohy – autonomní řízení na základě vstupů

Nejprve se ujisti se, že sis udělal **kopii souboru Brain.py** viz. předchozí úlohu!

Nyní nastává další část, která už bude vyžadovat využití dalších funkcí a principů programování.

V předchozím úkolu, jsme naprogramovali „mozek autíčka“ jako posloupnost konkrétních kroků, které má provést (jeď rovně, po 5 krocích zatoč vlevo, po 10 krocích zpomal, atd.).

Další úlohy mají za cíl naprogramovat složitější mozek bližší tomu reálnému. A to tak, aby bral v potaz vstupy ze senzorů na autíčku (lidar, orientace, souřadnice) – například, když zjistím, že přede mnou je zeď, tak zatočím.

Tvůj program tedy bude fungovat tak, že v každém okamžiku by měl zpracovat vstupy ze senzorů a patřičně na ně zareagovat. Můžeš si to představit tak, že se bude program spouštět pokaždé znovu a autíčko se mezi tím o kousek posune (například blíže ke zdi).

Princip mozku i nástin čtení dat ze senzorů jsou na obrázcích v části *Přílohy*.

Mapy pro následující úlohy jsou celkem tři, ale program budeš odevzdávat jen jeden. Tvůj algoritmus by tedy měl umět projet alespoň jednou mapou, ale v ideálním případě zvládne všechny tři!

První z těchto map (stejná jako v předchozí úloze) by se měla dát zvládnout s minimem příkazů. Tuto mapu otestuješ spuštěním souboru **task2.bat**.

Ale až se dostaneš k druhé mapě (mapa č. 2 v Přílohách), tak bude Tvůj algoritmus složitější. Pro otestování této mapy použij soubor **task3.bat**.

Edit: V *task3* se chybně detekují lidarem auta i bílé texty. Pro opravu zakomentujte v souboru src/Game.py řádky 283 a 284:

```
# self.screen.blit(gps_overlay, gps_overlay_rect)
# self.screen.blit(speed_overlay, speed_overlay_rect)
```

Dále se v *task3* objevují chyby v dojezdu do cíle (v některých případech nelze projet cílovou páskou), způsobené tím, že je za cílovou páskou konec. Problém lze vyřešit nahrazením řádky 81 v souboru src/Tracks.py na následující:

```
FinishLineSprite((850, 50), 96, True)
```

Další mapa (č. 3 v Přílohách) tvůj algoritmus prověří a zjistí, jestli je třeba jej ještě vylepšit. Ověř svůj algoritmus pomocí souboru **task4.bat**.

Poslední mapa je tajná. Její tvar neznáš a navíc se na ní při zemětřesení sesypala skála a vytvořila na cestě mnoho překážek. Tvůj algoritmus by měl být natolik univerzální, že si poradí s jakýmikoliv překážkami. Jedinou nápovědou je, že vždy existuje alespoň jedna cesta ze startu do cíle.

Poslední mapu si u sebe neotestuješ, protože tu známe jen my a otestujeme ji, až své programy odevzdáš.
Výsledný soubor s algoritmem pojmenuj **Brain_2.py**.

Závěr

Výsledkem tvé práce by měl být zip archiv, který bude obsahovat následující 3 soubory:

1. Input.txt
2. Brain_1.py
3. Brain_2.py

Tento zip pojmenuj prijmeni_j.zip, kde „j“ je iniciála křestního jména a zašli na olympiada-programovani@ddmp6.cz nebo nahraj na náš Ownloud zde: https://owncloud.cesnet.cz/index.php/s/x3cueo6mVwr7iyX (nedej se zmást, že i po nahrání bude stále zobrazen stav „Uploading…“, pokud je soubor zobrazen v „Uploaded files“, nahrál se.) V případě, že odevzdáš postupně víc verzí, budeme hodnotit poslední odevzdanou verzi. Odevzdání je možné do pondělí 22.3.2021, 23:59.

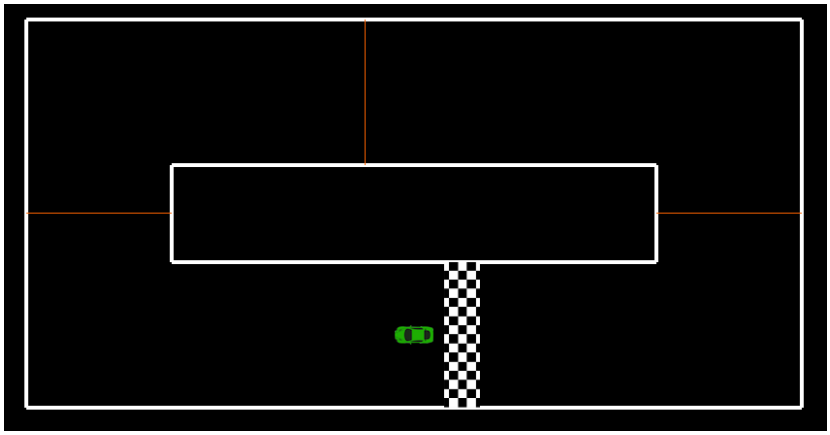
Hodně štěstí přejí DDMP6 a Eaton!

Powering Business Worldwide

Přílohy

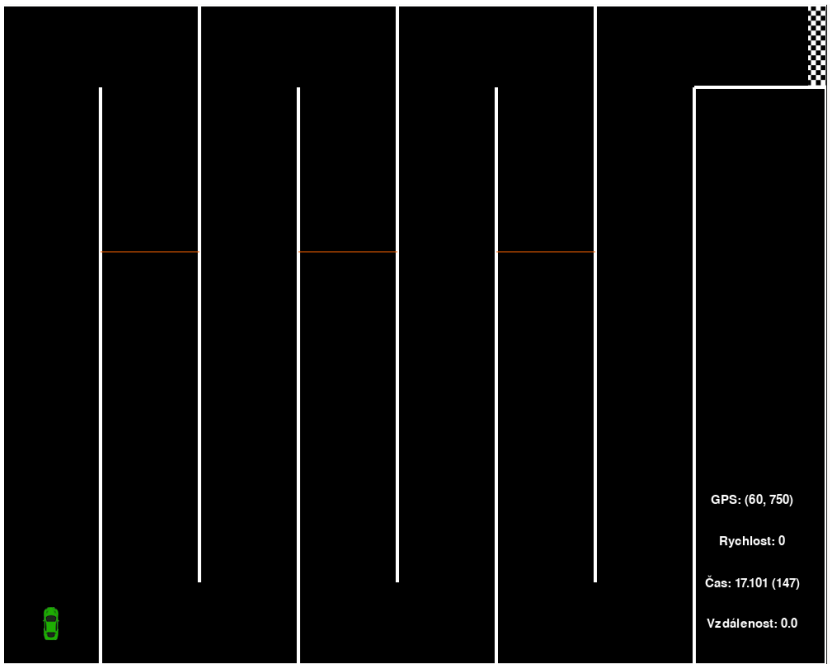
Pozor! Při jízdě je potřeba zachovávat **dostatečnou vzdálenost od překážek** - stěn a to i při dojezdu do cíle. Zatáčky je třeba objíždět s větší rezervou než rovné stěny. Cílem je podobně jako v reálném světě udržovat od překážek bezpečný odstup a nikoli je „hoblovat“. Pokud se tedy setkáváte s výbuchy auta, ačkoli ještě nedošlo přímo k nárazu, je to pravděpodobně z tohoto důvodu.

Mapy

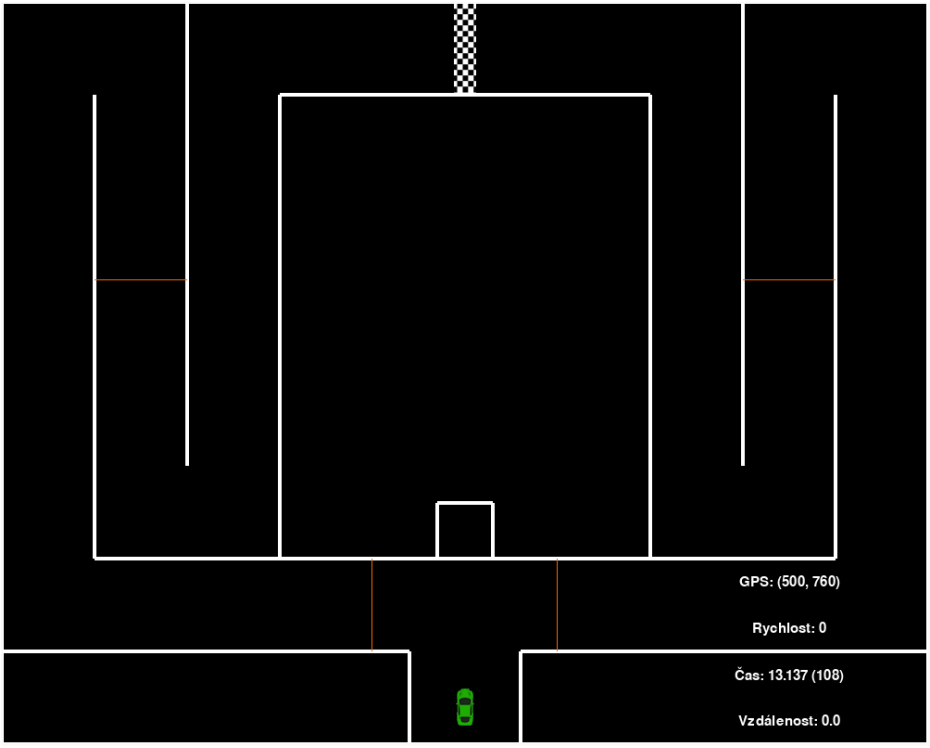


Mapa č. 1

Mapa č. 2



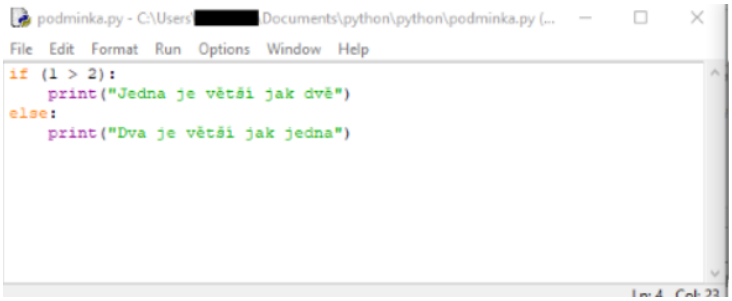
Mapa č. 3



Rady a odkazy na Python

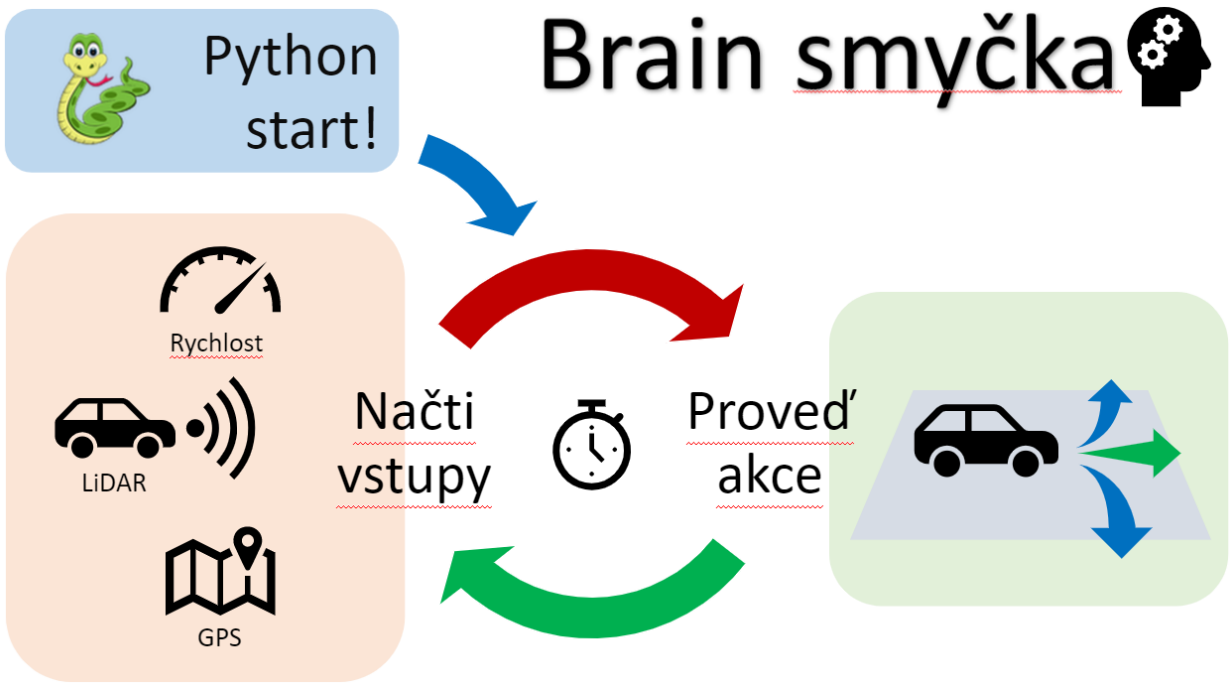
Python je docela jednoduchý programovací jazyk. Je docela rozšířený a používá jej tak mnoho lidí po celém světě a navíc se ti jeho znalost může hodit i v budoucnu až budeš hledat svou práci. Proto je dobré, když se o něm něco naučíš.

Protože se Python používá všude po světě, tak jsou základní příkazy v angličtině. Ale svoje funkce a další věci si už můžeš pojmenovat i v češtině. Také je potřeba vědět, že se v Pythonu část kódu, která se má vykonat společně odsadí o daný počet mezer (stačí použít klávesu Tab). V příkladu níže zkouším jestli (**if**) je jedno číslo větší jak druhé a podle výsledku porovnání vytisknu výsledek slovně (uvádí se v uvozovkách).



Pro další pokračování bude nejlepší se podívat na návody, kterých je na internetu plno, například Python v ČR: Pro začátečníky anebo Začátečnický kurz (python.cz)

Vysvětlení Brain / mozku



Popis orientace autíčka

Aktuální orientaci autíčka získáš pomocí příkazu self.getCarDirection()

self.getCarDirection() == 0 

self.getCarDirection() == 90 

self.getCarDirection() == 180 

self.getCarDirection() == 270 

EATON
Powering Business Worldwide

BÍLÁ HORA DDM PRAHA 6

Popis lidaru

Lidar je laserový radar, který snímá překážky před autíčkem v rozsahu 0 – 180° a předměty ve vzdálenosti 0-300 (0 je u autíčka, 300 znamená, že žádný předmět nebyl nalezen).

Zde jsou dva příklady hodnot, které vrátí příkaz getLidarData():