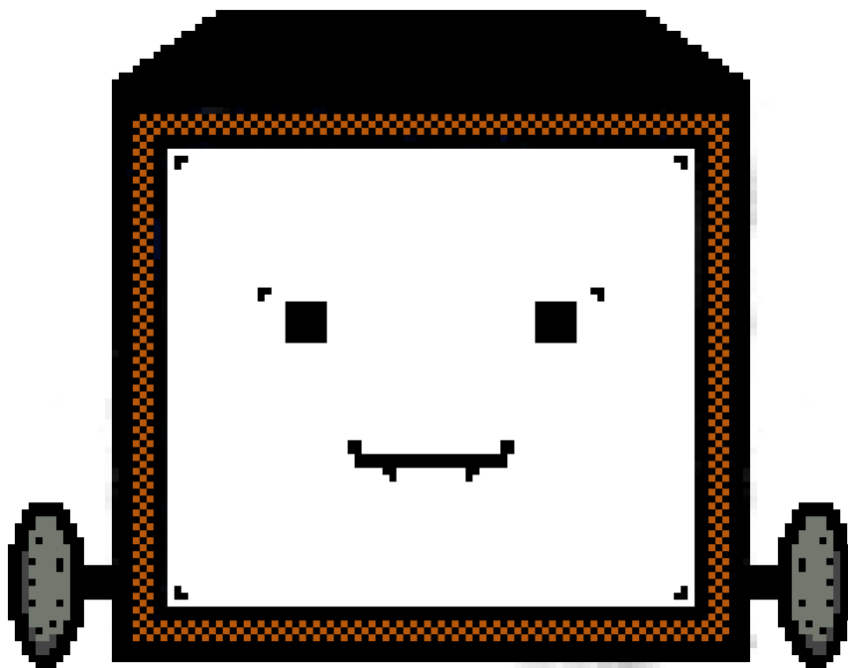




Gymnázium, Praha 6, Arabská 14

Obor programování, vyučující Mgr. Jan Lana



Logibots

Egor Čekunov

Květen 2024

Prohlašuji, že jsem jediným autorem tohoto projektu, všechny citace jsou řádně označené a všechna použitá literatura a další zdroje jsou v práci uvedené. Tímto dle zákona 121/2000 Sb. (tzv. Autorský zákon) ve znění pozdějších předpisů uděluji bezúplatně škole Gymnázium, Praha 6, Arabská14 oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

V..... dne

Egor Čekunov.....

Obsah

1.	ANOTACE.....	4
2.	VLASTNÍ VZTAH K PRÁCI	5
3.	POUŽITÉ TECHNOLOGIE	5
3.1.	GODOT ENGINE	5
3.1.1.	<i>Princip fungování Godotu.....</i>	<i>5</i>
3.2.	DJANGO	6
3.3.	KRITA	6
3.4.	FL STUDIO	7
4.	HERNÍ KOMPONENTY	7
4.1.	MENU SCÉNY.....	7
	<i>Hlavní menu.....</i>	<i>7</i>
4.1.1.	<i>Výběr profilu.....</i>	<i>8</i>
4.1.2.	<i>Výběr úrovně</i>	<i>8</i>
4.1.3.	<i>Online seznam umístění.....</i>	<i>9</i>
4.1.4.	<i>Pomocná scéna.....</i>	<i>9</i>
4.2.	LEVEL SCÉNA	10
4.2.1.	<i>Jednoduchá krabice.....</i>	<i>10</i>
4.2.2.	<i>Reproduktor.....</i>	<i>10</i>
4.2.3.	<i>Mikrofon.....</i>	<i>11</i>
4.2.1.	<i>Nášlapná deska</i>	<i>11</i>
4.2.2.	<i>Robot.....</i>	<i>11</i>
5.	TEMATICKÉ ROZDĚLENÍ ÚROVNÍ.....	11
5.1.	SVĚT POHYBU	12
5.2.	SVĚT POČÍTÁNÍ	12
5.3.	SVĚT ALGORITMU	12
6.	VLASTNÍ SKRIPTOVACÍ JAZYK	13
6.1.	OPERACE S PAMĚTÍ	13
6.2.	POHYB	13
6.3.	KOMUNIKACE	13
6.4.	POČÍTÁNÍ.....	14
6.5.	SKOKY.....	14
6.6.	TYPY SKOKU	14
6.7.	FUNKCE.....	15
7.	LOKÁLNÍ UKLÁDÁNÍ DAT	16
8.	SYNCHRONIZACE VE HŘE	16
8.1.	REALIZACE	16
9.	WEBOVÝ SERVER	16
10.	UKÁZKA ÚROVNÍ.....	17
10.1.	LEVEL 7	17
10.2.	LEVEL 14	18
11.	POSTUP INSTALACE.....	18
12.	ZÁVĚR – MŮJ NÁZOR NA PRÁCI.....	18
13.	SEZNAM ZDROJŮ.....	19
14.	SEZNAM OBRÁZKU	20

1. Anotace

Cílem bylo vytvořit hru, ve který uživatel bude řešit logické problémy pomocí programování. Cílem projektu je demonstrace synchronizačních algoritmů. Hráč bude programovat roboty, které budou řešit úlohy ve hře. Hra bude mít 16 úrovní. Cílem hry je řešení logických problému pomocí určitých nástrojů. Programovat se bude v jednoduchém vlastním jazyce. Jde o vlastní interpretovaný skriptovací jazyk. Hráč bude řešit logické problémy typu: sčítání, odčítání, násobení, třídění a další.

2. Vlastní vztah k práci

Cílem bylo vytvořit hru pro programátory a lidi, které baví řešit logické úlohy. Hráč bude programovat roboty, které budou řešit matematické a logické úlohy. Hra obsahuje 16 úrovní, rozdělené do 3. tematických skupin. Prvních 5 úrovní jsou zaměřeny na pohyb robotu a jejich synchronizaci. Dalších 5 jsou zaměřené na schopnost robotu počítat. Finálních 6 úrovní jsou zaměřené na algoritmické problémy.

3. Použité technologie

3.1. Godot engine

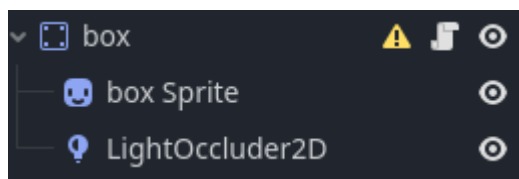


Obrázek 1 Godot logo https://upload.wikimedia.org/wikipedia/commons/thumb/5/5a/Godot_logo.svg/1200px-Godot_logo.svg.png

Godot je herní engine, který umožňuje vytvářet 2D a 3D hry. Funguje na principu uzlů, které se skládají do stromové struktury. Každý uzel představuje určitý aspekt hry, jako je postava, objekt, zvuk nebo efekt. Jedná se o otevřený software udržovaný komunitou, který má volně přístupný zdrojový kód. Godot vznikl v roce 2014, ale popularitu začal nabývat jen v nedávné současnosti. Na rozdíl od jiných populárních herních engineů Godot je velice odlehčený, samotný engine zabírá kolem 50 MB. Například Unity engine zabírá kolem 15 GB a Unreal Engine 5 115 GB. I přestože Godot zabírá 30krát méně paměti na disku oproti Unity, může nabídnout vývojářům slušné nástroje pro vytváření her.

3.1.1. Princip fungování Godotu

Godot funguje na principu uzlů, které se skládají do stromové struktury. Struktury uzlů jsou Scény, které jsou pilířem k vytváření her. Příkladem jednoduché scény v mém projektu je krabice.



Obrázek 2 Scéna Box

Box je scéna, která v sobě má 2 uzly. Box Sprite je uzel typu Sprite2D. Tento uzel slouží k vykreslení obrázku krabice. LightOccluder2D je uzel sloužící k interagování zdroje světla (robot, jiné objekty) a jeho odrazu. Na scénu se dá přihlížet jako na objekt.

Scénu krabice využívám v rodičovské scéně **Level** při generování úrovní ze souboru.

3.1.2. GDScript

Godot Engine podporuje 3 programovací jazyky, a to **GDScript**, **C#** a **C++**. Z oficiální dokumentace jsem zjistil, že nejvíce návodu má jazyk **GDScript**. Jazyk **C#** je také podporovat, ale

ten se doporučuje uživatelům, které mají s ním zkušenost. Jazyk **C++** se využívá především u optimalizací těžkých výpočtů. Moje 2d hra optimalizaci navíc nepotřebuje, takže jazyk **C++** jsem v projektu nepoužíval.

GDScript je programovací jazyk, který byl vytvořen speciálně pro **Godot Engine**. Syntaxi připomíná python a je vhodný pro začátečníky. Před vývojem dané hry jsem zkoušel dělat projekt v **C#**, a proto můžu říct, že **GDScript** působí více odlehčeně. Kód psaný v **GDScript** mi také přijde srozumitelnější než v **C#**.

3.2. Django



Obrázek 3 Django logo <https://www.djangoproject.com/m/img/logos/django-logo-positive.png>

Jedná se o bezplatný framework v programovacím jazyce Python. Django je otevřený software, který byl poprvé vydán v roce 2005. Hlavním cílem Django je usnadnit tvorbu složitých webových stránek založených na databázích. V mém projektu používám Django pro online žebříček hráčů.

3.3. Krita



Obrázek 4 Krita logo <https://upload.wikimedia.org/wikipedia/commons/thumb/7/73/Calligrakrita-base.svg/1200px-Calligrakrita-base.svg.png>

Krita je bezplatný grafický editor zaměřený na digitální umění a animaci. Je součástí kancelářského balíku KOffice od roku 2004. Krita byla vyvinuta komunitou KDE. Všechny obrázky v daném projektu jsou namalované pomocí programu Krita.

3.4. FL Studio



Obrázek 5 FL Studio logo <https://ih1.redbubble.net/image.3793467867.7589/bg,f8f8f8-flat,750x,075,f-pad,750x1000,f8f8f8.jpg>

FL Studio je digitální program pro skládání, aranžování, mixování a editace hudby, která byla vyvinutá belgickou společností **Image-Line**. FL Studio není otevřený software. Ve hře můžete slyšet 6 různých originálních melodií, které jsem složil.

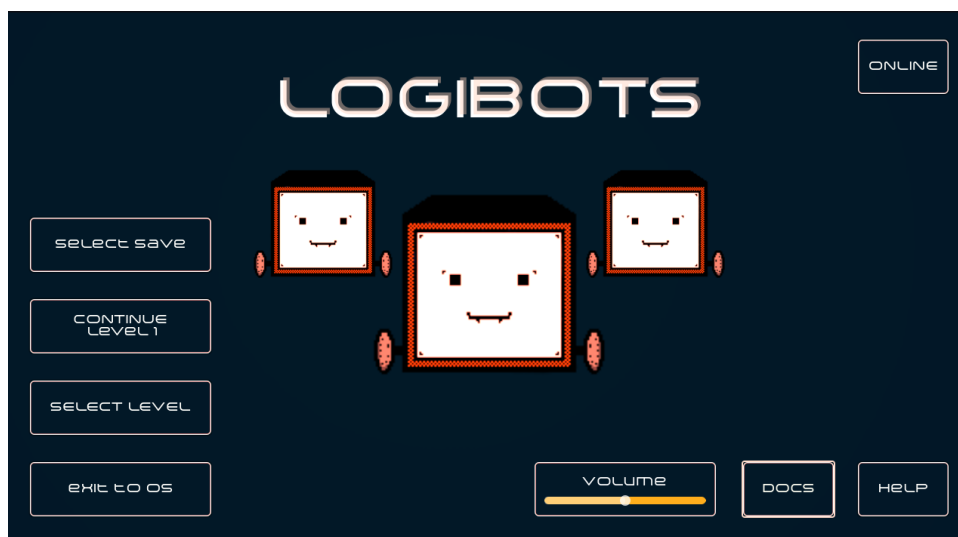
4. Herní komponenty

Nejdůležitější herní komponenty jsou v Godotu scény. Scéna může být jak objekt (například robot), tak i jenom grafický interface (hlavní menu).

4.1. Menu scény

Jedná se o scény s různým obsahem. Celkově ve hře je 5 menu scén.

4.1.1. Hlavní menu



Obrázek 6 Hlavní menu

Je to centrální prvek mého programu. Zde uživatel může vidět název hry, logo a ovládací prvky. První tlačítko přemístí uživatele do scény Výběru profilu. Tlačítko pod ním přemístí uživatele na jeho poslední rozehranou úroveň. **Select level** uživatele přemístí na scénu s výběrem úrovně. Spodní levé tlačítko vypne hru. Vpravo nahoře je tlačítko Online, které přemístí uživatele na scénu s online žebříčkem. Tlačítko Help přemístí na scénu s popisem objektu ve hře. Po zmačknutí tlačítka Docs

vyjede okno s dokumentací mého vlastního jazyka. Vlevo od dokumentace je ovládací prvek hlasitosti.

4.1.2. Výběr profilu



Obrázek 7 Výběr profilu

Výběr profilu je poměrně jednoduchá scéna. Zde si uživatel vybere jeden ze tří profilu. Následný jeho progres ve hře bude ukládán do jedné ze tří složek.

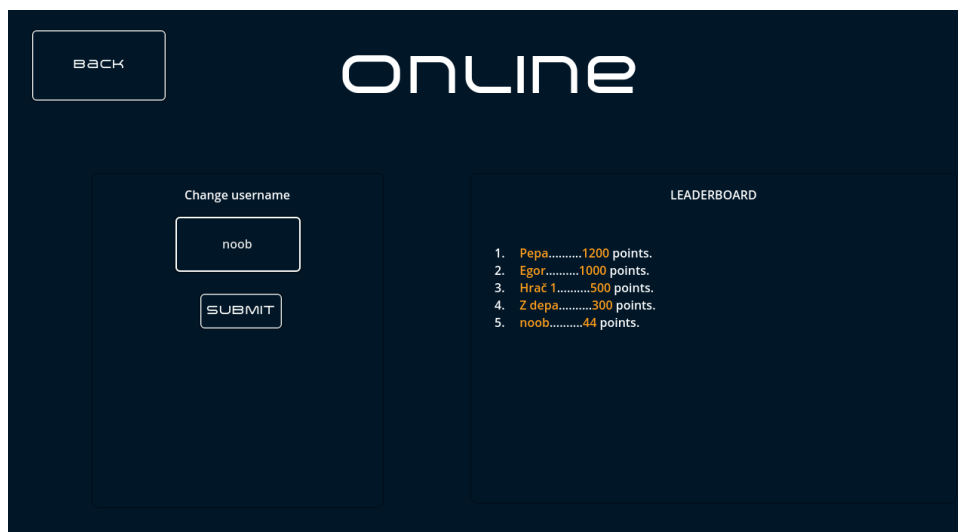
4.1.3. Výběr úrovně



Obrázek 8 Výběr úrovně

Zde uživatel může vybrat úroveň a spustit ho. Na začátku jsou všechny úrovně uzamčené. Takže hráč musí řešit hádanky postupně. Tlačítko Back vrátí uživatele zpátky do hlavního menu.

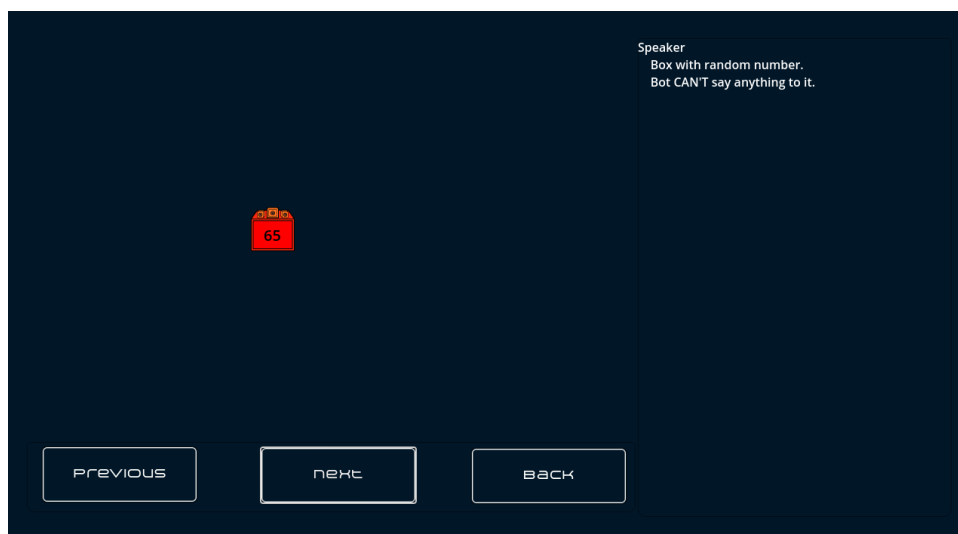
4.1.4. Online seznam umístění



Obrázek 9 Online Scéna

Zde uživatel může vybrat nebo změnit svoje jméno. Vpravo můžete vidět žebříček nejlepších hráčů. Tlačítko **Back** vrátí uživatele zpátky do hlavního menu.

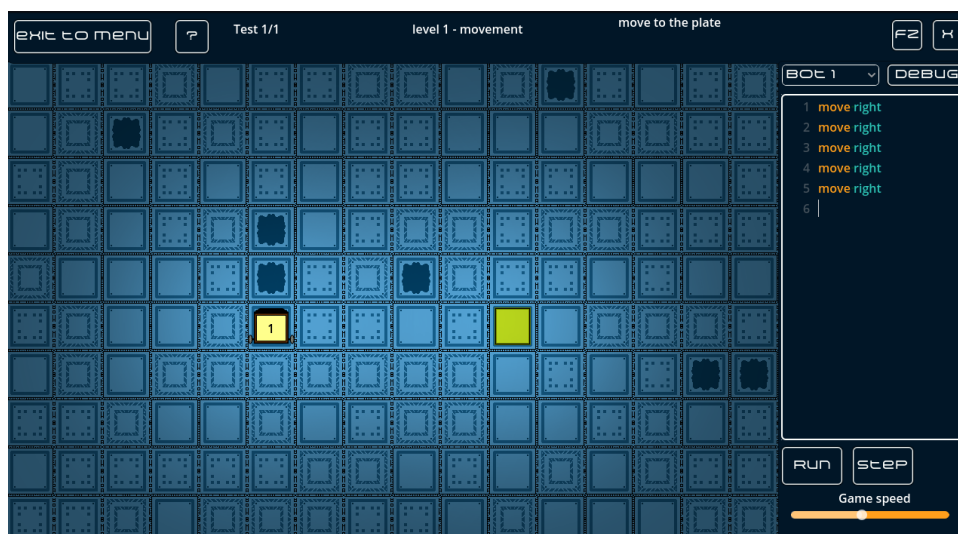
4.1.5. Pomocná scéna



Obrázek 10 Pomocná scéna

Tahle scéna seznámí hráče se všemi objekty ve hře. Zde si může hráč přečíst podrobné informace o objektech.

4.2. Level scéna



Obrázek 11 Level Scéna

Level scéna je hlavní scéna v mém projektu. Vlevo nahoře se nachází tlačítko Exit, které vás přesměruje na Hlavní menu. Tlačítko s otazníkem vám ukáže dokumentaci k mému jazyku. Uprostřed vidíte číslo úrovně a úkol, který máte splnit. Vpravo nahoře tlačítko F2 přepne hru do režimu celé obrazovky. Tlačítko X hru vypne. **Bot 1** je tlačítko pro výběr robotu. Vedle je tlačítko DEBUG, které ukáže více informací pro hráče. Pod ním můžete vidět TextEdit pole, do kterého píšete program. Tlačítko RUN a STEP spouští hru.

Level scéna se skládá z různých objektů. Objekty jsou Godot scény, které se skládají z uzlu.

4.2.1. Jednoduchá krabice



Obrázek 12 Box

Krabice je jednoduchý objekt, který se používá jako překážka. Roboti nemohou krabice posouvat. Při nabourávání robota do krabice robot se zničí a přestane fungovat.

4.2.2. Reprodukční



Obrázek 13 Reprodukční

Reprodukční je krabice, která zobrazuje číslo. Roboti mohou poslouchat (číst) čísla z reproduktoru. Roboti nemohou mluvit do reproduktoru, tím pádem nemůžou hodnotu reproduktoru změnit.

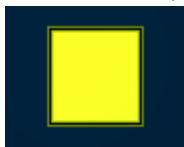
4.2.3. Mikrofon



Obrázek 14 Mikrofon

Mikrofon je krabice, která zobrazuje číslo. Roboti mohou poslouchat (číst) čísla z dané krabice. Roboti mohou mluvit do mikrofonu a tím měnit hodnotu v mikrofonu.

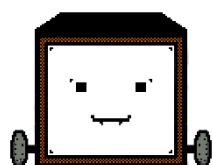
4.2.1. Nášlapná deska



Obrázek 15 Nášlapná deska

Jedná se o nejjednodušší objekt v mé hře. Slouží jen k označení konkrétního pole.

4.2.2. Robot



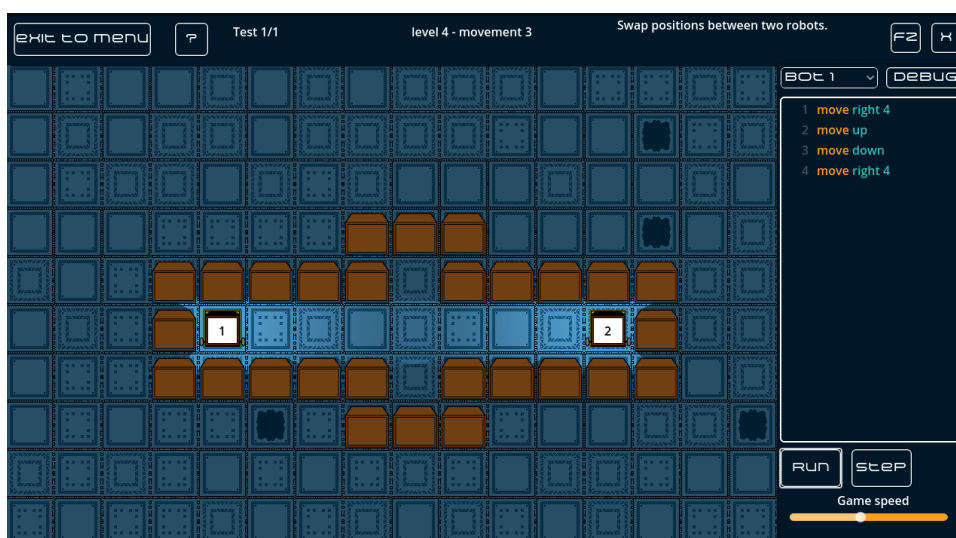
Obrázek 16 Robot

Robot je nejsložitější objekt ve hře. Vykonává příkazy uživatele. Robot si může pamatovat 2 celá čísla. Umí se pohybovat do 4 směru. Má také modul počítání, takže umí přičítat a odečítat celá čísla. Robot umí komunikovat mezi jinými roboty, umí poslouchat z reproduktoru, zapisovat a poslouchat z mikrofonu. Robot je na každé úrovni, nejvíce robotů můžete potkat na úrovni 13.

5. Tematické rozdělení úrovní

Hra Logibots má 16 základních úrovní. Rozdělil jsem úrovně do 3. tematických “světů”. V každém světě hraje jiná hudba a je jiné pozadí. Všechny úrovně jsou uspořádané s narůstající obtížností.

5.1. Svět pohybu

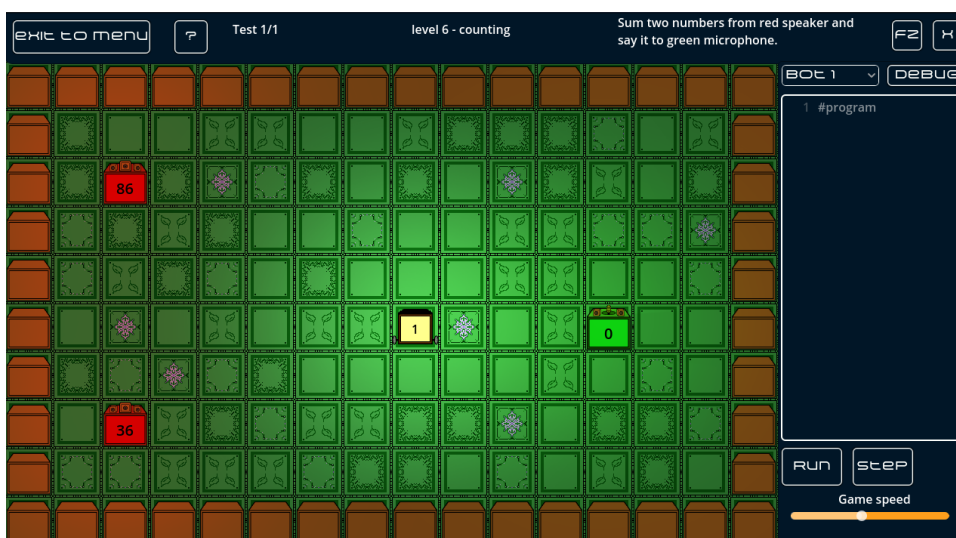


Obrázek 17 Ukázka úrovně 4

Každá z pěti úrovní je zaměřena na pohyb. Seznamuje hráče se schopnosti pohybu robota. Například na úrovni 4 musíte prohodit pozice mezi dvěma roboty. Řeší se to například tak, že první robot počká jeden tah, a potom pojede doprava. Zároveň druhý robot pojede doleva, zahne nahoru tím uvolní místo pro prvního robota. Popojede dolů a doleva.

5.2. Svět počítání

Skládá se z 5 úrovní. Seznamuje s logikou v mém jazyce.

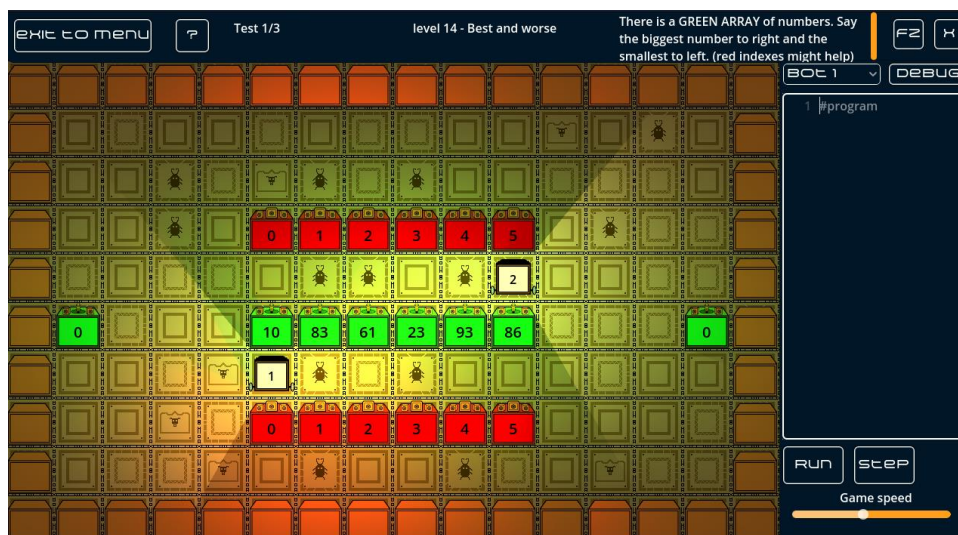


Obrázek 18 Ukázka úrovně 6

Na úrovni 6 hráč musí sečíst dvě čísla z červených reproduktorů. Čísla v reproduktorech jsou náhodná, tedy nejde to spočítat vizuálně a musí se to řešit programem.

5.3. Svět algoritmu

Skládá se z 6 úrovní. Zde jsou nejsložitější úlohy ve hře.



Obrázek 19 Ukázka úrovně 14

Na úrovni 14 musíte najít největší a nejmenší číslo v poli uprostřed. K dispozici máte dva roboty. Nad a pod polem jsou červené reproduktory s indexy v poli. Příklady úrovně a její řešení jsou v kapitole 10.

6. Vlastní skriptovací jazyk

Jedná se o jednoduchý skriptovací jazyk. Každý příkaz je jednořádkový. Každý robot vykonává jeden příkaz za jednu časovou jednotku. Jazyk má příkazy **move**, **say**, **listen**, **add**, **sub** a další.

Je důležité také zmínit, že robot má 2 paměťové složky. Ve hře jsem je pojmenoval jako aktivní a pasivní. Takže robot si může pamatovat 2 nezávislá čísla najednou. Na začátku každé úrovně aktivní a pasivní čísla mají hodnotu 0.

6.1. Operace s pamětí

Tyto příkazy pracují s pamětí robota. Příkaz **swap** prohodí pasivní a aktivní složku mezi sebou. Příkaz **save** uloží hodnotu aktivní složky do pasivní. Využití aktivní a pasivní paměti bude u dalších příkazů.

6.2. Pohyb

Příkazem **move** ovládáte robota. Například „move left“ pohne s robotem doleva. Jezdit s robotem můžete do 4 směrů. Při testování hry jsem zjistil, že stejný příkaz **move** se opakuje několikrát za sebou. Například když chcete pohnout robotem doprava o 10 políček tak musíte napsat desetkrát za sebou „move right“. Proto jsem hru vylepšil další syntaxí „move right 10“. Tyto příkazy jsou pro hru ekvivalentní.

6.3. Komunikace

Pro komunikaci mezi roboty se používají 2 příkazy **say** a **listen**. Když robot komunikuje s mikrofonom nebo reproduktorem tak stačí jenom aby robot stal vedle dané krabice. Poté robot musí provést například příkaz „listen left“ a hodnota v krabici se stane aktivní hodnotou robota.

Když robot potřebuje předat své aktivní číslo, tak provede příkaz „say up“. Tím přepíše hodnotu v mikrofону na svojí aktivní.

Když roboti komunikují mezi sebou tak jeden robot musí poslouchat a druhý robot mluvit.

6.4. Počítání

Robot umí sčítat a odčítat celá čísla. K tomu mu slouží dva příkazy **add** a **sub**. Příkazem „add 2“ přečteme k aktivní složce 2. Příkazem sub odečítáme od aktivního čísla.

6.5. Skoky

Skok je přemístění v kódu. Aby fungovaly skoky, nejdříve uživatel si musí vytvořit kotvy v kódu. Kotva je slovo, které je zakončené dvojtečkou. Poté u skoku si můžeme odkázat na kotvu. Skoky v kódu jsou důležité pro logiku v programu.

```
1 start:
2 move right
3 jump start
```

Obrázek 20 Ukázka skoku

Tady vidíte nekonečný cyklus, ve kterém robot jede doprava.

6.6. Typy skoku

Můj jazyk má 4 typy skoku. Nejjednodušší skok je „jump“, ten se provede vždycky. Skok „jumpz“ se provede jen tehdy, když v aktivní složce je 0. Jinak řádek přeskočí a jde na další příkaz. Skok „jumpg“ se provede, pokud číslo v aktivní složce je kladné. „jumpf“ se provede, když číslo je záporné.

```
1 add 5
2 loop:
3 move right
4 sub 1
5 jumpg loop
```

Obrázek 21 Ukázka užití skoku

Tady se přičte k aktivnímu číslu 5. Pote robot se pohne doprava. Odečte se jednička. Když je aktivní číslo kladné tak přeskočí na kotvu „loop“. Pohne doprava a zase odečte jedničku. Výsledek tohoto kódu je, že robot se pohne doprava 5krát.

6.7. Funkce

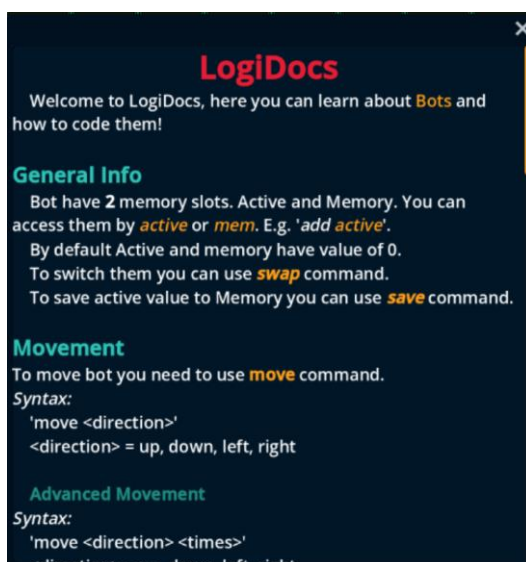
```
1 func funkce:
2 move up
3 move left
4 sub 2
5 endfunc
6
7 funkce
8 funkce
```

Obrázek 22 Ukázka funkce

Funkce se definuje klíčovým slovem „func“ a končí dvojtečkou. Uvnitř funkce můžete past libovolný kód kromě funkci. Každá funkce musí být zakončena klíčovým slovem „endfunc“. Když chcete funkci zavolat stačí napsat její název. Při programování funkci jsem využíval datovou strukturu zásobník, která umožňuje v mém jazyce provádět rekurzivní volání funkci.

6.8. Dokumentace

Ve hře Logibots si můžete vždycky podívat do dokumentace jazyka. K dokumentaci najdete přístup v hlavním menu a také po zmáčknutí tlačítka s otazníkem ve hře. V dokumentaci je popsána každá metoda robota, jsou tam ukázky kódu. Barevné ztvárnění jsem dosáhl díky použité technologii **BBCode**.



Obrázek 23 Ukázka dokumentace

7. Lokální ukládání dat

Godot poskytuje dynamickou cestu k lokálnímu uložení. Všechna data ukládám do cesty „user://“. Na platformě Windows je to složka

„C:\Users\user\AppData\Roaming\Godot\app_userdata\Logibots“.

Na platformě Linux je to složka „~/local/share/Logibots“.

Ve složce Logibots generuji dvě složky **Saves** a **levels**. V levels se nachází všechny úrovně v binárním formátu. Informace o úrovni ukládám do datové struktury slovník. Ta se zapisuje do souboru.

Ve složce Saves se nachází 4 složky a soubor settings, který v sobě má informaci o posledním profilu hráče. Ze 4 složek je jedna složka s názvem „online“. V ní se nachází 3 soubory. Jeden si pamatuje vaše jméno, druhý si pamatuje váš počet bodu a třetí vaše heslo. Heslo se generuje automaticky. Jedná se o klič s 500 náhodně vygenerovaných symboly.

Zbylé tři složky v sobě mají kompletní uložení každého profilu. Každý profil si pamatuje poslední úroveň a kód každého robota na každé úrovni.

8. Synchronizace ve hře

Synchronizace ve hře byla jedna ze základních úkolů v mé práci. Dosáhl jsem ji pomocí příkazů **say** a **listen**. Roboti v mém programu jsou poctiví, takže když někomu něco sdělují, tak čekají na druhého. To samé platí i pro poslouchání. Další synchronizační vylepšení, že za jednu jednotku času se vykonává jeden příkaz. Tudiž roboti s identickým kódem vykonávají vše synchronně.

8.1. Realizace

Programově každý robot je objekt. Robot si pamatuje svůj kód, a svojí pozici v kódu. Robot má funkci move, say, listen, add, jump. Před každým provedením příkazu scéna level vystupuje jako plánovač událostí. Zkouší nasimulovat další tah a naplánovat všechny pohyby a činnosti robotu. Když zpracuje všechny roboty a naplánuje její činnosti, zavolá signál „AllBotsAreReady“, po zavolání signálu každý bot značně svůj pohyb a hráč uvidí animaci.

Plánovač ve scéně level kontroluje správnost kódu hráče. Kdyby například někdo měl špatný řádek v kódu, tak plánovač hodí chybu a hráč uvidí v okně informaci kde, na jaké řádce a co se pokazilo.

9. Webový server

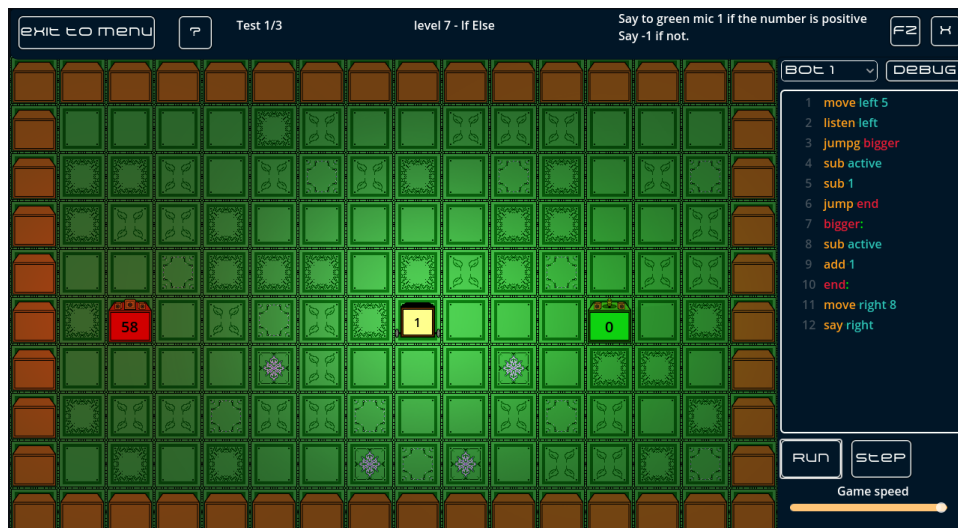
Online přehled nejlepších hráčů je nadstavbou nad projektem, který jsem měl v zadání. Žebříček existuje jako motivace pro hráče, aby mohli zlepšovat vlastní řešení.

Webový server jsem udělal v programovacím jazyce **Python**. Pomocí moderního frameworku **Django** a **Django Rest Frameworku**. Na ukládání záznamu u o hráčích používám databázi **SQLite3**. V databázi mám jednu tabulku **player**, která si eviduje 3 věci, jméno, klíč a body. **Django** projekt má jednu aplikaci „api“, ve které mám definované modely a endpointy.

10. Ukázka úrovní

V této kapitole vám ukazu zajímavé úrovně a jejich řešení. Jestli chcete poctivě vyřešit úrovně, tak můžete tuhle kapitolu přeskočit.

10.1. Úroveň 7



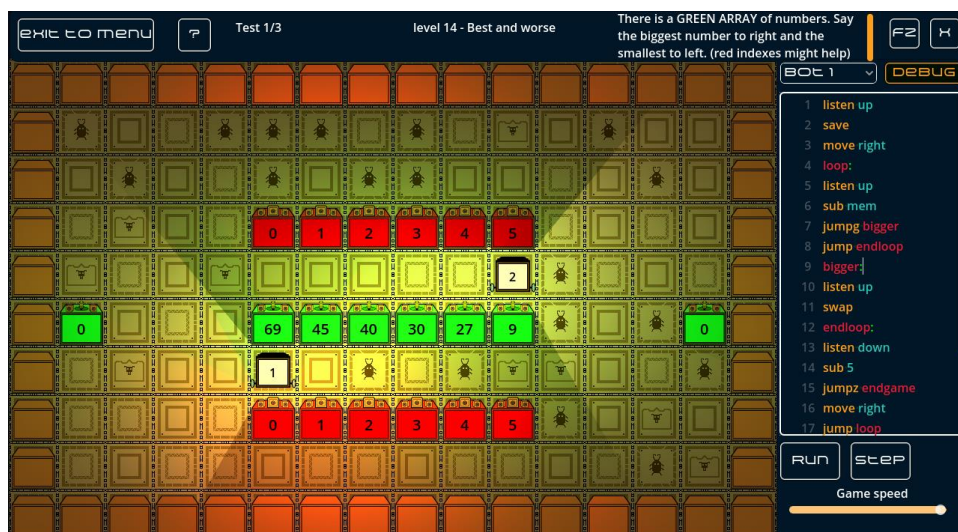
Obrázek 24 Level 7

Na úrovni 7 máte náhodné číslo v červeném reproduktoru. Vy se musíte podívat, zda číslo je větší než nula, nebo není. Nyní vám popíšu vlastní řešení. Pro lepší orientaci v kulatých závorkách je řádek v kódu.

Nejdříve provedu příkaz „move left 5“ (1). Tím se robot dostane na políčko vedle reproduktoru. Příkazem „listen left“ (2) načtu číslo z krabice do aktivní paměťové složky. Příkaz „jumpg“ (3) se provede jen tehdy, když aktivní číslo je kladné. Tedy přeskočíme na řádek 7. následně odečteme od aktivního čísla aktivní číslo (4 nebo 8), tím ho vynulujeme. Příkazem „add 1“ (9) přičteme k nule jedničku. Dále už jenom dojedeme k mikrofonu (11) a řekneme mu číslo (12). Kdyby náhodou číslo se rovnalo nule nebo bylo záporné tak 3. řádek se přeskočí. Poté se vynuluje číslo a odečte se jednička (5). Na řádku 6 přeskočíme na kotvu „end“.

Při programování této úrovně jsem narazil na problém řešení hrubou silou. A proto jsem přišel s vylepšením scény Level. Přidal jsem Testy. Tím pádem nejde uhádnout výsledek. Na úrovni 7 mám tři testy, a to se záporným číslem, nulou a kladným číslem.

10.2. Úroveň 14



Obrázek 25 Level 14

Na úrovni 14 máte pole náhodných čísel. Cílem je najít nejmenší a největší číslo v poli. Nyní vám popíšu vlastní řešení. Pro zjednodušení budu popisovat program pro nalezení největšího čísla. Nejmenší číslo jde najít analogicky. Pro lepší orientaci v závorkách bude řádek programu.

Přečtu číslo nad sebou (1), uložím si je do pasivní paměti (2). Popojedu doprava (3) a dostavám se pod kotvu „loop“. Načtu číslo nad sebou (5). Odečtu od něj pasivní paměť (6) (tam se nacházelo první číslo). Když rozdíl je větší než nula, tak to znamená, že poslední číslo co robot přečetl je větší. Načtu to číslo do aktivní paměti (10) a uložím do pasivní (11). Kdyby rozdíl byl menší než jedna tak krok s uložením do paměti se přeskočí (8). Následně je kontrola, zda nejsem na konci pole (12). Přečtu číslo pod robotem (13) a odečtu 5 (14). Když výsledek je 0, tak to znamená že robot je na konci pole (15). Když je to nula tak se „loop“ opustí a jde na „endgame“ kotvu (18). Tam se robot popojede doprava a zapíše výsledek do mikrofonu. Když to nula není tak se pohne doprava (16) a přeskočí na kotvu „loop“ (17).

11. Postup instalace

V Git repozitáři můžete kliknout na odkaz a stáhnout verzi pro váš počítač. Stažený soubor je spustitelný. Hra je kompatibilní s Windows (Logibot.exe) a Linux (Logibot.x86_64).

12. Závěr – Můj názor na práci

S projektem jsem více než spokojen. Splnil jsem stanovené zadání. Projekt mě naučil pracovat v Godot Game Engine a zanechal dobrý pocit o programování her v Godotu.

Jako nadstavbu jsem udělal jednoduchý webový server, který hostuji na studentském vývojovém serveru, pomocí systemd programu.

Všechno, až na font jsem ztvárnil sám. Naučil jsem se lip používat program **Krita**. Pozadí každé úrovně je vygenerováno náhodným způsobem. Každý svět má sadu mnou namalovaných čtverců a ty se náhodně vybírají.

Vytváření audia do vlastní hry bylo zábavné a kreativní. Používal jsem základní nástroje v programu **FL Studio**. Tvoření hudby bylo časově náročné, a proto jsem využíval některé skladby z minulých let.

13. Seznam zdrojů

- [1] "Godot Engine - Free and open source 2D and 3D game engine." 01 3.. 2024,
<https://godotengine.org/>.
- [2] "Your first 2D game — Godot Engine (stable) documentation in English."
https://docs.godotengine.org/en/stable/getting_started/first_2d_game/index.html.
- [3] "Krita | Digital Painting. Creative Freedom.." <https://krita.org/en/>.
- [4] "Getting started | Django documentation | Django."
<https://docs.djangoproject.com/en/5.0/intro/>.
- [5] "Home - Django REST framework." <https://www.django-rest-framework.org/>.
- [6] "FL Studio [Official] | Overview." <https://www.image-line.com/fl-studio/>.

14. Seznam obrázku

Obrázek 1 Godot logo

[https://upload.wikimedia.org/wikipedia/commons/thumb/5/5a/Godot_logo.svg/1200px-](https://upload.wikimedia.org/wikipedia/commons/thumb/5/5a/Godot_logo.svg/1200px-Godot_logo.svg.png)

Godot_logo.svg.png 5

Obrázek 2 Scéna Box 5

Obrázek 3 Django logo <https://www.djangoproject.com/m/img/logos/django-logo-positive.png>..... 6

Obrázek 4 Krita logo <https://upload.wikimedia.org/wikipedia/commons/thumb/7/73/Calligrakrita-base.svg/1200px-Calligrakrita-base.svg.png> 6

Obrázek 5 FL Studio logo <https://ih1.redbubble.net/image.3793467867.7589/bg,f8f8f8-flat,750x,075,f-pad,750x1000,f8f8f8.jpg>..... 7

Obrázek 6 Hlavní menu 7

Obrázek 7 Výběr profilu 8

Obrázek 8 Výběr úrovně 8

Obrázek 9 Online Scéna..... 9

Obrázek 10 Pomocná scéna 9

Obrázek 11 Level Scéna 10

Obrázek 12 Box..... 10

Obrázek 13 Reproduktor 10

Obrázek 14 Mikrofon..... 11

Obrázek 15 Nášlapná deska 11

Obrázek 16 Robot..... 11

Obrázek 17 Ukázka úrovně 4..... 12

Obrázek 18 Ukázka úrovně 6..... 12

Obrázek 19 Ukázka úrovně 14..... 13

Obrázek 20 Ukázka skoku 14

Obrázek 21 Ukázka užití skoku 14

Obrázek 22 Ukázka funkce 15

Obrázek 23 Ukázka dokumentace 15

Obrázek 24 Level 7 17

Obrázek 25 Level 14 18