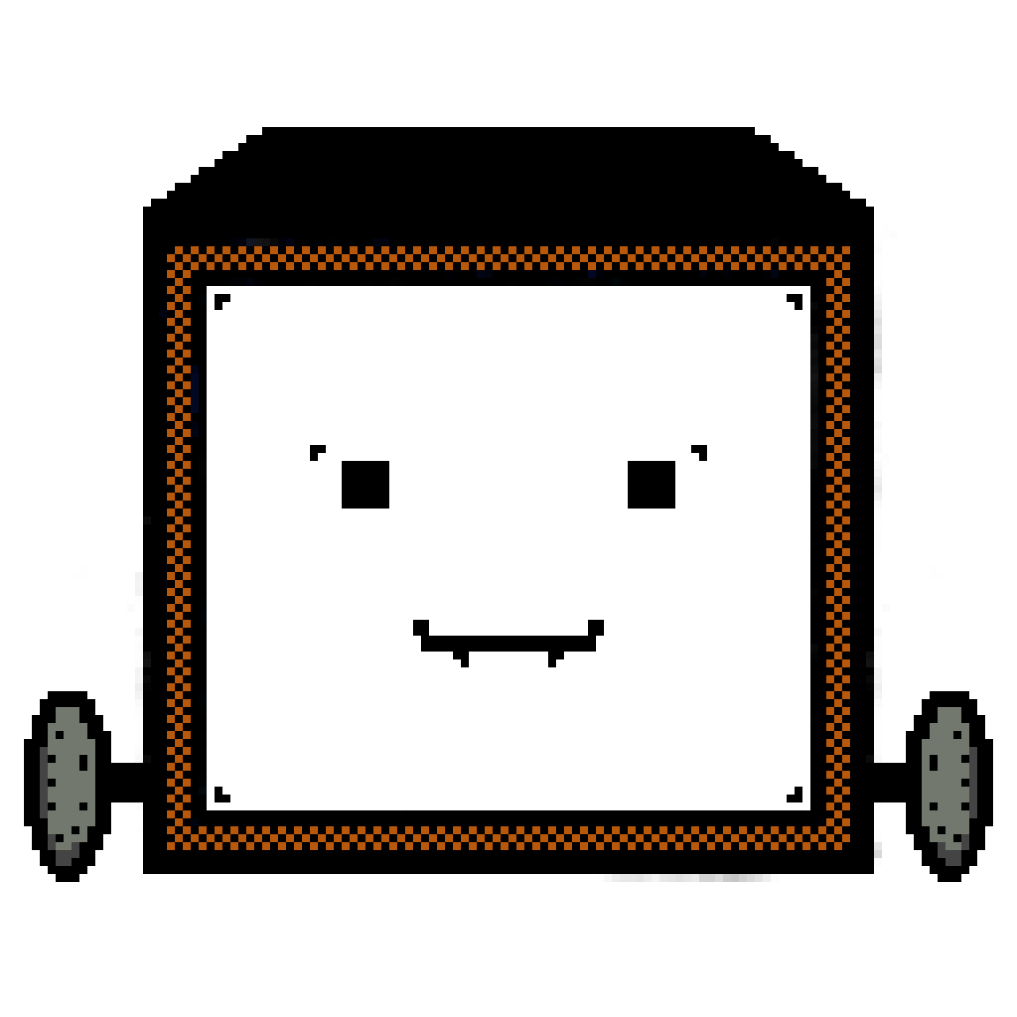
Gymnázium, Praha 6, Arabská 14

Obor programování, vyučující Mgr. Jan Lána



**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](#_Toc162695987)

[2. Vlastní vztah k práci 5](#_Toc162695988)

[3. Použité technologie 6](#_Toc162695989)

[3.1. Godot engine 6](#_Toc162695990)

[3.1.1. Princip fungovaní Godotu 6](#_Toc162695991)

[3.2. Django 7](#_Toc162695992)

[3.3. Krita 7](#_Toc162695993)

[3.4. FL Studio 8](#_Toc162695994)

[4. Herní komponenty 8](#_Toc162695995)

[4.1. Menu scény 8](#_Toc162695996)

[Hlavní menu 8](#_Toc162695997)

[4.1.1. Výběr profilu 9](#_Toc162695998)

[4.1.2. Výběr úrovně 9](#_Toc162695999)

[4.1.3. Online seznam umístěni 10](#_Toc162696000)

[4.1.4. Pomocná scéna 10](#_Toc162696001)

[4.2. Level scéna 11](#_Toc162696002)

[4.2.1. Jednoduchá krabice 11](#_Toc162696003)

[4.2.2. Reproduktor 11](#_Toc162696004)

[4.2.3. Mikrofon 12](#_Toc162696005)

[4.2.1. Nášlapná deska 12](#_Toc162696006)

[4.2.2. Robot 12](#_Toc162696007)

[5. Tematické rozděleni úrovní 12](#_Toc162696008)

[5.1. Svět pohybu 13](#_Toc162696009)

[5.2. Svět počítaní 13](#_Toc162696010)

[5.3. Svět algoritmu 14](#_Toc162696011)

[6. Vlastní skriptovací jazyk 14](#_Toc162696012)

[6.1. Operace s paměti 14](#_Toc162696013)

[6.2. Pohyb 14](#_Toc162696014)

[6.3. Komunikace 14](#_Toc162696015)

[6.4. Počítaní 15](#_Toc162696016)

[6.5. Skoky 15](#_Toc162696017)

[6.6. Typy skoku 15](#_Toc162696018)

[6.7. Funkce 16](#_Toc162696019)

[7. Lokální ukládaní dat 17](#_Toc162696020)

[8. Synchronizace ve hře 17](#_Toc162696021)

[8.1. Realizace 17](#_Toc162696022)

[9. Webový server 17](#_Toc162696023)

[10. Ukázka úrovní 18](#_Toc162696024)

[10.1. Level 7 18](#_Toc162696025)

[10.2. Level 14 19](#_Toc162696026)

[11. Postup instalace 19](#_Toc162696027)

[12. Závěr – Můj názor na práci 19](#_Toc162696028)

[13. Seznam zdrojů 20](#_Toc162696029)

[14. Seznam obrázku 21](#_Toc162696030)

# Anotace

Cílem bylo vytvořit hru, ve které uživatel bude řešit logické problémy pomocí programováni. 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 pomoci určitých nástrojů. Programovat se bude v jednoduchém vlastním jazyce. Jde o vlastní interpretovaný skriptovací jazyk. Hráč bude řešit logické problémy typu: sčítáni, odčítaní, násobení, třídění a další.

# Annotation

The concept was to create a game in which the user would solve logic problems using programming. The goal of the project is to demonstrate synchronization algorithms. The player will program robots to solve problems in the game. The game will have 16 levels. The purpose of the game is to solve logic problems using certain tools. Programming will be done in a simple custom language. It is a custom interpreted scripting language. The player will solve logic problems like: addition, subtraction, multiplication, sorting and more.

# 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 zaměřený na pohyb robotu a jejích synchronizaci. Dalších 5 jsou zaměřené na schopnost robotu počítat. Finálních 6 úrovni jsou zaměřené na algoritmické problémy.

# Použité technologie

## Godot engine

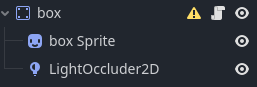


Obrázek 1 Godot logo https://upload.wikimedia.org/wikipedia/commons/thumb/5/5a/Godot\_logo.svg/1200px-Godot\_logo.svg.png

Godot je herní engine, který umožňuje vytvářit 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 enginů **Godot** je velice odlehčený, samotný engine zabírá kolem 50 MB. Například **Unity engine** zabírá kolem 15 GB a **Unreal Engine 5** dokonce 115 GB. I přestože Godot zabírá 30krát méně pamětí na disku oproti Unity, muže nabídnout vývojářům slušné nástroje pro vytváření her.

### Princip fungovaní 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 generováni úrovní ze souboru.

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

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

## 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 otevřeny software, který byl poprvé vydán v roce 2005. Hlavním cílem **Djanga** je usnadnit tvorbu složitých webových stránek založených na databázích. V mém projektu používám **Django** pro online žebříček hráčů.

## Krita

Krita - Wikipedie


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 součásti kancelářského balíku **KOffice** od roku 2004. **Krita** byla vyvinuta komunitou KDE. Všechny obrázky v daném projektu jsou namalované pomocí programu **Krita**.

## 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, mixovaní a editace hudby, která byla vyvinutá belgickou společností **Image-Line**. FL Studio není otevřený software. Ve hře můžete slyšet 6 různých originálních melodii, které jsem složil.

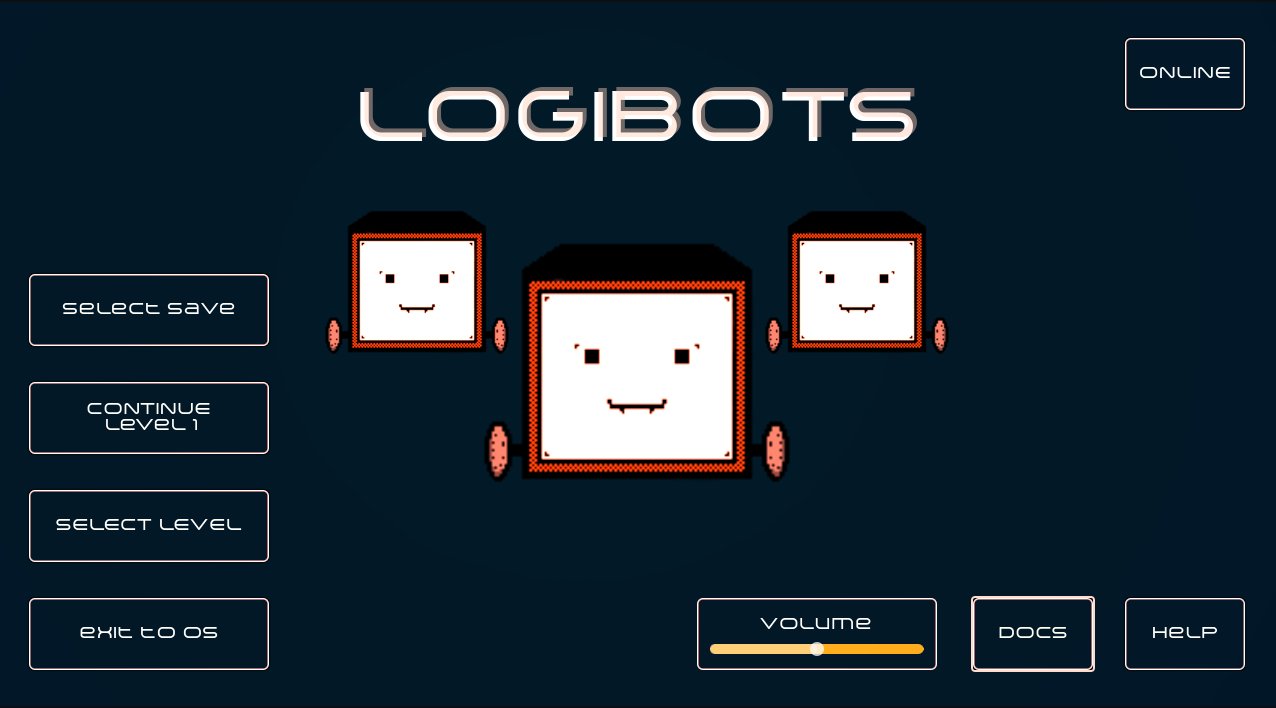
# Herní komponenty

Nejdůležitější herní komponenty jsou v **Godotu** scény. Scéna muže byt jak objekt (například robot), tak i jenom graficky interface (hlavní menu).

## Menu scény

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

### Hlavní menu



Obrázek 6 Hlavní menu

Je to centrální prvek mého programu. Zde uživatel muže vidět název hry, logo a ovládací prvky. První tlačítko přemísti uživatele do scény Výběru profilu. Tlačítko pod nim přemísti uživatele na jeho poslední rozehranou úroveň**.** Select leveluživatele přemísti na scénu s výběrem úrovní. Spodní levé tlačítko vypne hru. Vpravo nahoře je tlačítko Online, které přemístí uživatele na scénu s online žebříčkem. Tlačítko Help přemísti na scénu s popisem objektu ve hře. Po zmačknutí tlačítka Docs vyjede okno s dokumentaci mého vlastního jazyka. Vlevo od dokumentace je ovládací prvek hlasitosti.

### 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.

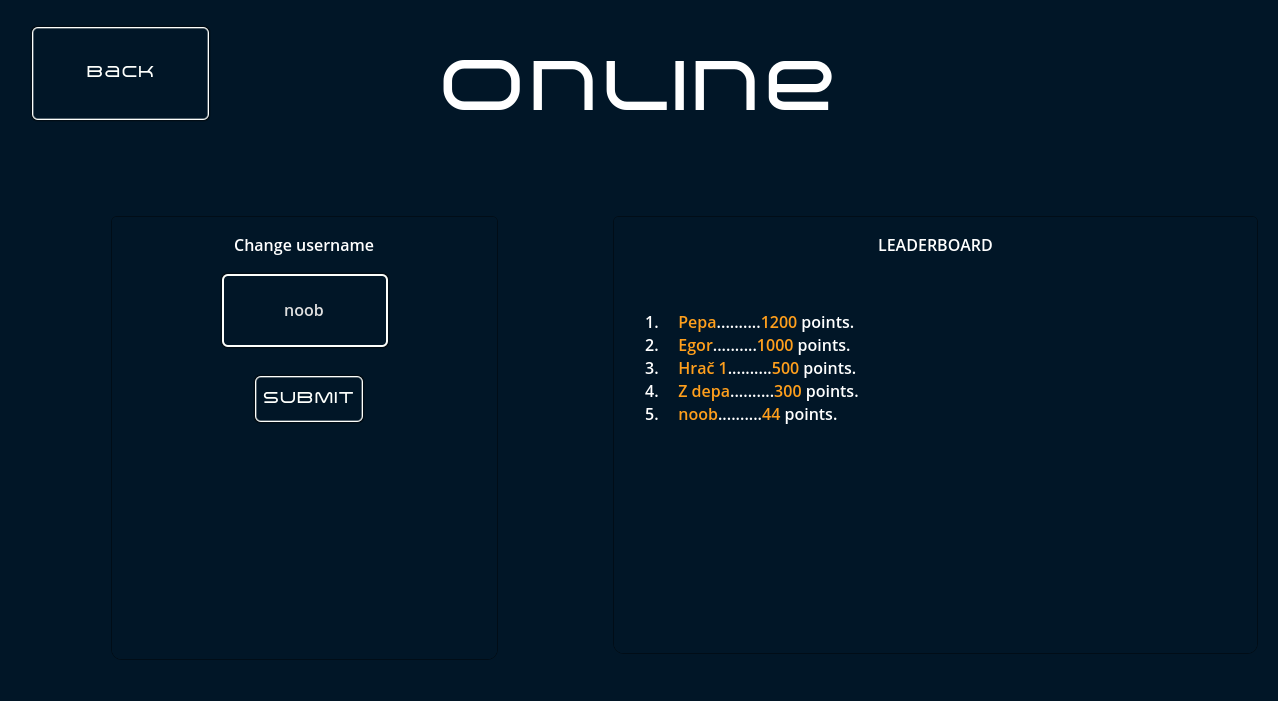
### Výběr úrovně



Obrázek 8 Výběr úrovně

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

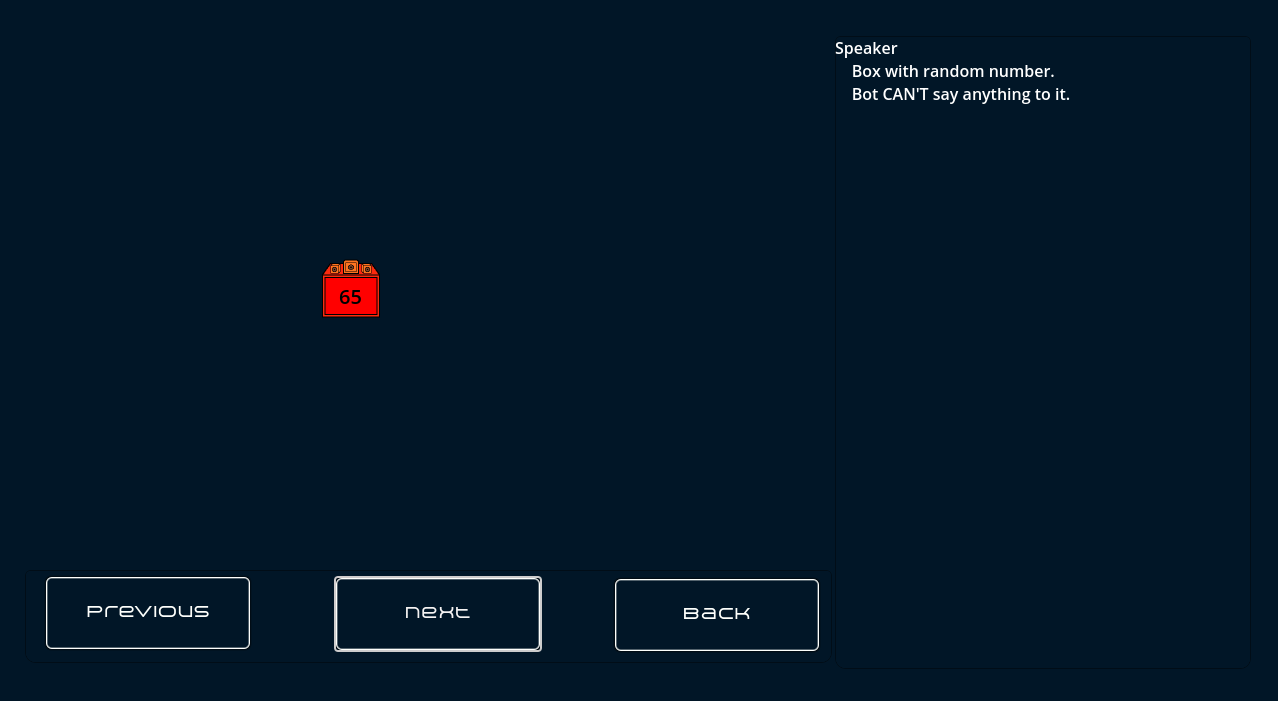
### Online seznam umístěni



Obrázek 9 Online Scéna

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

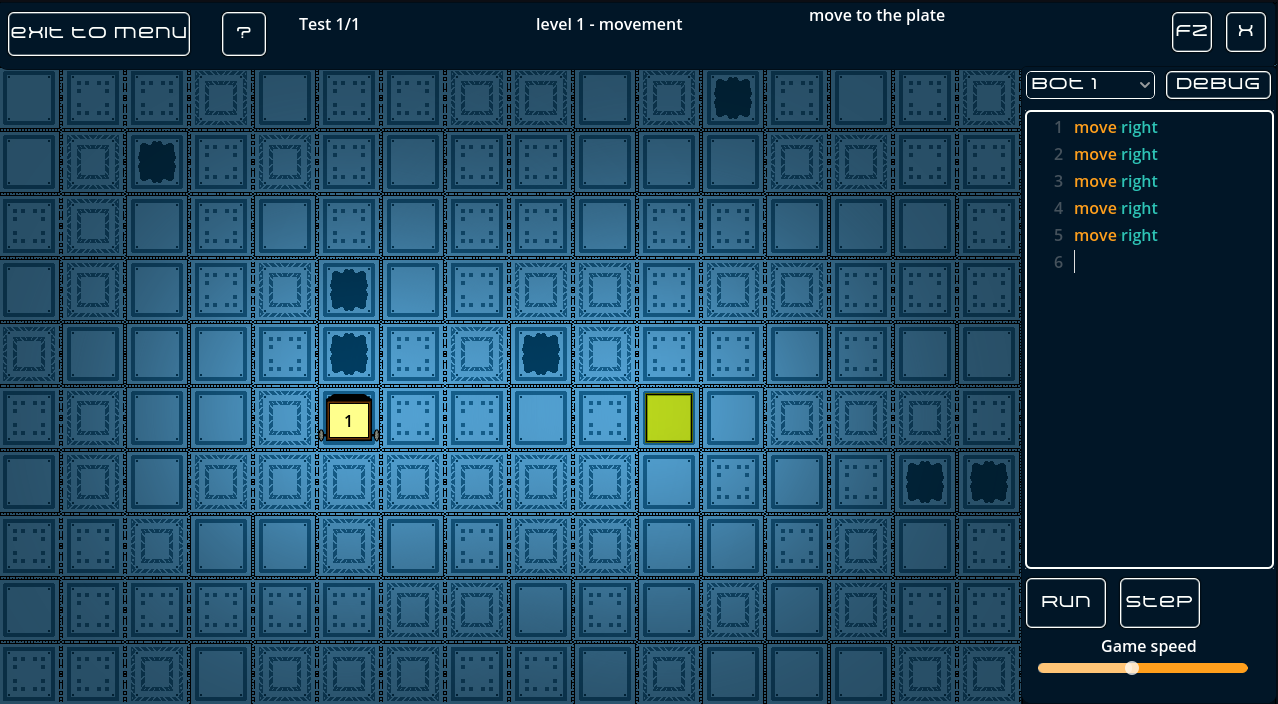
### Pomocná scéna



Obrázek 10 Pomocná scéna

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

## 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 dokumentací k mému jazyku. Uprostřed vidíte číslo úrovní a úkol, který maté splnit. Vpravo nahoře tlačítko F2 přepne hru do režimu celé obrazovky. Tlačítko Xhru vypne. Bot 1jetlačítko pro výběr robotu. Vedle je tlačítko DEBUG, které ukáže vice informaci pro hráče. Pod nim můžete vidět TextEdit pole, do kterého píšete program. Tlačítko RUN a STEP spouští hru.

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

### Jednoduchá krabice



Obrázek 12 Box

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

### Reproduktor



Obrázek 13 Reproduktor

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

### 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.

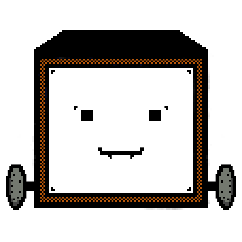
### Nášlapná deska



Obrázek 15 Nášlapná deska

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

### Robot



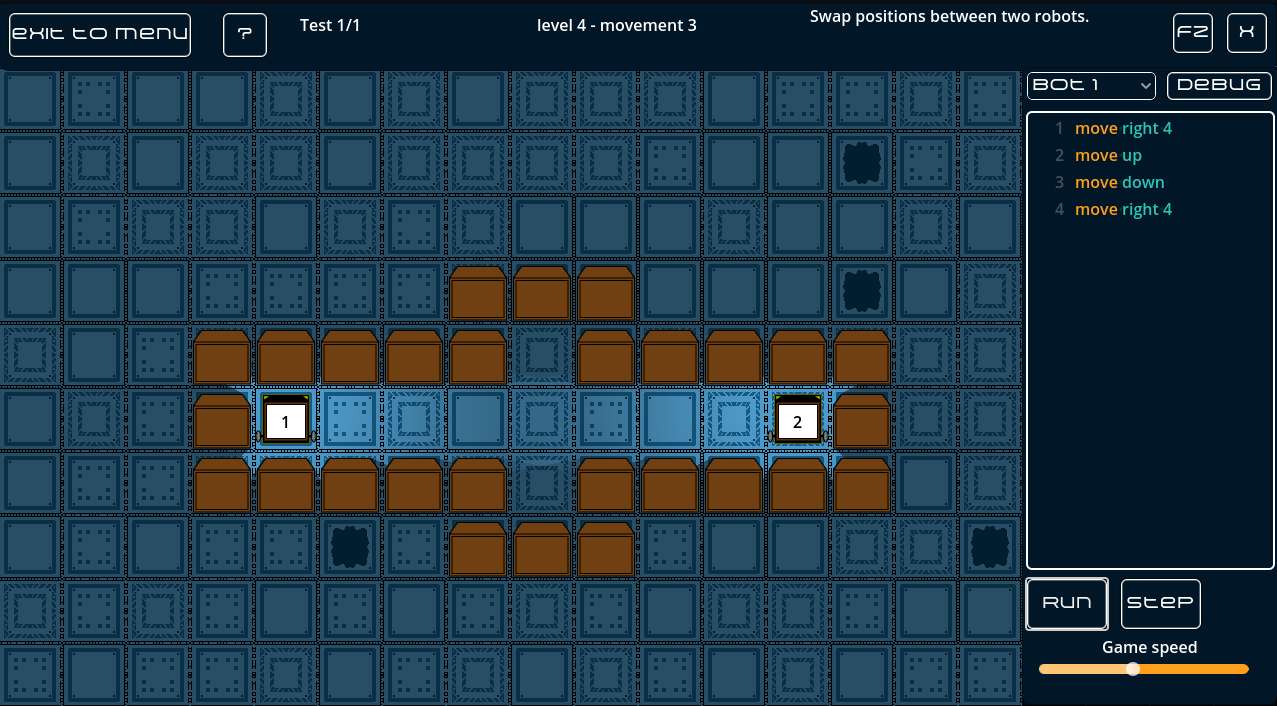
Obrázek 16 Robot

Robot je nejsložitější objekt ve hře. Vykonává příkazy uživatele. Robot si muže pamatovat 2 cela čísla. Umí se pohybovat do 4 směru. Má také modul počítaní, takže umí přičítat a odečítat cela čísla. Robot umí komunikovat mezí jiny roboty, umí poslouchat z reproduktoru, zapisovat a poslouchat z mikrofonu. Robot je na každé úrovní, nejvíce robotu můžete potkat na úrovní 13.

# Tematické rozděleni úrovní

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

## Svět pohybu



Obrázek 17 Ukázka úrovně 4

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

## Svět počítaní

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

Obsah obrázku snímek obrazovky, text, čtverec

Popis byl vytvořen automaticky

Obrázek 18 Ukázka úrovně 6

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

## Svět algoritmu

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

Obsah obrázku text, snímek obrazovky, Multimediální software, software

Popis byl vytvořen automaticky

Obrázek 19 Ukázka úrovně 14

Na úrovní 14 musíte najít největší a nejmenší číslo v poli uprostřed. K dispozici maté dva roboty. Nad a pod polem jsou červené reproduktory s indexy v poli. Příklady úrovní a jejích řešení jsou v kapitole 10 Ukázka úrovní.

# 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 muže pamatovat 2 nezávislá čísla najednou. Na začátku každé úrovní aktivní a pasivní čísla mají hodnotu 0.

## Operace s paměti

Tyto příkazy pracuji s pamětí robota. Příkaz „swap“ prohodí pasivní a aktivní složku mezi sebou. Příkaz „save“ uloží hodnotu aktivní složky do pasivní. Využiti aktivní a pasivní paměti bude u dalších přikážu.

## Pohyb

Příkazem „move“ovládáte robota. Například „move left“ pohne s robotem doleva. Jezdit s robotem můžete do 4 směru. Při testovaní 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 syntaxi „move right 10“. Tyto příkazy jsou pro hru ekvivalentní.

## Komunikace

Pro komunikací mezi roboty se používají 2 příkazy „say“ a „listen“. Když robot komunikuje s mikrofonem 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“. Tim přepíše hodnotu v mikrofonu na svojí aktivní.

Když roboti komunikují mezí sebou tak jeden robot musí poslouchat a druhy robot mluvit.

## Počítaní

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.

## Skoky

Skok je přemístěni 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.

Obsah obrázku Písmo, text, snímek obrazovky, Grafika

Popis byl vytvořen automaticky

Obrázek 20 Ukázka skoku

Tady vidíte nekonečny cyklus, ve kterém robot jede doprava.

## 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. „jumpl“ se provede, když číslo je záporné.

Obsah obrázku text, Písmo, snímek obrazovky, Grafika

Popis byl vytvořen automaticky

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.

## Funkce

Obsah obrázku text, snímek obrazovky, Písmo, Grafika

Popis byl vytvořen automaticky

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. Pří programováni funkci jsem využíval datovou strukturu zásobník, která umožňuje v mém jazyce provádět rekurzivní volaní funkci.

## Dokumentace

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

Obsah obrázku text, snímek obrazovky, software, Písmo

Popis byl vytvořen automaticky

Obrázek 23 Ukázka dokumentace

# Lokální ukládaní dat

Godot poskytuje dynamickou cestu k lokálnímu uložišti. 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 úrovní 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 tří složky v sobě mají kompletní uloženi každého profilu. Každý profil si pamatuje poslední úroveň a kód každého robota na každé úrovní.

# Synchronizace ve hře

Synchronizace ve hře byla jedna ze základních úkolu v mé prací. Dosáhl jsem ji pomocí příkazu **say** a **listen**. Robotí v mém programu jsou poctivý, takže když někomu něco sděluji, tak čekají na druhého. To samé platí i pro poslouchání. Další synchronizační vylepšení, že za jednu jednotku času se vykonává jeden příkaz. Tudíž roboti s identickým kódem vykonávají vše synchronně.

## 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 události. Zkouší nasimulovat další tah a naplánovat všechny pohyby a činnosti robotu. Když zpracuje všechny roboty a naplánuje její činnosti, zavolá signál „AllBotsAreReady“, po zavoláni signálu každý bot značné 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.

# Webový server

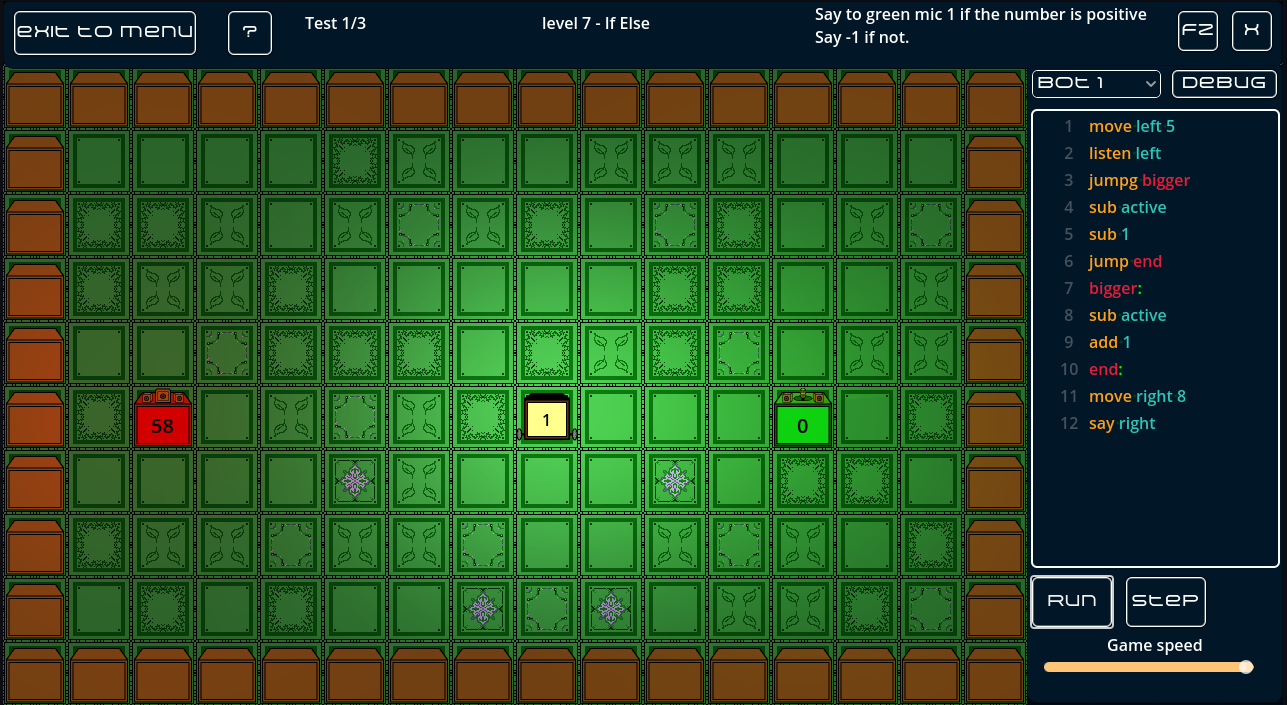
Online přehled nejlepších hráčů je nadstavbou nad projektem, který jsem měl v zadáni. Žebříček existuje jako motivace pro hráče, aby mohli zlepšovat vlastní řešeni.

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

# Ukázka úrovní

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

## Úroveň 7



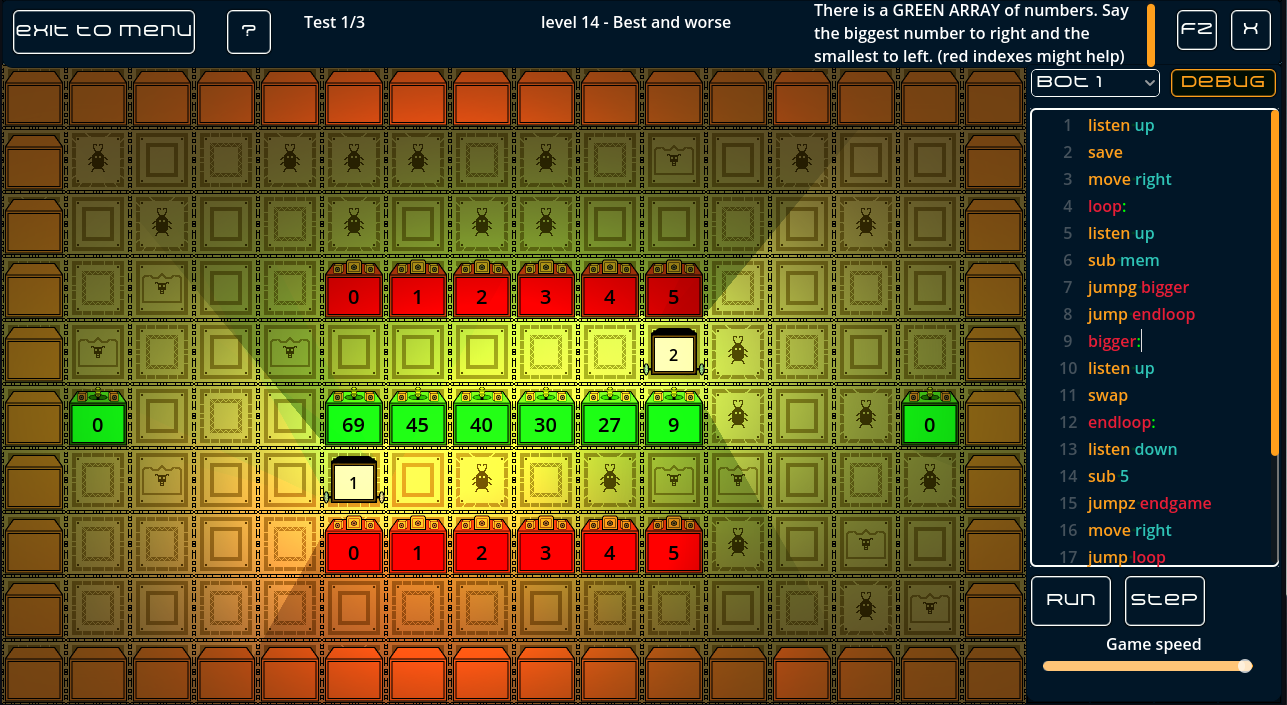
Obrázek 24 Level 7

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

Nejdříve provedu příkaz „move left 5“ (1). Tim se robot dostane na poličko 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) přečteme k nule jedničku. Dále už jenom dojedeme k mikrofonu (11) a řekneme mu čislo (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“.

Pří programovaní této úrovní jsem narazil na problém řešení hrubou silou. A proto jsem přišel s vylepšením scény Level. Přidal jsem Testy. Tim pádem nejde uhádnout výsledek. Na úrovní 7 mam tří testy, a to se záporným číslem, nulou a kladným číslem.

## Úroveň 14



Obrázek 25 Level 14

Na úrovní 14 maté pole náhodných čísel. Cílem je najit nejmenší a největší číslo v poli. Nyní vám popíšu vlastní řešeni. Pro zjednodušení budu popisovat program pro nalezeni největšího čísla. Nejmenší číslo jde najit 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 čilo 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).

# Postup instalace

V Git repositáři můžete kliknout na odkaz a stáhnout verzi pro váš počítač. Staženy soubor je spustitelný. Hra je kompatibilní s **Windows** (Logibot.exe) a **Linux** (Logibot.x86\_64).

# Závěr – Můj názor na práci

S projektem jsem více než spokojen. Splnil jsem stanovené zadáni. Projekt mě naučil pracovat v **Godot Game Enginu** a zanechal dobry pocit o programovaní her v Godotu.

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

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**. Tvořeni hudby bylo časově náročné, a proto jsem využíval některé skladby z minulých let.

# 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/>.

# Seznam obrázku

[Obrázek 1 Godot logo https://upload.wikimedia.org/wikipedia/commons/thumb/5/5a/Godot\_logo.svg/1200px-Godot\_logo.svg.png 6](#_Toc162817047)

[Obrázek 2 Scéna Box 6](#_Toc162817048)

[Obrázek 3 Django logo https://www.djangoproject.com/m/img/logos/django-logo-positive.png 7](#_Toc162817049)

[Obrázek 4 Krita logo https://upload.wikimedia.org/wikipedia/commons/thumb/7/73/Calligrakrita-base.svg/1200px-Calligrakrita-base.svg.png 7](#_Toc162817050)

[Obrázek 5 FL Studio logo https://ih1.redbubble.net/image.3793467867.7589/bg,f8f8f8-flat,750x,075,f-pad,750x1000,f8f8f8.jpg 8](#_Toc162817051)

[Obrázek 6 Hlavní menu 8](#_Toc162817052)

[Obrázek 7 Výběr profilu 9](#_Toc162817053)

[Obrázek 8 Výběr úrovně 9](#_Toc162817054)

[Obrázek 9 Online Scéna 10](#_Toc162817055)

[Obrázek 10 Pomocná scéna 10](#_Toc162817056)

[Obrázek 11 Level Scéna 11](#_Toc162817057)

[Obrázek 12 Box 11](#_Toc162817058)

[Obrázek 13 Reproduktor 11](#_Toc162817059)

[Obrázek 14 Mikrofon 12](#_Toc162817060)

[Obrázek 15 Nášlapná deska 12](#_Toc162817061)

[Obrázek 16 Robot 12](#_Toc162817062)

[Obrázek 17 Ukázka úrovně 4 13](#_Toc162817063)

[Obrázek 18 Ukázka úrovně 6 13](#_Toc162817064)

[Obrázek 19 Ukázka úrovně 14 14](#_Toc162817065)

[Obrázek 20 Ukázka skoku 15](#_Toc162817066)

[Obrázek 21 Ukázka užití skoku 15](#_Toc162817067)

[Obrázek 22 Ukázka funkce 16](#_Toc162817068)

[Obrázek 23 Ukázka dokumentace 16](#_Toc162817069)

[Obrázek 24 Level 7 18](#_Toc162817070)

[Obrázek 25 Level 14 19](#_Toc162817071)