

Gymnázium  
Veľká okružná 22, 010 01 Žilina

# Botanica - simulátor rastlín na báze celulárnych automatov

Stredoškolská odborná činnosť

Č. odboru: 11

Riešitelia: František Knapec, Michael Sklenka, Marek Beňo  
Ročník štúdia: 4.

Mesto: Žilina  
Rok: 2025

Gymnázium  
Veľká okružná 22, 010 01 Žilina

# Botanica - simulátor rastlín na báze celulárnych automatov

Stredoškolská odborná činnosť

Č. odboru: 11

Riešitelia: František Knapec, Michael Sklenka, Marek Beňo

Ročník štúdia: 4.

Školiteľ: PaedDr. Jana Pekárová, PhD.

Konzultant: Ing. Tomáš Milet, PhD.

Mesto: Žilina

Rok: 2025

**Čestné vyhlásenie:**

Prehlasujeme, že sme prácu na tému Botanica - simulátor rastlín na báze celulárnych automatov vypracovali samostatne s použitím literatúry uvedenej v zozname použitej literatúry. Zároveň prehlasujeme, že sme predloženú písomnú prácu neprihlásili a ani neprezentovali v žiadnej inej súťaži, ktorá je pod gestorstvom MŠM VVaŠ SR. Sme si vedomí zákonných dôsledkov, ak v nej uvedené údaje nie sú pravdivé.

# Obsah

<b>Úvod</b>	<b>5</b>
<b>1 Problematika a prehľad literatúry</b>	<b>7</b>
1.1 Spôsoby simulovania rastlín . . . . .	7
1.2 Celulárne automaty . . . . .	9
1.2.1 Príklady na klasické CA . . . . .	9
1.2.2 Príklady CA zamerané na simuláciu rastlín . . . . .	10
<b>2 Ciele práce</b>	<b>11</b>
<b>3 Materiál a metodika</b>	<b>12</b>
<b>4 Aplikácia</b>	<b>13</b>
4.1 Svet . . . . .	13
4.1.1 Perlinov šum . . . . .	13
4.1.2 Vytváranie terénu . . . . .	14
4.1.3 Živiny . . . . .	14
4.2 Rastliny . . . . .	15
4.2.1 Premenné v rastlinách . . . . .	15
4.2.2 Rastlinné voxely . . . . .	15
4.2.3 Genetika . . . . .	16
4.3 Algoritmus . . . . .	16
4.3.1 Vznik rastlín . . . . .	16
4.3.2 Redistribúcia živín . . . . .	17
4.3.3 Zasadenie rastlín . . . . .	17
4.3.4 Produkcia . . . . .	17
4.3.5 Smrť . . . . .	18
4.3.6 Kalkulácia bonusov zo živín . . . . .	18
<b>5 Výsledky práce</b>	<b>19</b>
<b>6 Diskusia</b>	<b>20</b>
<b>7 Závery práce</b>	<b>21</b>

<b>8</b>	<b>Zhrnutie</b>	<b>22</b>
<b>9</b>	<b>Zoznam použitej literatúry</b>	<b>23</b>

# Úvod

# 1 Problematika a prehľad literatúry

Táto práca sa zaoberá implementovaním rastlín do digitálnej podoby a následným simulovaním ich správania relatívne k podmienkam, v ktorých sa nachádzajú, ako aj k iným rastlinám v ich okolí. Do tejto práce patrí aj vyobrazenie rastlín v trojdimenzionálnom priestore. Pôjde aj o schopnosť dlhodobého pozorovania ekosystému naprieč časom.

Ľudia k problému tvorenia digitálnych rastlín postupovali rôznymi spôsobmi, avšak väčšina z nich len realistické rastliny vykresľovali, ako je napríklad známy príklad L-systémov,<sup>1</sup> ktorý je založený na opisovaní štruktúry rastlín sériou pravidiel, ako je napr. otočenie, vytvorenie úsečky alebo vrátenie sa na určitú pozíciu. Ďalej sa štruktúra zdokonaľuje za pomoci algoritmu, ktorý nahrádza časti jej série pravidiel za iné, dopredu určené a detailnejšie série. Tento postup dokáže vytvoriť celkom realistické rastliny ako je znázornené na obr. 1.



Obr. 1: Príklad trávy vygenerovanej s použitím L-systému v 3D

(zdroj: <https://en.wikipedia.org/wiki/L-system>)

Toto riešenie a jemu podobné však dané rastliny len vytvárajú, no našim cieľom je ich aj simulovať.

## 1.1 Spôsoby simulovania rastlín

Kôli samotnej simulácii vzniklo viacero metód simulovania s vlastnými kladmi, ale aj zápornmi. Medzi takéto metódy patria:

- **Fyzikálne založené modelovanie** využívajúce body prepojené pružnými štruktúrami, na ktoré pôsobia rôzne sily. Aj keď realistické, modelovanie neumožňuje rastliny tvoriť, iba simulovať.

---

<sup>1</sup>Lindenmayerov systém - <https://en.wikipedia.org/wiki/L-system>

- **Simulácia pomocou strojového učenia**<sup>2</sup> využíva štatistické algoritmy, pomocou ktorých sa dokáže počas tréningu učiť vzťahy medzi vstupmi a výstupmi.
- **Celulárne automaty**<sup>3</sup> je názov pre matematický model a nástroj pre simuláciu. Jedná sa o starší koncept, ktorý obsahuje štvorcovú sieť so „zafarbenými“ políčkami. Každá farba je určitý stav, ktorý má špecifické správanie závisiace od okolitých políčok.

Fyzikálne založené modelovanie, aj keď mocné, vyžaduje aby rastliny boli vygenerované pred tým, ako začne simulácia. Po spustení simulácie nie sú tieto rastliny schopné sa ďalej vyvíjať. Navyše vyžaduje fyzikálne založenú simuláciu, ktorá je príliš komplikovaná pre túto prácu.

Na druhú stranu, strojové učenie sa stáva čím ďalej tým využívanéjšie pre riešenie všetkých možných problémov. Avšak je tento spôsob náročný na čas a výpočtovú techniku.

Nakoniec Celulárne automaty ponúkajú jednoduchosť a možnosť ich jednoducho modifikovať, čo otvára pole možných využití. Vďaka týmto výhodám je simulácia v tejto práci založená na princípe celulárnych automatoch.

---

<sup>2</sup>Strojové učenie - [https://en.wikipedia.org/wiki/Machine\\_learning](https://en.wikipedia.org/wiki/Machine_learning)

<sup>3</sup>Celulárne automaty - [https://en.wikipedia.org/wiki/Cellular\\_automaton](https://en.wikipedia.org/wiki/Cellular_automaton)



## 1.2 Celulárne automaty

CA sa skladá z týchto základných častí: mriežka (často štvorcová sieť) a bunky tejto mriežky, ktoré majú vlastný stav, pričom každý stav obsahuje určité pravidlá, ktoré zapríčiňujú správanie buniek, a tým celého systému. Tieto časti z CA robia systém, ktorý je jednoduché prispôbiť ktorýmkoľvek požiadavkám.

Klasickým príkladom takéhoto systému je Conwayova hra života.<sup>4</sup> V tejto hre nadobúda každá bunka jednu z dvoch hodnôt: buď je bunka živá, alebo mŕtva.



Obr. 2: Príklad živej a mŕtvej bunky Conwayovej hry života.

Pravidlá definujú správanie bunky v závislosti jej stavu od susedov. Susedov môžeme taktiež definovať 2 spôsobmi v  $N+1$  dimenziách, ak nerátame diagonálne bunky ako susedné, ide o von Neumannove susedstvo, ak rátame aj diagonály, ide o Moorove susedstvo.

Zmena času sa odohráva v diskretných časových odsekoch, toto má za príčinu našu schopnosť sledovať zmeny všetkých stavov buniek naraz.

### 1.2.1 Príklady na klasické CA

Conwayova hra života je pravdepodobne najznámejší príklad celulárnych automatov. Jedná sa o dvojrozmernú štvorcovú mriežku, kde bunky majú dva stavy: „živé“ alebo „mŕtve“, ich stav závisí od stavu okolitých buniek.

Wolframove celulárne automaty kategorizujú správanie jednorozmerných CA. Sú schopné generovať komplexné a nepredpokladateľné vzorce, ak k nim pridáme druhú dimenziu reprezentujúcu čas z nekomplexných pravidiel.

<sup>4</sup>Conwayova hra života - [https://en.wikipedia.org/wiki/Conway%27s\\_Game\\_of\\_Life](https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life)

Viacrozmerné CA využívajú tri alebo viac dimenzií a sú prevažne využívané na modelovanie fyzikálnych javov, ako je tvorba kryštalických štruktúr alebo difúzia.

### **1.2.2 Príklady CA zamerané na simuláciu rastlín**

„Simulation of root forms using cellular automata model“ - zameriava sa na korene rastlín a dopad premenných ako napr. živiny v pôde, voda a prekážky v raste. Jedná sa o prácu, ktorá je viac zameraná na matematický rast koreňov, avšak my sme rast koreňov simulovali genetikou.

„A Model Based on Cellular Automata for the Simulation of the Dynamics of Plant Populations“ - zameriava sa na 2D simuláciu heterogénnej populácie rastlín, zameraná na konkurenciu medzi druhmi, rýchlosť rastu a interakcie s prostredím.

## 2 Ciele práce

Cieľom tejto práce je vytvorenie aplikácie schopnej vytvárania a simulácie pseudo-realistických rastlín v trojrozmernej mriežke pomocou celulárnych automatov a implementovanej genetiky. Tieto rastliny by mali byť schopné reagovať na prostredie v reálnom čase a adaptujú svoje správanie, tvar a iné charakteristiky v závislosti od podmienok, v ktorých sa nachádzajú.

Veríme, že tieto metódy sú využiteľné ako na vytváranie dynamického a nerepetitívneho prostredia vo video-hrách, tak aj na viac vedecké účely ako pozorovanie správania rastlín vo virtuálne vytvorených podmienkach a ich ideálne charakteristiky za daných podmienok.

### **3 Materiál a metodika**

Simulátor bol vytvorený vo Visual studio code za pomoci jazykov C++ a GLSL. Ďalej sme využili knižnicu GLFW, ktorá pomáha s vývojom aplikácií vďaka tomu, že poskytuje jednoduché API na vývoj aplikácií. Ďalej sme využili Glad, softvér, ktorý generuje multi-jazykové loadery, matematickú knižnicu GLM a logovaciu knižnicu spdlog. Na vykreslenie trojrozmerného prostredia sme využili OpenGL.

## 4 Aplikácia

Na realizovanie algoritmu sme vyvinuli jednoduchú aplikáciu, ktorá má všetky potrebné prostriedky na demonštráciu všetkých aspektov tohto algoritmu. Aplikácia umožňuje vizualizáciu rastu rastlín na základe celúrných automatov a poskytuje možnosť meniť parametre simulácie na dosiahnutie ideálneho vzhľadu rastlín.

### 4.1 Svet

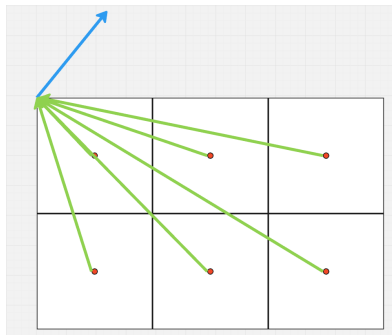
Svet aplikácie pozostáva z mriežky buniek o rozmeroch 32x32x32 voxelov. Každý voxel nadobúda určitý stav. Môže sa jednať o časti rastlín (stonka, list, koreň alebo ovocie), ale aj o časti terénu (vzduch, voda, pôda).

#### 4.1.1 Perlinov šum

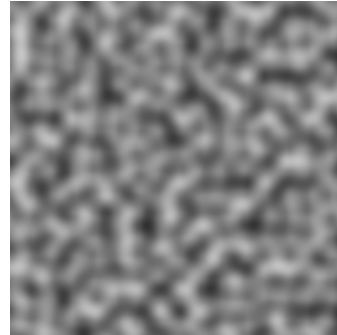
Terén na určenie svojej výšky, aby nebol len rovina, využíva funkciu zvanú Perlin noise. Perlin noise, je spôsob generovania hodnôt, ktoré sú náhodné, pričom body, ktoré sú si susedné, sa v hodnotách moc nemenia. Toto vytvára hodnoty, ktoré sa dajú skvelo využiť na generovanie realistického terénu. Čo sa týka algoritmu na jeho generovanie, v určitých miestach sa vytvoria vektory s náhodným smerom (modrý). Z bodov (červené), ku ktorým chceme pripísať hodnoty, sa vytvoria vektory smerujúce k týmto miestam (zelené). Z náhodného vektoru a vektoru smerujúceho z bodu sa vytvorí skalárny súčin a tým dosiahneme hodnotu v danom bode.

Avšak väčšinu času sa hodnoty nepočítajú len pre jeden náhodný vektor, ale viacero, najčastejšie 4 najbližšie a hodnoty ich skalárnych súčtov sa spriemerujú. Často sa využíva aj viacero oktáv, pričom každá októva generuje viacej náhodných vektorov, ktoré ovplyvňujú menej buniek, ako predchádzajúca októva, avšak vplýva menej na výslednú hodnotu. Ak vyjadríme hodnoty bodov farebnou škálou kde vyššie hodnoty sú viac biele a nižšie viac čierne dostaneme nasledujúci obrázok:

Naša aplikácia, vďaka jej obmedzenej veľkosti, využíva 4 vektory umiestnené na rohoch simulácie a iba 1 októvu.



(a) ...



(b) ...

(zdroj: [https://en.wikipedia.org/wiki/Perlin\\_noise](https://en.wikipedia.org/wiki/Perlin_noise))

Obr. 3: Perlinov šum

#### 4.1.2 Vytváranie terénu

Vytváranie sa začne vytvorením Perlin noise, podľa ktorého sa určí jeho výška. Pre každú dvojicu súradníc  $X$  a  $Z$  sa vytvorí hodnota Perlin noise. Táto hodnota bude ovplyvňovať os  $Y$ . Každý voxel, ktorého hodnota súradnice  $Y$  je nižšia alebo rovná hodnote noise v  $X, Z$  bude voxel terénu. Inak sa bude jednať o voxel vzduchu. Na vytvorenie voxelov vody sa nastaví hodnota, ktorá bude označovať hladinu vody a ak hocaký voxel vzduchu má hodnotu  $Y$  menšiu alebo rovnú ako táto hodnota, zmení sa na vodu. Táto hodnota nie je spojená s noisom, ide o jedno číslo univerzálne pre celú simuláciu. Zoberme si príklad: máme zmenšenú simuláciu a pre každú dvojicu  $X, Z$  pripadajú len 4 voxely  $[0, 1, 2, 3]$ . Tieto voxely majú rovnaké súradnice  $X, Z$  ale odlišné  $Y$ , ktorá korešponduje s ich číslom. Vytvoríme Perlin noise, ktorý bude mať pre danú dvojicu  $X, Z$  hodnotu 1. To znamená, že voxely s hodnotou  $Y$  0 a 1 sa premenia na voxely pôdy a voxely 2, 3 na voxely vzduchu. Ďalej hladina vody má hodnotu 2, takže voxel 2, ktorý bol pred tým vzduch sa premení na vodu.

#### 4.1.3 Živiny

Simulácia obsahuje rôzne živiny, vodu a vzduch, tieto veci rastlina potrebuje získavať aby rástla a prežila. Voda a vzduch majú zastúpenie v podobe ich voxelov. Pôda obsahuje vodu a živiny. Každá živina napomáha rôznym procesom rastliny, ako napríklad fotosyntézovať, čerpať vodu alebo získavať viac živín. Ich nadbytok má kladný vplyv na rastliny, avšak ich nedostatok má zase negatívny. Toto pridáva na komplexnosti, realizme a dovoľuje nám získať rôzne typy rastlín, podľa toho, ku ktorým živinám majú alebo nemajú

prístup. Napríklad ak rastlina nemá dostatok dusíka, ktorý je v simulátore dôležitý na fotosyntézu, rastlina to musí kompenzovať viacerými listami, inak zanikne. Živiny sú 3, dusík, draslík a fosfor, pričom bola snaha o to, aby napomáhali realistickým procesom, ktoré tieto prvky vyžadujú. Draslík je nápomocný pri absorpcii vody, takže ak má rastlina nadbytok draslíka, efektívnejšie získava vodu. Využitie dusíka je široké, ale v simulátore je jeho funkcia obmedzená len na zlepšenie fotosyntézy, keďže je hlavnou zložkou chlorofylu. Fosfor je rovnako ako dusík dôležitý v rôznych častiach, avšak v simulátore je zameraný na absorpciu živín.

## **4.2 Rastliny**

Rastliny sú v tomto simulátore objekty so spoločnou „triedou.“ Trieda je predloha premenných a špeciálnych funkcií zvaných metód, ktoré využívajú jednotlivé entity, taktiež známe ako objekty. Takáto šablóna nám umožňuje jednoducho a usporiadane vytvárať rastliny, pretože nemusíme písať samostatný kód pre každú rastlinu.

### **4.2.1 Premenné v rastlinách**

Rastliny majú viacero premenných. Ich hlavnou úlohou je opísať stav rastliny. Medzi ne patria údaje o skonsumovaných živinách za toto kolo, zoznam všetkých voxelov, v ktorých sa nachádzajú časti rastliny a ich gény. V premenných taktiež máme aj bonusy pre živiny. Hovoria nám o nadbytku alebo nedostatku živín. Čo tieto bonusy robia je rozbrané v kapitole „Živiny“.

### **4.2.2 Rastlinné voxel**

Fyzické zobrazenie rastlín sa skladá zo špecializovaných voxelov, ktoré v mriežke reprezentujú ich časti. Tieto voxel taktiež voláme aj bunky. Existujú 4: bunky koreňov, stonky, listov a ovocia. Každá skupina buniek plní vlastnú úlohu. Koreň má za úlohu rastline získať živiny a vodu, to dosahuje odoberaním týchto živín z okolitých voxelov. Koreň môže vzniknúť iba na miestach, kde je pôda. Úlohou listu je fotosyntéza, tú dosiahne tým, že skontroluje, či políčka nad ním sú vzduch. Predpokladáme, že na voxel vzduchu priamo svieti slnko. Môže nahradiť políčka kde je vzduch. Stonka uskladňuje živiny pre rastlinu. Rastie iba zvislo. Ovocie vytvára nové rastliny s rovnakým genetickým kódom ktorý trochu zmutuje. Na pár kôl nahrádza 1 list, ktorý sa po premenení ovocia na novú rastlinu

znova zmení na list.

### 4.2.3 Genetika

Na základe genetiky rastliny rozhodujú, aké časti majú rásť (listy, stonka. . . ) a v prípade listov a koreňov sú gény, ktoré hovoria, akým smerom majú rásť. Takže existujú 3 gény: čo má rásť, kde majú rásť korene a kde majú rásť listy. Gén „čo má rásť“ je vo všetkých prípadoch list so 4 číslami. Sú 4, pretože sú 4 typy buniek, ktoré môže rastlina rásť: korene, listy, stonku a ovocie. Čísla v týchto listoch označujú, ako moc chce daná rastlina danú časť rásť. Takže ak má rastlina v časti označujúcu stonku 2 a koreň 10, daná rastlina bude chcieť koreň rásť 5-krát viac, ako stonku. Ktorú časť bude rastlina naozaj rásť sa rozhoduje náhodne. Takže ak sa vrátíme k nášmu príkladu dokopy  $2+10=12$ , takže to je ako keby naša rastlina hádzala s 12-stenou kockou. Ak padne číslo, ktoré je väčšie ako 2, rastie koreň, inak stonka. Gény „kde majú rásť korene“ a „kde majú rásť listy“ sú rovnaké, len ako ich názov hovorí, jeden sa zaoberá koreňmi a druhý listami. Sú to listy s 26 číslami. Je ich 26 pretože bunka má v trojrozmernom priestore 26 susedov, ak rátame aj diagonály. Zvyšok funguje rovnako ako gén „čo má rásť“. Takže sú tam čísla, ktoré označujú pravdepodobnosť rastu do toho smeru. Ak sa tieto 2 gény zavolajú vyberie sa náhodná bunka (list alebo koreň, podľa toho, ktorý gén sa zavolá) a ak nemôže rásť do toho smeru, tak sa vyberie iná bunka, ak žiadna nemôže rásť, vyberie sa iný smer.

## 4.3 Algoritmus

Ako bolo spomínané, simulácie nebeží súvisle, ale v časových „krokoch“. Za každý krok sa vykonajú určité akcie, tzv. cyklus. Tieto akcie slúžia na beh simulácie a starajú sa o veci ako je rast rastlín, ich vzniknutie, smrť atď. Algoritmus za týmto môžeme rozdeliť do nasledujúcich častí, ktoré sú spísané chronologicky, čiže postupne, ako ich simulácia vykonáva.

### 4.3.1 Vznik rastlín

Je jediná časť algoritmu, ktorá nebeží v krokoch, ale iba raz, a to na začiatku simulácie. Po vytvorení terénu sa vyberie náhodné miesto na jeho povrchu, kde je pôda, a tam vznikne rastlina. Toto sa dá opakovať viackrát, na vytvorenie viacerých rastlín. Takéto rastliny sa



skladajú z 1 koreňa, 1 stonky a 1 listu. Ich genetika je náhodná, keďže ju nemôžu dediť z materínskych rastlín, ktoré neexistujú.

#### **4.3.2 Redistribúcia živín**

Je prvá akcia, ktorá sa vykonáva v cykle, každý krok simulácie. Živiny sú totiž ukladané v dvoch listoch, jeden permanentný, ktorý drží informácie o počte živín a vody zo začiatku simulácie a druhý dočasný, z ktorého rastliny čerpajú. V tomto bode sú odobrané všetky živiny, ktoré sa udržiavajú v rastlinách a list dočasných živín je znovunastavený na hodnoty rovné permanentnému listu. Toto umožňuje rastlinám znovu brať živiny v novom cykle.

#### **4.3.3 Zasadenie rastlín**

Ak má rastlina ovocie, ktoré existuje po určitú dobu, tak sa z neho stane nová rastlina. Rozdiel vo vzniku a zasadení je, že tentokrát genetický kód nie je vygenerovaný náhodne, ale ho rastlina zdedí zo svojej materinskej rastliny, pričom v každej časti genetiky jednu hodnotu náhodne zmení, teda „zmutuje“.

#### **4.3.4 Produkcia**

V tejto fázy rastlina získava živiny vďaka svojím bunkám, konkrétne listom a koreňom. Listy sa pozerú na bunky horizontálne nad nimi a podľa toho, koľko z nich sú vzduch, „produkujú“ svetlo. Korene zase pozerú bunky v ich susedstve a podľa toho generujú živiny a vodu. Koľko z okolitých buniek dokážu listy a korene dostať je určený samozrejme, počtom živín v bunkách terénu, z ktorého rastlina čerpá, ale rastlina nezoberie jednoducho všetky živiny v okolitom teréne. Maximálny počet živín, ktoré dokáže bunka získať je stanovený konštantou a jej bonusom zo živín. V stonkách sa uskladňujú živiny, čiže rastlina nemôže skonzumovať viac, ako jej to limit odvodený z počtu buniek stoniek dovoľuje. Na príklad: chceme fosfor, je koreň s 5 okolitými bunkami, pričom každá obsahuje 15 jednotiek fosforu. Konštantu produkcie máme 10, povedzme že máme bonus pre fosfor krát 1.2, čiže z každej dokážeme brať až 12 jednotiek. To máme do kopy 60 jednotiek, ktoré dokáže daný koreň získať, ale povedzme, že kapacita stoniek je len 50, takže náš koreň nakoniec vyprodukuje len 50 jednotiek fosforu.

#### **4.3.5 Smrt'**

Ak rastlina má nedostatok jednej zo živín, zomrie. Koľko živín rastlina potrebuje na prežitie je odvodené od jej veľkosti, čiže počtu jej voxelov a premennej určenej na začiatku simulácie zvanej „obtiažnosť“. Ako príklad dajme rastlinu s 10 bunkami, obtiažnosťou 5 a 45 jednotkami živiny, ktorú pozorujeme ako napr. fosfor. Vynásobíme počet buniek a obtiažnosť, takže  $10 \cdot 5 = 50$ . To znamená, že rastlina potrebuje najmenej 50 jednotiek zo všetkých živín, aby prežila. Keďže má len 45 jednotiek fosforu, zomrie na jeho nedostatok.

#### **4.3.6 Kalkulácia bonusov zo živín**

Podobne ako smrť, sa odvíja od veľkosti rastliny a konštanty. Pre každý bonus sa ráta zvlášť a keďže tento krok je až na konci cyklu v úplne prvom cykle simulácie majú všetky rastliny bonusy rovné 1.

## **5 Výsledky práce**

## **6 Diskusia**

## **7 Závěry práce**

## **8 Zhrnutie**

## **9 Zoznam použitej literatúry**