



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV AUTOMATIZACE A INFORMATIKY

INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

CELULÁRNÍ AUTOMATY – 1D A 2D SIMULÁTOR

CELLULAR AUTOMATONS - 1D AND 2D SIMULATOR

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Filip Sova

VEDOUCÍ PRÁCE

SUPERVISOR

prof. Ing. Radomil Matoušek, Ph.D.

BRNO 2024

Zadání bakalářské práce

Ústav:	Ústav automatizace a informatiky
Student:	Filip Sova
Studijní program:	Strojírenství
Studijní obor:	Aplikovaná informatika a řízení
Vedoucí práce:	prof. Ing. Radomil Matoušek, Ph.D.
Akademický rok:	2023/24

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Celulární automaty – 1D a 2D simulátor

Stručná charakteristika problematiky úkolu:

Celulární, či buněčný automat (CA, Cellular Automaton) je diskrétní dynamický systém, který je tvořen pravidelnou strukturou buněk definovaných v N-rozměrném prostoru. Každá buňka může nabývat jeden z K možných stavů.

Hodnoty stavů buněk se pro navazující časové kroky vypočtou synchronně na základě lokální přechodové funkce. Smyslem uvedené práce bude vytvořit implementaci a vizuálně pojatý simulátor CA s vhodným GUI rozhraním.

Cíle bakalářské práce:

- 1/ Nastudovat a stručně shrnout problematiku CA ve vztahu k realizaci práce a provést rešerši dostupných knihoven pro práci s CA.
- 2/ Realizovat vlastní implementaci CA (případně rozšířit stávající zvolenou knihovnu) v kontextu 1D, 2D automatu.
- 3/ Vhodně uživatelsky zpřístupnit definici rozměru automatu, topologii světa, okolí buňky a definici pravidel pro vlastní vizuální simulátor v GUI.
- 4/ Na několika vhodných příkladech prezentovat význam CA.

Seznam doporučené literatury:

WOLFRAM, S., 1994. Cellular automata and complexity. Wokingham: Addison-Wesley. ISBN 0-201-62664-0.

LOUIS, Pierre-Yves a Francesca R. NARDI, [2018]. Probabilistic cellular automata. New York: Springer. Emergence, complexity and computation. ISBN 978-3-319-65556-7.

CODD, E. F., 1968. Cellular automata. London: Academic Press. ACM monograph series.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2023/24

V Brně, dne

L. S.

Ing. Pavel Heriban, Ph.D.
ředitel ústavu

doc. Ing. Jiří Hlinka, Ph.D.
děkan fakulty

ABSTRAKT

Tato bakalářská práce se zabývá návrhem simulátoru celulárních automatů (CA). Implementován byl v jednorozměrném (1D) a dvourozměrném (2D) světě. Teoretická část se zabývá historií, základními principy, klasifikací a praktickým využitím CA, včetně přehledu knihoven v Pythonu. Praktická část zahrnuje popis uživatelského rozhraní pro definici a vizualizaci CA pomocí knihoven pygame, pygame_gui a numpy. Výsledkem je uživatelsky jednoduchý a flexibilní simulátor, který může být využit pro studijní účely celulárních automatů.

ABSTRACT

This bachelor thesis is focused on design of a cellular automata (CA) simulator. It was implemented in a one-dimensional (1D) and two-dimensional (2D) world. The theoretical part covers the history, basic principles, classification and practical use of CAs, including an overview of libraries in Python. The practical part includes a description of the user interface for defining and visualizing CA using the pygame, pygame_gui and numpy libraries. The result is a user-friendly and flexible simulator that can be used for the study of cellular automata.

KLÍČOVÁ SLOVA

Celulární automaty, diskrétní matematický model, simulátor, celulární programování

KEYWORDS

Cellular automata, discrete mathematical model, simulator, cellular programming



ÚSTAV AUTOMATIZACE
A INFORMATIKY



2024

BIBLIOGRAFICKÁ CITACE

SOVA, Filip. *Celulární automaty - 1D a 2D simulátor*. Brno, 2024. Dostupné také z: <https://www.vut.cz/studenti/zav-prace/detail/157953>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav automatizace a informatiky, Vedoucí práce: prof. Ing. Radomil Matoušek, Ph.D.

ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že tato bakalářská práce je mým původním dílem, vypracoval jsem ji samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury.

Jako autor uvedené práce dále prohlašuji, že v souvislosti s vytvořením této práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků.

V Brně dne 24. 5. 2024

.....

Filip Sova

PODĚKOVÁNÍ

V této bakalářské práci bych rád poděkoval vedoucímu své práce, prof. Ing. Radomilu Matouškovi, Ph.D., za jeho odborné rady a připomínky. Dále bych chtěl poděkovat své rodině a přátelům za jejich podporu během psaní této práce, mého studia i mimo něj.

OBSAH

1	ÚVOD	15
2	CELULÁRNÍ AUTOMATY	17
2.1	Historie celulárních automatů	17
2.2	1D celulární automaty	20
2.2.1	Elementární 1D celulární automaty	20
2.2.2	Totalistické celulární automaty	22
2.2.3	Vícestavové 1D celulární automaty	23
2.2.4	Využití 1D celulárních automatů	24
2.3	2D celulární automaty	25
2.3.1	Hra života	26
2.3.2	Využití 2D celulárních automatů	30
2.4	Klasifikace celulárních automatů	31
2.5	Okrajové podmínky	34
2.5.1	Periodické okrajové podmínky	34
2.5.2	Náhodné okrajové podmínky	34
2.5.3	Rozložené okrajové podmínky	34
2.5.4	Konstantní okrajové podmínky	35
2.5.5	Nulové okrajové podmínky	35
2.5.6	Shrnutí okrajových podmínek	36
3	KNIOVNY PRO CELULÁRNÍ AUTOMATY	37
3.1	CellPyLib	37
3.2	Golly	37
3.3	Numpy a Pygame	38
4	SIMULÁTOR CELULÁRNÍCH AUTOMATŮ	39
4.1	Úvodní menu	39
4.2	1D simulátor	40
4.3	2D simulátor s ortogonální mřížkou	42
4.4	2D simulátor s hexagonální mřížkou	45
4.5	Ukázka simulátoru	46
4.5.1	1D mřížka	46
4.5.2	2D ortogonální mřížka	47
4.6	2D hexagonální mřížka	48
5	ZÁVĚR	49
	SEZNAM POUŽITÉ LITERATURY	51
	SEZNAM PŘÍLOH	55

1 ÚVOD

Celulární automaty (CA) představují fascinující oblast teoretické informatiky a matematického modelování, která nachází uplatnění v široké škále vědeckých a technických disciplín. Tyto automaty tvořené pravidelně uspořádanými buňkami, které mění své stavy na základě předem definovaných pravidel a stavů svých sousedů, umožňují simulaci a studium komplexních dynamických systémů.

Cílem této bakalářské práce bylo prozkoumat základní problematiku celulárních automatů se zaměřením na jednorozměrné (1D) a dvourozměrné (2D) automaty a vytvořit simulátor, který umožní uživatelsky přívětivou definici a vizualizaci těchto systémů. Práce se skládá ze dvou hlavních částí: teoretického úvodu do celulárních automatů, představení dosavadních možností knihoven pro celulární automaty v jazyce Python a popisu vlastně naprogramovaného simulátoru.

V první části práce je představena historie celulárních automatů, základní koncepty spojené s 1D a 2D automaty a jejich využití vyobrazeno na vhodných příkladech, které prezentují možnou komplexnost těchto systémů. Dále je proveden přehled dostupných knihoven v jazyce Python, které umožňují práci s celulárními automaty.

Druhá část práce se zaměřuje na praktickou realizaci simulátoru. Je popsáno uživatelské rozhraní pro ovládání samotného simulátoru a jeho možnosti. Tento simulátor umožňuje uživatelům definovat rozměry automatu, topologii světa, okolí buňky, rychlost a pravidla pro simulaci. Cílem bylo vytvořit intuitivní a flexibilní nástroj, který usnadní studium a experimentování s celulárními automaty.

V této práci je kladen důraz na flexibilitu a uživatelsky přívětivé ovládání simulátoru, který uživatelům umožňuje snadno definovat a vizualizovat celulární automaty. Tato práce poskytuje přehled o celulárních automatech a představuje vlastní řešení simulátoru jako nástroje pro práci a studium s celulárními automaty.

2 CELULÁRNÍ AUTOMATY

Celulární automat je diskrétní matematický model využívaný k simulování komplexních systémů. Skládá se z mřížky buněk umístěných v n -rozměrném prostoru. Mřížka může mít různé tvary, například čtvercová nebo šestiúhelníková. Nejčastěji se používá dvourozměrná mřížka s čtvercovými buňkami. Každá buňka může nabývat několika stavů. Většinou jde pouze o dva stavy. 0 – mrtvá buňka, 1 – živá buňka. Stav buněk se aktualizuje v pravidelných časových intervalech podle pevného pravidla a je závislý na předchozím stavu buňky a stavu sousedních buněk. Celulární automaty jsou využívány ke zkoumání různorodých systémů, například biologických nebo fyzikálních procesů, díky schopnosti generovat složité vzory chování z jednoduchých pravidel [1].

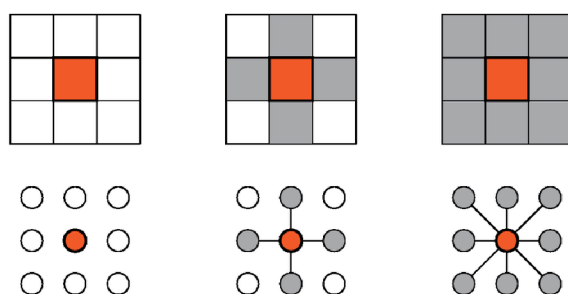
2.1 Historie celulárních automatů

Informace pro následující kapitolu zabývající se historií celulárních automatů byly částečně čerpány z knihy „Introduction to Computational Social Science“ [2].

Základy v oblasti celulárních automatů položili ve 40. letech 20. století matematici John von Neumann [1903 - 1957] a Stanislaw Ulam [1909 - 1984]. Jejich nápady v oblasti teorie automatů byly poprvé veřejně prezentovány roku 1948 a publikovány roku 1951 pod názvem „The General and Logical Theory of Automata“.

Rozšíření aplikace celulárních automatů do sociálních věd zahájil v roce 1949 sociolog James M. Sakoda ve své doktorské disertační práci. Ta vyšla později, roku 1971, v „Journal of Mathematical Sociology“, kde tento model pojmenoval „checkboard model“ (šachovnicový model).

V 60. letech představil počítačový vědec Edward Forrest Moore kocept prostředí, které pracovalo s 8 sousedy pro každou buňku. To představuje alternativu k von Neumannovu prostředí se 4 sousedy (viz obr. 1).

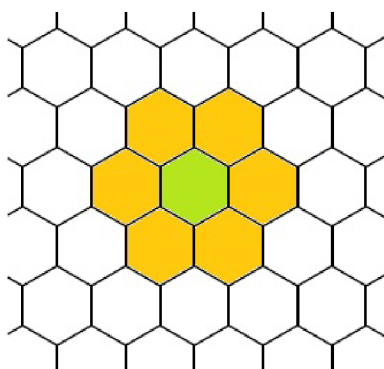


Obr. 1: Prostředí bez sousedů, se 4 sousedy, s 8 sousedy, převzato z [3].

V roce 1970 John Horton Conway vytvořil dnes populární model „Conway’s Game of Life“ (Hra života), která byla následně popularizován Martinem Gardnerem.

V 70. a 80. letech 20. století psycholog Bibb Latané formuloval teorii sociálního dopadu, což byl milník modelování za pomoci celulárních automatů v sociálním odvětví.

Roku 1977 se politolog Stuart A. Bremer stal průkopníkem modelování pomocí celulárních automatů v politologii díky simulaci války a míru v mezinárodním systému za pomoci šestiúhelníkového prostředí (viz obr. 2), systému „Machiavelli in Machina“, publikovaného Karlem W. Deutschem.



Obr. 2: Šestiúhelníkové prostředí, převzato z [4].

Okolo roku 1981 začal fyzik Stephen Wolfram pracovat na elementární teorii a modelování celulárních automatů. O dva roky později publikoval svůj první článek v „Reviews of Modern Physics“, později navrhl obecnou klasifikaci modelů celulárních automatů do čtyř hlavních kategorií.

Roku 1987 publikoval počítačový vědec James E. Doran svoji zásadní práci „Distributed Artificial Intelligence and the Modelling of Socio-Cultural Systems“, o modelování založeném na agentech.

V roce 1996 Rainer Hegselmann publikoval dvě vlivné práce: „Cellular Automata in the Social Sciences“ a „Understanding Social Dynamics“, které jsou považovány za jedny z nejlepších úvodů do simulace celulárních automatů ve společenských vědách.

Práce Stephena Wolframa vyvrcholily v roce 2002 v jeho publikaci „A New Kind of Science“, kde na 1280 stranách posunul pochopení složitých systémů a představil celulární automat jako mocný nástroj pro jejich modelování. V této práci popsal jeho vnímání celulárních automatů a určil tak nový směr výzkumu v oblasti celulárních automatů a jiných komplexních systémů.

Téhož roku byl spuštěn i první semestrální předmět na George Mason university, který se zabýval celulárními automaty a ABM (agent based modeling) v rámci programu Computational Social Science.

Roku 2006 bylo vydáno první číslo časopisu Journal of Cellular Automata, jehož cílem bylo šíření kvalitních prací, kde jsou celulární automaty teoreticky studovány nebo používány jako výpočetní modely v matematických, fyzikálních, chemických, biologických, sociálních a inženýrských systémech.

V roce 2010 vydal Andrew Adamatzky knihu „Game of Life Cellular Automata [5]“, která ukázala vědeckou zralost na první pohled jednoduchého Conwayova modelu Hry Života.

2.2 1D celulární automaty

Celulární automat v jedné dimenzi se skládá z lineárního řetězce buněk, které mohou nabývat jednoho z n možných stavů. Jejich stav závisí na předchozím stavu buňky a stavu jejich dvou sousedních buněk. Touto závislostí vzniká n^3 možných kombinací a n^{n^3} možných pravidel, kde n je počet možných stavů. Vývoj celulárního automatu lze popsat tabulkou určující stav, který bude mít daná buňka v následujícím kroku. [6], [7].

■ ■ ■	■ ■ □	■ □ ■	■ □ □	□ ■ ■	□ ■ □	□ □ ■	□ □ □
□	■	□	■	■	□	■	□

rule 90

Obr. 3: Tabulka pro pravidlo „90“, $n = 2$, převzato z [6].

■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■	□ □ □
■	□	□	■	■	■	■	□
<i>1</i>	<i>0</i>	<i>0</i>	<i>1</i>	<i>2</i>	<i>1</i>	<i>1</i>	<i>0</i>

Obr. 4: Tabulka pro pravidlo „777“, $n = 3$, totalistické pravidlo (viz kap. 2.2.2), převzato z [6].

2.2.1 Elementární 1D celulární automaty

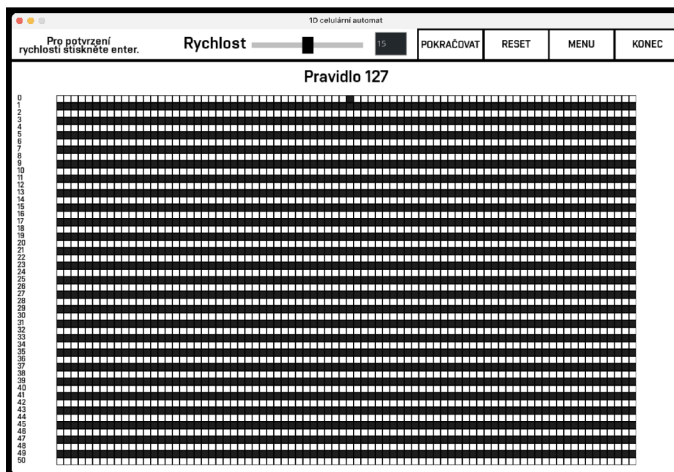
Elementární celulární automaty mají pouze dvě možné hodnoty pro každou buňku (0 nebo 1). Existuje pro ně tudíž 8 (2^3) možných kombinací a 256 (2^8) možných pravidel. Například tabulka vycházející z pravidla „30“ ($30 = 00\ 011\ 110_2$) je vyobrazena na obr. 5 [6].

■ ■ ■	■ ■ □	■ □ ■	■ □ □	□ ■ ■	□ ■ □	□ □ ■	□ □ □
□	□	□	■	■	■	■	□

rule 30

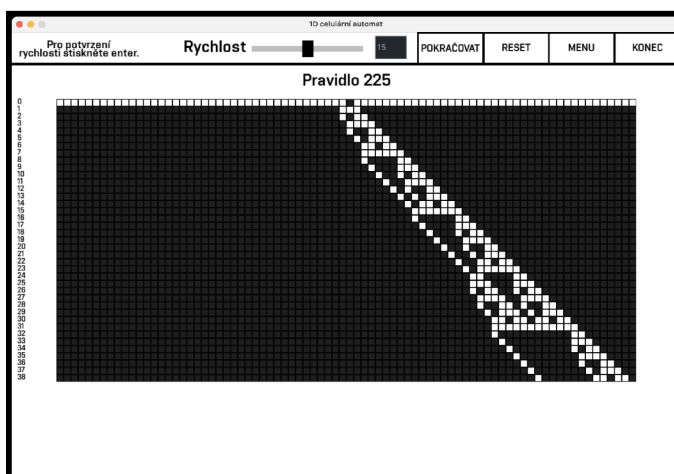
Obr. 5: Tabulka pro pravidlo „30“, elementární 1D automat (černá = 1, bílá = 0), převzato z [6].

Mezi 256 pravidly existuje několik opakujících se témat, které se dají řadit do tříd (viz kap. 2.4). Například pro pravidla 0 a 128 platí, že po prvním kroku všechny následující buňky nabývají stavu 0 a jsou tudíž všechny bílé, zatímco u pravidla „255“ jsou všechny černé. Pro pravidla „7 a 127“ platí, že se barvy každý řádek střídají (viz obr. 6) [6].



Obr. 6: Pravidlo „127“.

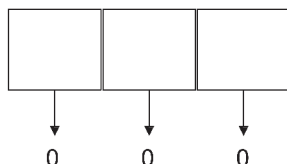
Většina pravidel ve výsledku generuje repetitivní vzory, avšak asi 14 % elementárních buněčných automatů dává složitější druhy vzorů. Z nich nejběžnější jsou vzory vnořené. Nejjednodušší a nejběžnější jsou například vzory pro pravidla „22 a 102“, existují ale i složitější formy, jako je pravidlo „225“, kde šířka obrazce se zvětšuje úměrně druhé odmocnině z počtu kroků (viz obr. 7) [6].



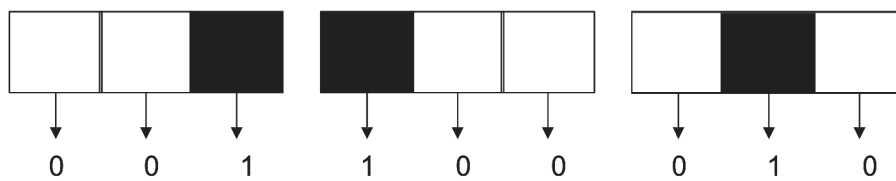
Obr. 7: Pravidlo „225“.

2.2.2 Totalistické celulární automaty

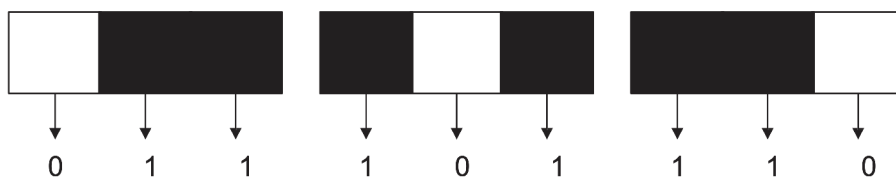
Totalistický celulární automat je druh celulárního automatu, kde záleží pouze na součtu hodnot okolních buněk. S touto myšlenkou přišel Stephen Wolfram v roce 1983. Narozdíl od elementárního celulárního automatu, kde existuje 8 možných kombinací, se u totalistického celulárního automatu pro dva stavy objevují pouze 4 pravidla. Ta vychází z možných součtů sousedních buněk 0, 1, 2, 3 (viz obr. 8 - 11). Počet možných pravidel proto klesá na pouhých 16 (2^4) možných. [8].



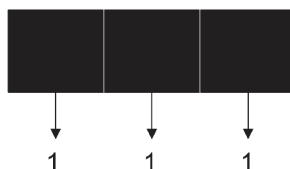
Obr. 8: Možné stavy buněk se součtem hodnot rovných 0, převzato z [8].



Obr. 9: Možné stavy buněk se součtem hodnot rovných 1, převzato z [8].

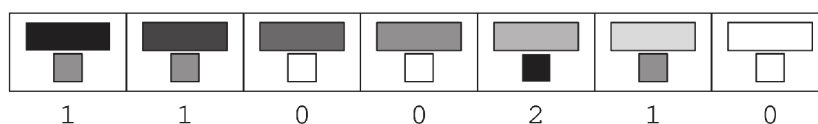


Obr. 10: Možné stavy buněk se součtem hodnot rovných 2, převzato z [8].



Obr. 11: Možné stavy buněk se součtem hodnot rovných 3, převzato z [8].

Stejně jako u elementárního celulárního automatu lze vývoj totalistického celulárního automatu popsat tabulkou určující stav buňky pro následující krok [8]. Obr. 12 ukazuje 3stavový automat, který by měl 27 možných kombinací. Totalistické pravidlo kombinace snižuje na 7 možných.



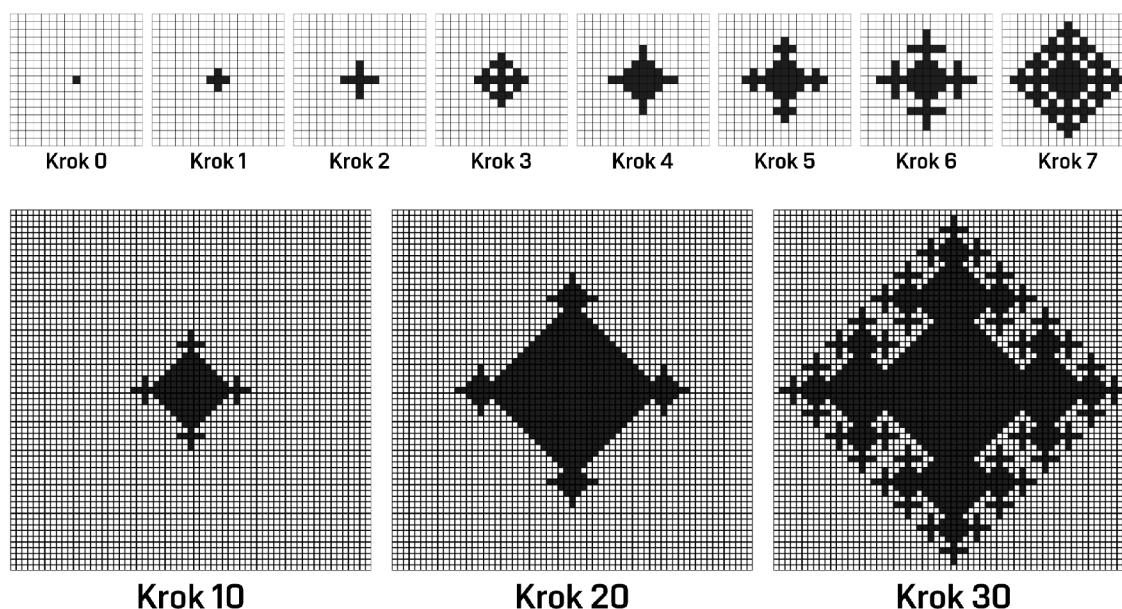
2.2.4 Využití 1D celulárních automatů

Pro naprostou většinu využití 1D celulárních automatů jsou dostatečné automaty využívající dva možné stavy. Tyto automaty generují dostatečně složité vzorce s menším nárokem na výkon výpočetního zařízení. 1D celulární automaty své uplatnění najdou například v generování pseudonáhodných čísel, které lze využít například v kryptografii [9], modelování samoorganizace ve statistické mechanice [10], nebo simulaci fyzikálních jevů [11].

Jedním z výše zmíněných využití je generace pseudonáhodných čísel. Touto aplikací se zabýval například i Stephen Wolfram. Jeho článek „Random Sequence Generation by Cellular Automata“ představuje 1D elementární celulární automaty právě jako nástroj pro generaci pseudonáhodných čísel (PRNG - pseudorandom number generator). V této studii se zabýval analýzou pravidel ideálních pro PRNG, hlavně pravidlem „30“, které nemá opakující se tendenci a bylo ve výsledku studie pravidlem s generací nejlepších náhodných čísel, tudíž čísel, která jsou nejméně předvídatelná v následujícím kroku [9]. Avšak článek „Generating High-Quality Random Numbers By Cellular Automata With PSO“ samostatné 1D elementární celulární automaty označuje jako nedostatečné ke generaci náhodných čísel podle sady statistických testů diehard. Jako nejvhodnější se vůči těmto testům jevílo pravidlo „105“, následovala pravidla „90 a 30“. Pro o něco klavitnější náhodná čísla tento článek označuje celulární automaty, které navíc využívají optimalizační algoritmus PSO (Particle Swarm Optimization) [12].

2.3 2D celulární automaty

Celulární automat využívající dvě dimenze se skládá z mřížky $n \cdot m$ buněk, které současně aktualizují své stavy v čase. Stav buněk stejně jako u 1D celulárních automatů závisí na předchozím stavu buňky a buněk sousedních. Vývoj 2D celulárního automatu může být popsán kódovým pravidlem, které představil Stephen Wolfram a Normanem Packardem [13], tento kód je velmi podobný tomu u 1D elementárních automatů. Příkladem je kód 942 ($942 = 1110101110_2$), kde buňka se stane živou v případě, kdy je právě 1 nebo 4 okolní buňky jsou živé (viz obr. 13). Princip dekódování není na první pohled jednoduše zřejmý jako u ostatních způsobů. Poslední číslice (první bit) určuje stav prostřední buňky v případě všech sousedních buněk mrtvých a prostřední buňky mrtvé, předposlední číslice určuje stav prostřední buňky, pokud je prostřední buňka živá a všechny sousední mrtvé. Každá předchozí číslice pak udává stav, který nastane pokud je postupně více sousedů živých.



Obr. 13: Vývoj pravidla „942“ pro elementární 2D celulární automat s Von Neumannovým sousedstvím, insporováno z [6].

Druhý způsob zápisu byl použit Mirkem Wójtowiczem pro software Mirek's Celebration (zkráceně MCell). Zde byl využit zápis ve tvaru „x/y“, kde hodnoty na pozici „x“ značí, kolik je potřebných buněk na udržení stavu 1 pro prostřední buňku. Hodnoty na pozici „y“ značí, kolik je potřebných sousedních buněk, aby prostřední buňka přešla ze stavu 0 na stav 1. Například pravidlo „23/3“ nám říká, že prostřední buňka se stane živou, pokud má přesně 2 nebo 3 sousedy a zůstává živou, pokud má přesně 3 sousedy [14].

Třetí způsob zápisu byl využit v softwaru pro simulaci celulárních automatů Golly. Tento zápis je velmi podobný druhému zmíněnému zápisu. Pravidlo je zapsáno jako „By/Sx“, kde S znamená „survive (přežít)“ a B „born (narodit se)“. Hodnoty „x“ a „y“ poté mají stejný smysl jako u zápisu pro MCell [5].

2.3.1 Hra života

Široký zájem veřejnosti o celulární automaty způsobil až John Horton Conway a jeho Hra života (anglicky Game of Life), když v roce 1970 publikoval svůj článek ve vědeckém časopise „Scientific American“ [15]. Specifičností těchto pravidel, které vedly k popularizaci bylo pravděpodobně objevení „oscilátorů“ (pravidelně opakujících se forem života) a „kluzáků“ (pohybujících se oscilátorů) [5].

Pravidla hry

Po dlouhém experimentování vybral John Conway svá pravidla tak, aby splnil tyto tři požadavky [15]:

1. Neměl by existovat žádný počáteční vzorec, pro který by existoval jednoduchý důkaz, že může populace neomezeně růst.
2. Měly by existovat počáteční vzorce, které zjevně rostou bez omezení.
3. Měly by existovat jednoduché počáteční vzorce, které rostou a mění se po určité dlouhou časovou dobu, než skončí třemi možnými způsoby: úplným vymizením (z důvodu přelidnění, nebo ředkosti populace), ustálením ve stabilní konfiguraci, která je poté neměnná, nebo přejitím do oscilační fáze, kde se opakuje nekonečný cyklus dvou nebo více period.

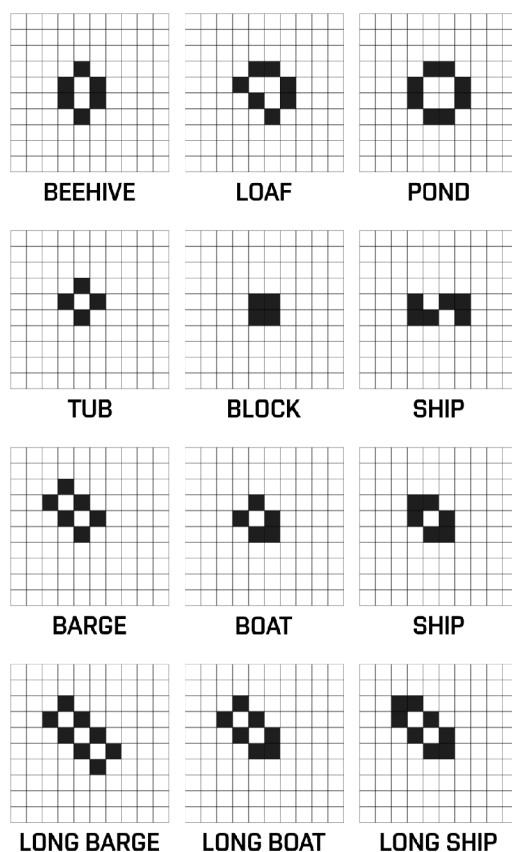
Souhrně by tyto požadavky měly docílit nepředvídatelnosti populace.

Hra života využívá Moorova sousedství, tudíž každá buňka závisí na stavu osmi sousedních buněk a stavu buňky samotné. Pravidla byla popsána následovně [15]:

1. Přežití. Každá živá buňka, která má dvě nebo tři sousední buňky živé, přežívá pro další generaci.
2. Úmrtí. Každá živá buňka, která má čtyři a více sousedů, umírá na přelidnění. Každá buňka, která má jednoho nebo žádného souseda, umírá na samotu.
3. Narození. Každá mrtvá buňka, která má přesně tři sousední buňky živé, stává se živou buňkou.

Stojící život (Still life)

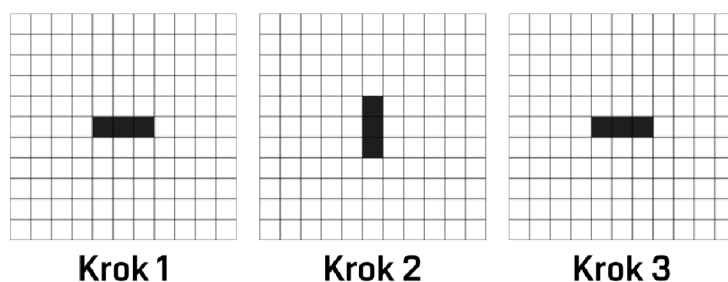
Nejkratším možným opakujícím se cyklem je stojící život, který má periodu cyklu 1, tím pádem se tento shluk buněk v čase nemění. Nejjednodušším příkladem je shluk živých buněk 2×2 , kde každá buňka má tři živé sousedy a je tak po celou dobu živá. Buňky mimo tento shluk mají nanejvýš dva živé sousedy, tudíž v nich nevznikne nový život. [5]. Možných forem stojících životů je však nekonečně mnoho [16].



Obr. 14: Základní formy stojícího života, inspirováno z [15].

Oscilátory (Oscillators)

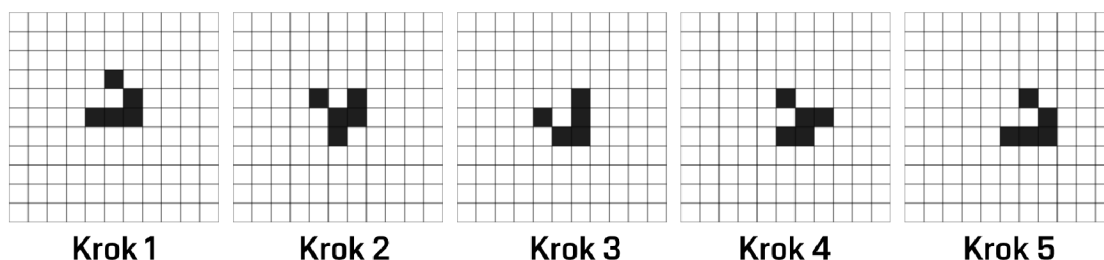
Oscilátory jsou nestabilní formy života, které jsou sami sobě předchůdcem. Za předchůdce se považuje vzor, ze kterého se vyvine jiný. U oscilátorů se tyto vzory cyklicky opakují. Oscilátory s nejkratší periodou (1) se označují za stojící život. Nejjednodušším oscilátorem s délkou periody 2 je tzv. „blikač (blinker)“ (viz obr. 15), který se pravidelně mění ze tří vertikálně sousedících živých buněk na tři horizontálně sousedící, tento oscilátor objevil John Conway v roce 1969 [17].



Obr. 15: Blikač.

Hvězdná loď (spaceship)

Hvězdná loď je konečný vzor, který se po určitém počtu generací vrátí do výchozího stavu, avšak na jiném místě. Nejmenším a nejjednodušším typem hvězdné lodě je tzv. „kluzák (glider)“, který představil Richard K. Guy v roce 1969, když zkoumal „Repentomino“, druh malého vzorce, který potřebuje velký počet generací k ustálení a v určitém okamžiku průběhu se jeho velikost stane mnohem větší, než původní vzor. V tomto vzoru si právě Richard K. Guy všiml, že se při 69. generaci vytvoří menší vzor, který se diagonálně posouvá, dokud nenarazí na jiný vzor. Kluzák posléze roku 1970 popsal John Conway, kterému jsou často milně přisuzovány zásluhy za jeho objevení [15], [18].

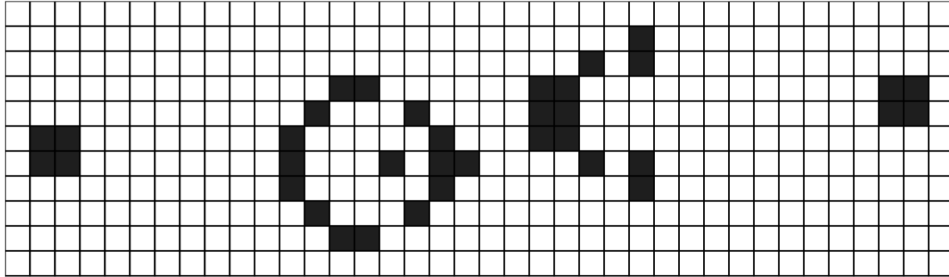


Obr. 16: Kluzák.

Dělo (gun)

John Conway původně předpokládal, že neexistuje vzor, který by mohl růst do nekonečna, tj. že pro každý počáteční vzor existuje konečný počet generací, po kterých se buďto populace stabilizuje, nebo zanikne. Při uvedení hry v časopise „Scientific American“ John Conway nabídl cenu 50\$ (což v roce 2024 odpovídá 410\$ [19]) za prokázání nebo vyvrácení této domněnky do konce roku 1970.

Tuto cenu vyhrál tým z „Massachusetts Institute of Technology (MIT)“ vedený Bill Gosperem, který vytvořil první dělo „Gosper glider gun“. To v 15. generaci vytvoří první kluzák a poté každou 30. generaci tvoří další kluzáky. Tento vzor byl po mnoho let nejmenším dělem a je dodnes nejznámějším ve Hře života. V roce 2015 byl objeven „Simkin glider gun“. Dělo, které vytvoří kluzák každých 120 generací má méně živých buněk, avšak tento vzor pokrývá větší potřebné pole pro svoji existenci [20].



Obr. 17: Gosper glider gun.

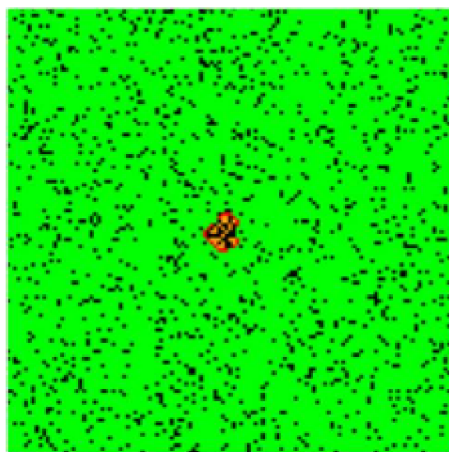
2.3.2 Využití 2D celulárních automatů

2D celulární automaty mají široké využití v různých oblastech. Jednou z aplikací je například medicína, kde celulární automaty mohou pomoci k modelování komplexních růstových vzorců buněk, například v kontextu nádorových onemocnění [21].

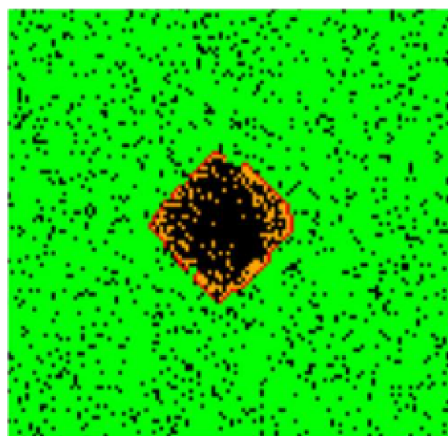
Další možností využití celulárních automatů je šíření požárů v lesních a zemědělských oblastech. Tento model v konkrétní studii [22] využívá šestistavového pravděpodobnostního celulárního automatu, což umožňuje dynamické zahrnutí různých faktorů, které ovlivňují šíření požáru, jako jsou typ a hustota vegetace, rychlost a směr větru a efekt „spottingu“, kdy jsou hořící materiály nesené větrem a mohou zapálit nové oblasti mimo hlavní frontu ohně. Tento model je zvláště užitečný pro řízení lesních požárů, poskytuje nástroje pro lepší plánování evakuací a zásahů hasičských sborů a také umožňuje lepší pochopení dynamiky šíření ohně v závislosti na místních podmínkách [22].

Jedním z využití je i modelování dynamiky hojnosti sinic v jezerech, kde se využívá celulární automat s proměnlivými pravidly vznikajícími na základě implementovaného genetického algoritmu [23].

V neposlední řadě lze využít celulárních automatů pro modelování šíření epidemie, kdy model dokáže zohledňovat demografické parametry jako jsou pohlaví, věková struktura, pohyb populace, inkubační a léčebná období, imunita a další, což umožňuje analýzu dopadů různých kontrolních strategií a identifikaci kritických faktorů pro řízení infekce. Tento model byl využit například pro analýzu šíření onemocnění COVID-19 [24].



(a) Počáteční konfigurace



(b) Rozšíření ohně

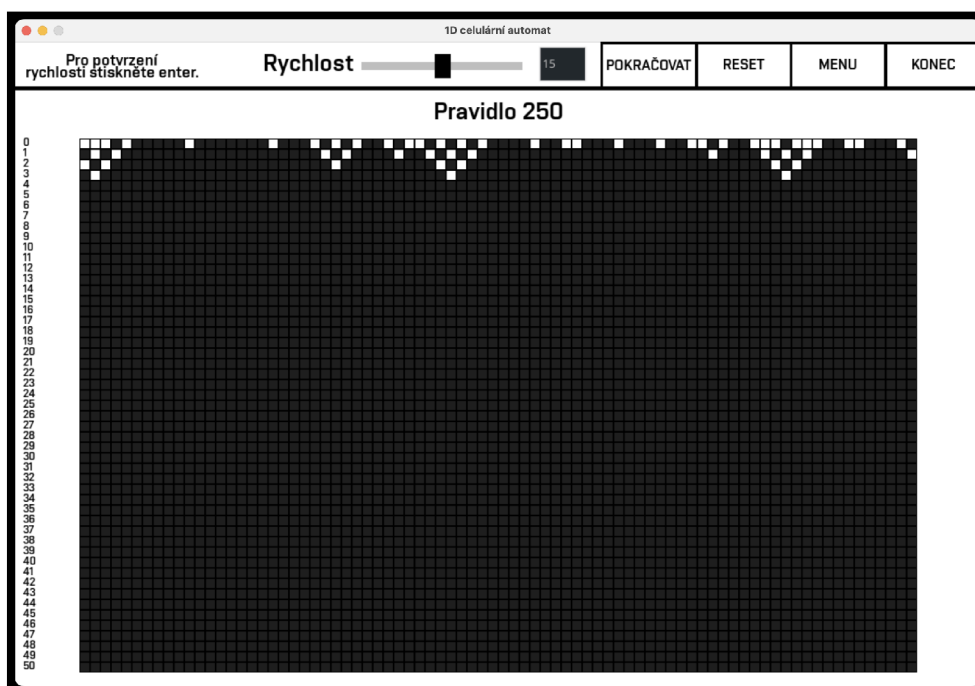
Obr. 18: Model šíření požáru v lesních oblastech pomocí celulárního automatu při 90% hustotě vegetace, převzato z [22].

2.4 Klasifikace celulárních automatů

Stephen Wolfram při pozorování chování celulárních automatů zjistil, že při náhodné inicializaci počátečního stavu se může zdát, že se každé pravidlo chová jinak. Avšak při hlubším pozorování se dá zjistit, že i když je každý vzorec v detailech jiný, počet významně odlišných vzorů je naopak velmi omezený. Podle Wolframa lze vznikající vzory téměř vždy poměrně snadno přiřadit do jedné z pouhých čtyř základních tříd. Tyto třídy vzestupně značí komplexitu dané třídy a každá z nich má vlastní charakteristické rysy. [6].

1. třída

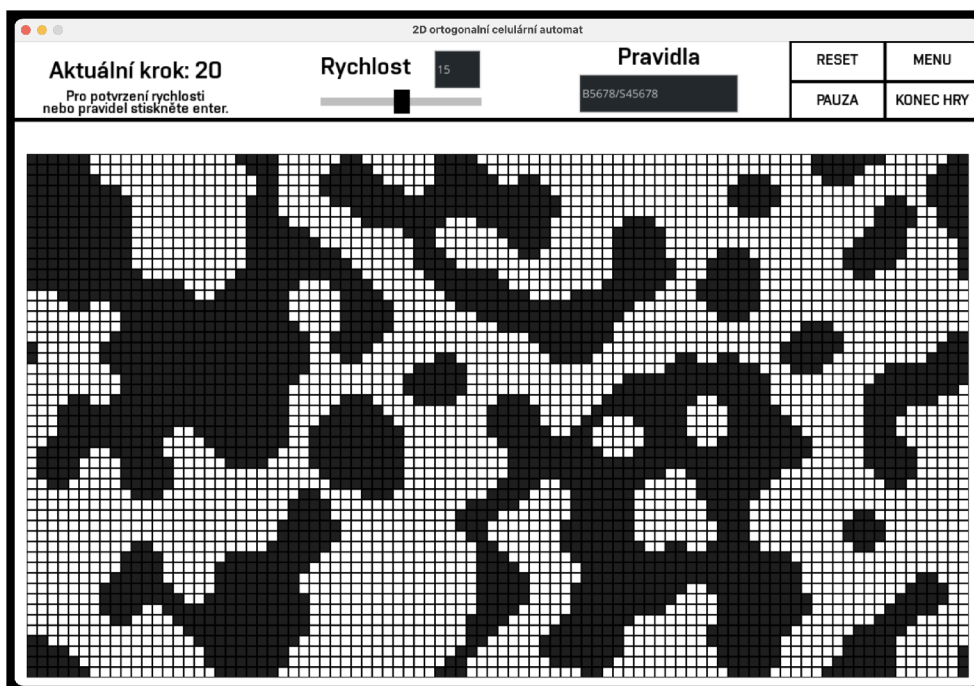
Tato třída je nejjednodušší a nejvíce předvídatelná. Do první třídy spadají automaty, které se téměř ze všech možných počátečních podmínek ustálí v jednotný konečný stav. Příkladem může být pravidlo „0“, kde se všechny buňky po prvním kroku změň na 0 a zůstanou tak po celou dobu simulace. [6].



Obr. 19: Příklad 1. třídy, pravidlo „250“.

2. třída

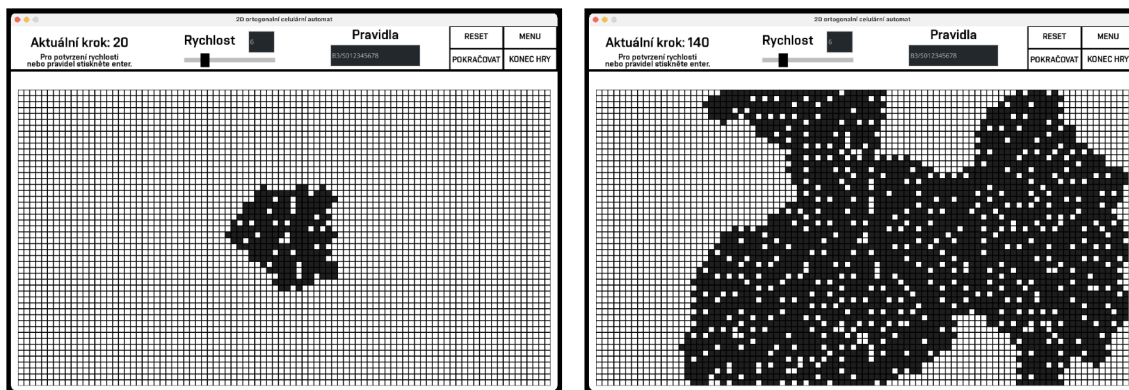
Ve druhé třídě již existuje mnoho různých konečných stavů. Všechny se skládají z určité množiny jednoduchých struktur, které se buďto ustálí, nebo se periodicky opakují [6]. Jedním z nejznámějších pravidel této třídy z 2D celulárních automatů je pravidlo „B5678/S45678“ [25].



Obr. 20: Příklad 2. třídy, pravidlo „B5678/S45678“.

3. třída

Ve třetí třídě je chování komplikovanější a v mnoha ohledech se zdá být náhodné, i přes to trojúhelníky a další malé struktury jsou vždy na nějaké úrovni viditelné [6]. Například pravidlo „Life without death (B3/S012345678)“, které roste do nekonečna [25].



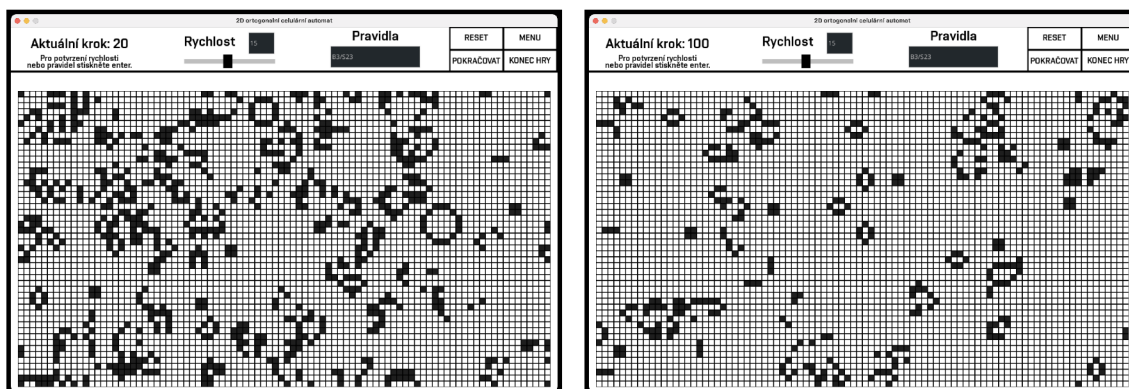
(a) Krok 20

(b) Krok 140

Obr. 21: Příklad 3. třídy, pravidlo „Life without death (B3/S012345678)“.

4. třída

Do poslední, čtvrté, třídy patří automaty, kde většina vzorů směřuje k směsi řádu a náhodnosti. V mikroskopickém měřítku zde spadá například také „Conwayova Hra života (B3/S23)“ [25].



(a) Krok 20

(b) Krok 100

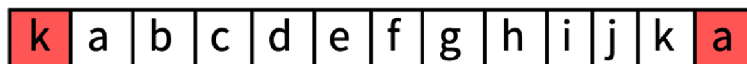
Obr. 22: Příklad 4. třídy, pravidlo „Conwayova Hra života (B3/S23)“.

2.5 Okrajové podmínky

Okrajové podmínky definují způsob chování buněk na okraji mřížky. Ideální celulární automat má mřížku nekonečně velkou, tudíž se okrajové podmínky neuvažují. V praxi z důvodu výpočetního výkonu a paměťové náročnosti se však využívají mřížky konečné. Tudíž je nutné uvažovat s okrajovými podmínkami, které právě uvažování konečné mřížky umožňují [26].

2.5.1 Periodické okrajové podmínky

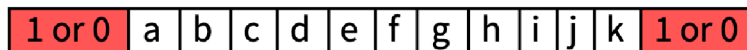
Hodnota buňky za okrajem mřížky je určena jako hodnota buňky nacházející se na protilehlém okraji mřížky. Tento způsob uvažování umožňuje simulovat mřížku jako toroidní svět, kde konce mřížky navazují na konec protilehlý [26].



Obr. 23: Periodické okrajové podmínky, převzato z [26].

2.5.2 Náhodné okrajové podmínky

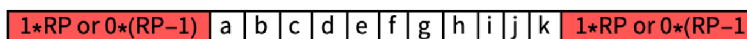
Hodnota buňky za okrajem mřížky je určena náhodně. Náhodná hodnota může být nastavena na začátku simulace, nebo se v průběhu náhodně měnit [26].



Obr. 24: Náhodné okrajové podmínky, převzato z [26].

2.5.3 Rozložené okrajové podmínky

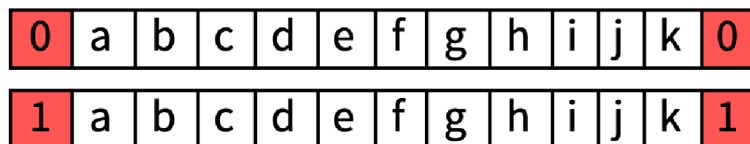
Rozložené okrajové podmínky jsou podobné těm náhodným, avšak hodnota buňky za okrajem mřížky je rovnoměrně náhodně rozložena [26].



Obr. 25: Rozložené okrajové podmínky, převzato z [26].

2.5.4 Konstantní okrajové podmínky

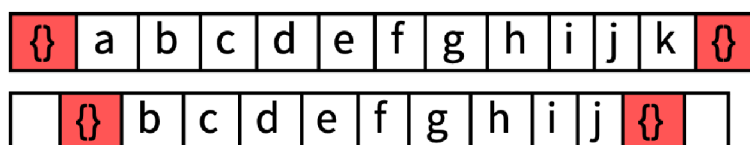
Hodnota buňky za okrajem mřížky je konstantní, od začátku simulace se nastaví buďto na 1, nebo na 0 a je celou dobu simulace neměnná [26].



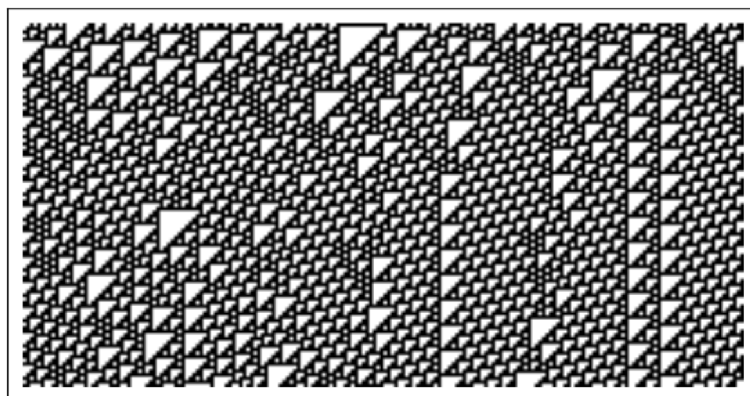
Obr. 26: Konstantní okrajové podmínky, převzato z [26].

2.5.5 Nulové okrajové podmínky

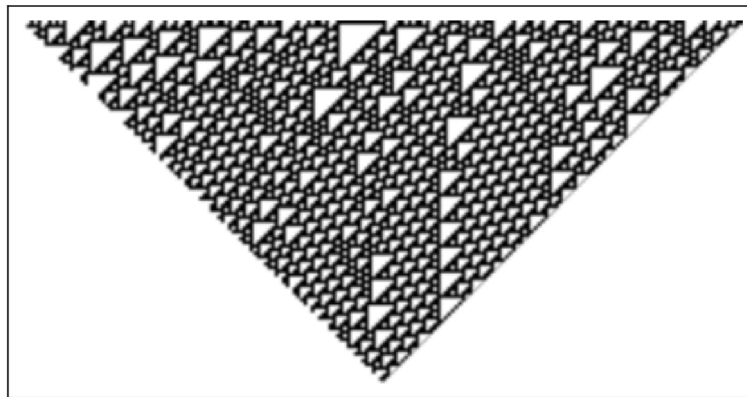
Hodnota buňky za okrajem mřížky není žádná, buňka na okraji tedy do dalšího kroku ztrácí informaci a přestává existovat. Automat tedy postupně ztrácí počet buněk, dokud nezůstane pouze jedna [26].



Obr. 27: Nulové okrajové podmínky, dvě generace pro ukázkou ztráty informace, převzato z [26].



Obr. 28: 1D CA, pravidlo „110“, periodické okrajové podmínky, převzato z [26].



Obr. 29: 1D CA, pravidlo „110“, nulové okrajové podmínky, převzato z [26].

2.5.6 Shrnutí okrajových podmínek

Ze závěru článku „The Effects of Boundary Conditions on Cellular Automata [26]“ vyplývá, že periodické, rozložené a konstantní okrajové podmínky jsou nejstabilnější a zvyšují předvídatelnost systému zejména pro systémy 2. třídy. Nulové okrajové podmínky naopak zajišťují nejkompexnější chování systému. Náhodné okrajové podmínky rovněž zajišťují komplexitu systému, avšak ne takovou, jakou zajišťují nulové okrajové podmínky.

3 KNIHOVNY PRO CELULÁRNÍ AUTOMATY

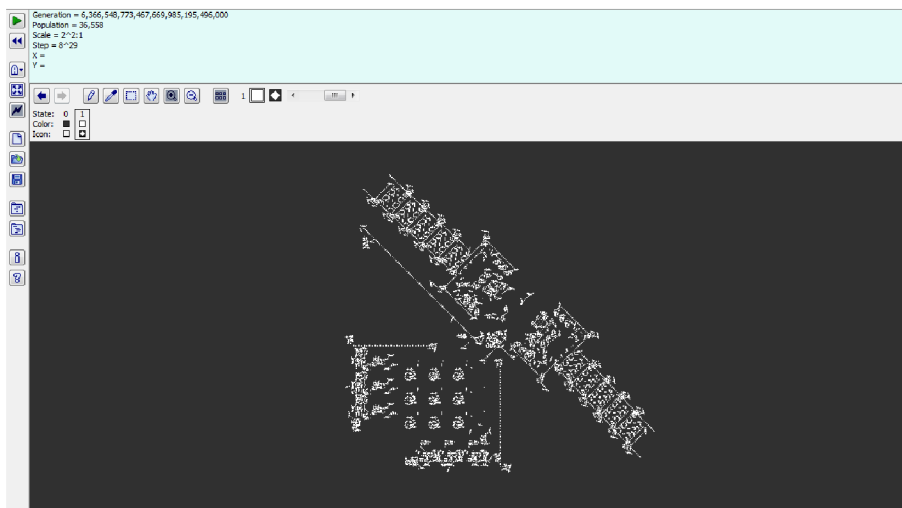
V programování celulárních automatů v Pythonu se setkáváme s řadou knihoven, které umožňují efektivně implementovat jak jednorozměrné (1D), tak dvourozměrné (2D) celulární automaty. Tyto knihovny poskytují základní i pokročilé nástroje pro definici pravidel, nastavení počátečních podmínek a vizualizaci výsledků, což je nezbytné pro správné fungování a analýzu těchto systémů.

3.1 CellPyLib

CellPyLib je jednou z knihoven, která je široce uznávána pro svou flexibilitu a uživatelsky přívětivé rozhraní. Tato knihovna je ideální jednoduchým přístupem například pro vzdělávací účely, neboť umožňuje snadno definovat a simulovat různé typy celulárních automatů, včetně komplexních a nestandardních pravidel. CellPyLib podporuje různé režimy hranic, což umožňuje experimentování s otevřenými nebo uzavřenými systémy a poskytuje vynikající nástroje pro vizualizaci, které jsou klíčové pro interpretaci dynamiky a chování modelů. Po napsání kódu se zobrazí simulovaný vývoj buněk v čase, což umožňuje uživatelům sledovat interakce mezi jednotlivými buňkami a zkoumat vliv různých pravidel a podmínek na celkový vývoj systému do kterého avšak nelze zasahovat. [27]

3.2 Golly

Golly je aplikace a knihovna, která překračuje běžné hranice Pythonu a nabízí výkonné rozhraní pro design, analýzu a simulaci celulárních automatů na velmi rozsáhlé mřížce. Golly je známý svou schopností zvládnout opravdu velké simulace s miliony buněk ve více dimenzích, což je činí vhodným nástrojem pro pokročilé uživatele a výzkumné projekty, které vyžadují vysokou úroveň detailu a výpočetní sílu. Kromě podpory v Pythonu, Golly umožňuje uživatelům manipulovat s automaty prostřednictvím grafického uživatelského rozhraní a obsahuje rozsáhlou knihovnu předdefinovaných pravidel a konfigurací [28].



Obr. 30: Golly, převzato z [29].

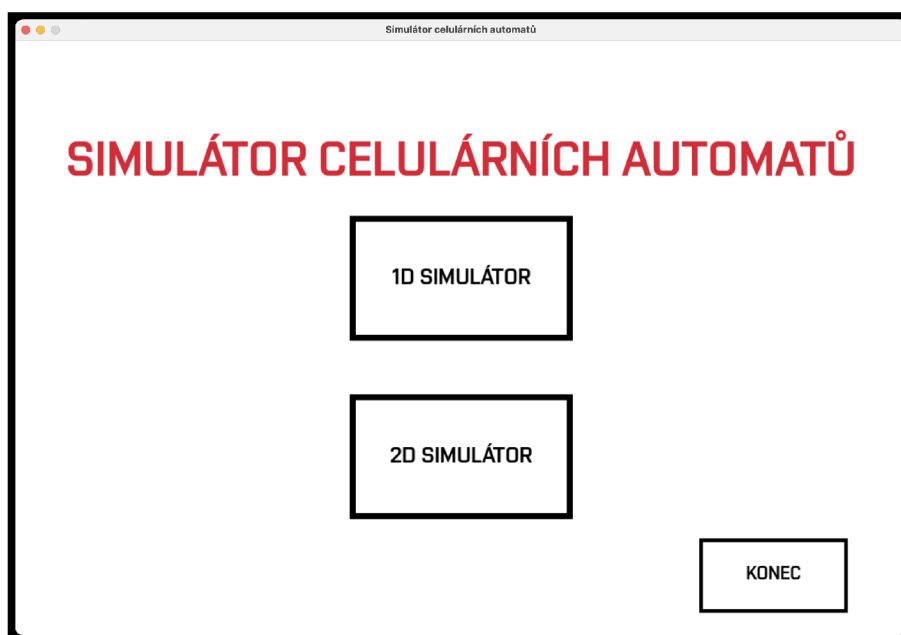
3.3 Numpy a Pygame

Kromě specializovaných knihoven, jako jsou CellPyLib a Golly, existuje také možnost využití kombinace obecnějších knihoven, jako jsou Numpy a Pygame, pro programování celulárních automatů. Numpy poskytuje výkonné nástroje pro numerické operace a efektivní manipulaci s velkými datovými strukturami, což je zásadní pro zpracování stavů buněk v celulárních automatech. Pygame pak nabízí rozsáhlé možnosti pro tvorbu interaktivních uživatelských rozhraní a dynamickou vizualizaci procesů, což umožňuje real-time zobrazení vývoje stavů celulárních automatů. Kombinace těchto dvou knihoven umožňuje vývojářům tvořit vlastní řešení simulátoru celulárních automatů, které je plně přizpůsobitelné z hlediska uživatelského rozhraní, pravidel a vizualizace.

4 SIMULÁTOR CELULÁRNÍCH AUTOMATŮ

Pro návrh simulátoru pro celulární automaty byl využit programovací jazyk Python. Pro zpracování grafického prostředí byla použita knihovna Pygame, kterou doplňovala knihovna `pygame_gui` pro možnost textových polí a výběrových tabulek. Zpracování buněk obstarává knihovna NumPy, která umožňuje efektivní práci s maticemi. Zpracování pravidel probíhá pomocí smyčky `for`, kde se pro každou buňku zjistí její okolí a podle pravidel se buňka změní na živou nebo mrtvou. Tlačítka, která umožňují uživateli pohyb v prostředí simulátoru, jsou vytvořena jako vlastní třída, která zpracovává informace jako pohyb myši přes tlačítko, nebo jeho stisknutí. Vykreslení tlačítka je následně zpracováno pomocí knihovny Pygame.

4.1 Úvodní menu



Obr. 31: Úvodní menu simulátoru, pozn. barvy upraveny vzhledem k tisku.

Při spuštění simulátoru se zobrazí úvodní menu, kde je možné zvolit buďto simulátor s 1D mřížkou, nebo 2D mřížkou. V celém prostředí je možné ukončit simulátor tlačítkem.

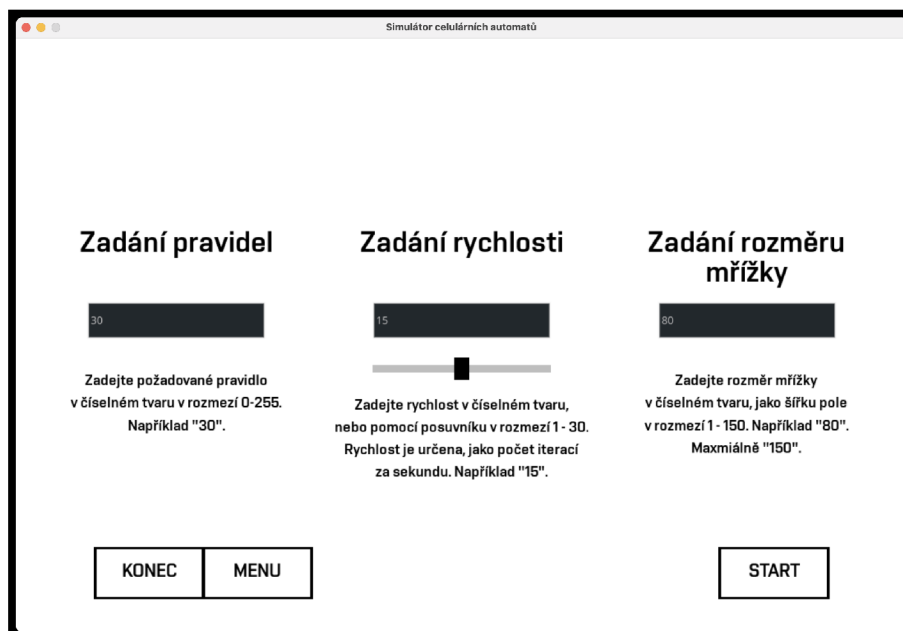
Po proklukutí 1D simulátoru dostane uživatel na výběr výchozího stavu, kde má možnosti zvolit simulátor vycházející z náhodného uspořádání živých buněk v nulté generaci, jedné živé buňky nacházející se uprostřed pole nebo prázdného pole pro zadání vlastního počátečního stavu.

Při výběru 2D simulátoru je na výběr z náhodného uspořádání živých buněk v nulté generaci, přednastavených vzorů, nebo prázdného pole pro zadání vlastního počátečního stavu.

Z přednastavených vzorů je na výběr kluzákové dělo (viz kap. 2.3.1), R-pentomino (viz kap. 2.3.1) nebo jedna živá buňka umístěna uprostřed pole, což umožňuje testování chování různých pravidel z jednoho života.

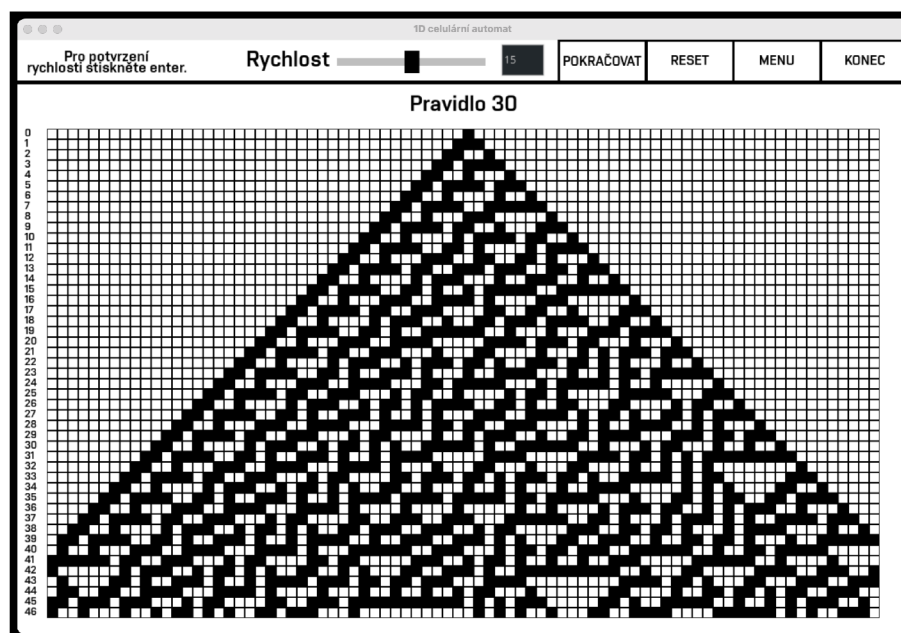
4.2 1D simulátor

Po výběru výchozího stavu 1D simulátoru se zobrazí možnost uživatelského zadání pravidel, rychlosti a rozměr mřížky. Po stisknutí tlačítka „START“, nebo stisknutí klávesy „Enter“ se spustí simulace.



Obr. 32: Definice prostředí 1D celulárního automatu.

Pravidla jsou zadána v číselném formátu a poté převedena do binárního kódu (viz kap. 2.2.1). Rychlost je rozvněž určena číselně a udává počet odsimulovaných generací za sekundu, kdy je možnost zadat rychlost od 1 do 30, pokud uživatel zadá 0, spustí se simulátor v pozastaveném stavu. V případě zadání hodnoty vyšší než 30 se rychlost automaticky nastaví na 30. Poslední atribut, rozměr mřížky, je určen jako počet buněk automatu na šířku. Maximální šířka mřížky je 150 buněk. Vyobrazená maximální výška mřížky je úměrná počtu buněk na šířku a výšce okna simulátoru. V případě zadání nesprávných hodnot se zobrazí chybová hláška a simulace se nepustí.



Obr. 33: 1D simulační prostředí, pravidlo „30“.

Po zadání a potvrzení všech atributů se spustí simulace 1D celulárního automatu, který lze pozastavit stisknutím klávesy „Mezerník“, nebo stisknutím tlačítka „POZASTAVIT“, které se po stisknutí změní na „POKRAČOVAT“, tímto tlačítkem lze simulační prostředí opět spustit. Tlačítkem „RESET“ se simulační prostředí pozastaví a uvede do výchozího stavu. Rychlost simulačního prostředí lze nastavit buď pomocí posuvníku, nebo zadáním hodnoty do textového pole. Po stisknutí tlačítka „Enter“ se potvrdí změněná rychlost.

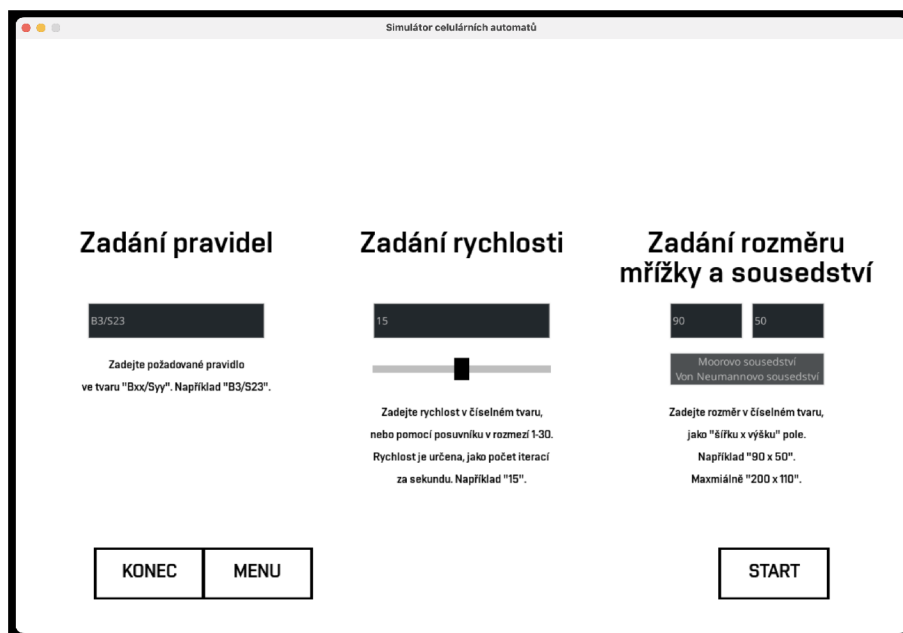
Na levé straně simulačního prostředí je očíslován krok simulačního prostředí, nad simulačním prostředím je aktuální zvolené pravidlo. V případě nesprávně zadané rychlosti (například zadání písmene) se zobrazí chybová hláška místo aktuálního pravidla a neproběhne žádná změna.

Jakmile se simulační prostředí dostane na konec mřížky, simulační prostředí se pozastaví. Při opětovném spuštění simulačního prostředí se řádky simulačního prostředí začnou posouvat a vrchní generace se začne přepisovat.

Simulační prostředí využívá periodické okrajové podmínky, tudíž se od určitého kroku začnou generace lišit od generací, které by vznikaly na nekonečné mřížce.

4.3 2D simulátor s ortogonální mřížkou

Po výběru náhodného stavu 2D ortogonálního (pravoúhlého) simulátoru se zobrazí obdobná možnost výběru pravidel jako u 1D simulátoru. Uživatel zadá pravidla, rychlost, rozměr mřížky a vybere typ sousedství. Po stisknutí tlačítka „START“, nebo stisknutí klávesy „Enter“ se spustí simulace.



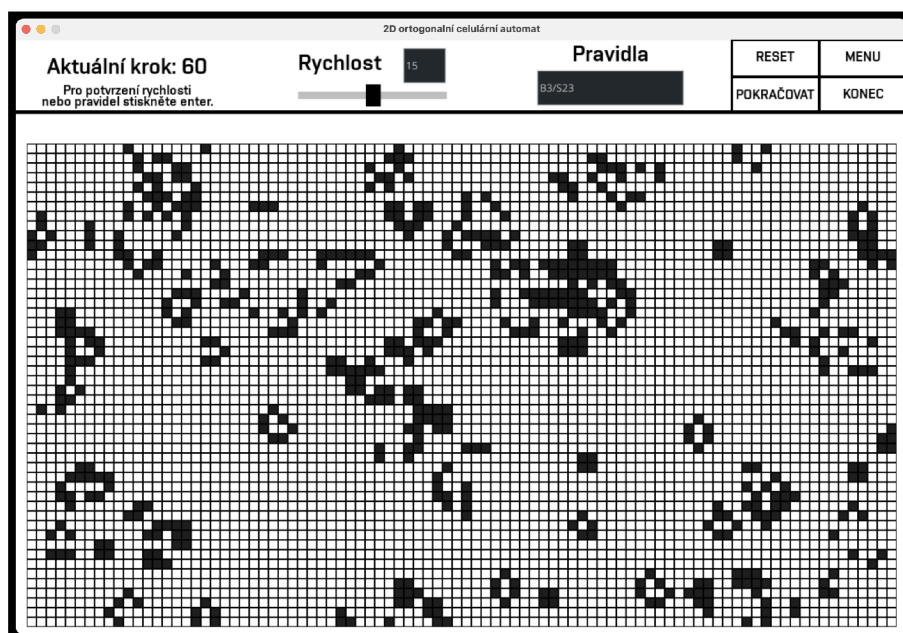
Obr. 34: Definice prostředí 2D ortogonálního celulárního automatu.

Pravidla jsou zadána ve tvaru „Bxx/Syy“ (viz kap. 2.3), rychlost je opět zadána číselně v rozmezí od 1 do 30, hodnoty neodpovídající tomuto rozmezí se nastaví stejně jako v případě 1D simulátoru (viz kap. 4.2). Rozměr mřížky je určen jako počet buněk na šířku a výšku mřížky, velikost buňky je dpočítána z rozměru mřížky a velikosti okna simulátoru. Maximální rozměr mřížky je 150 buněk na šířku a 100 na výšku. Typ sousedství je možné vybrat z možností „Mooreovo“ (čtyřokolí) nebo „Von Neumannovo“ (osmiokolí) (viz kap. 2.1). V případě zadání nesprávných hodnot, nebo nezvolení typu sousedství se zobrazí chybová hláška a simulace se nepustí.



Obr. 35: Definice přednastavených stavů 2D ortogonálního simulátoru.

Při výběru z přednastavených vrozů „R-pentomino“ a „glider gun“ chybí možnost nastavení pravidel a sousedství. Pravidla jsou zde nastavena fixně, konkrétně „B3/S23“ a sousedství je zde Mooreovo, což umožňuje správné fungování těchto vzorů. Uživatel může nastavit základní rychlost simulace a rozměr mřížky. Je zde doporučený minimální rozměr mřížky pro uživatele na základě zvoleného vzoru, aby byla zajištěna správná funkčnost simulace. Tyto dva vzory využívají konstantní okrajové podmínky, konkrétní nastavená hodnota je 0. Toto nastavení zabraňuje kluzákům, které by se v případě periodických okrajových podmínek dostaly na nepříslušné místa mřížky a tím by se narušila struktura přednastaveného vzoru.



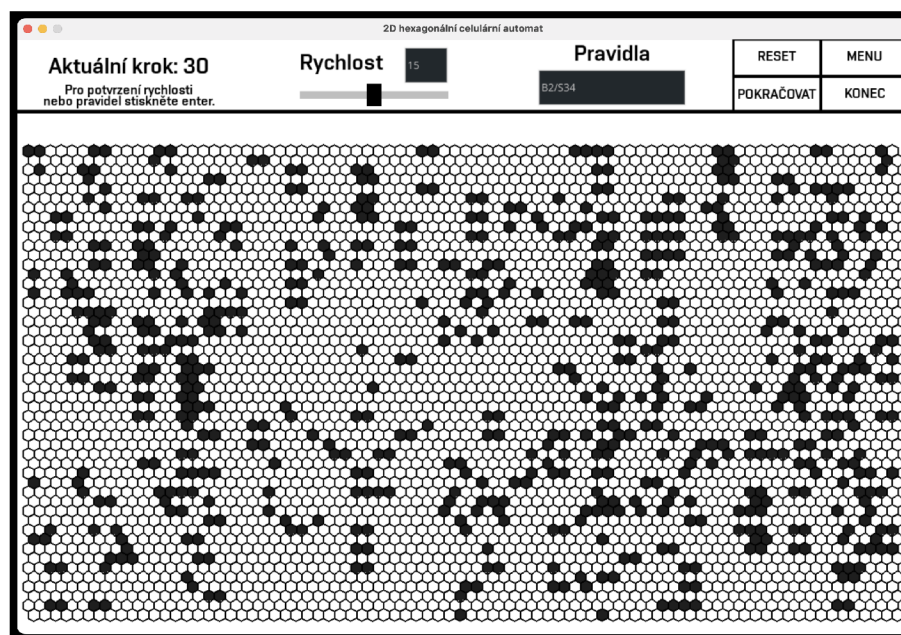
Obr. 36: 2D simulátor s ortogonální mřížkou, pravidlo „B3/S23“.

Po spuštění samotného simulátoru se zobrazí ortogonální mřížka buněk, které se mění podle aktuálního pravidla a sada uživatelských nastavení, podobná jako u 1D simulátoru. Tlačítkem „PAUZA / POKRAČOVAT“, nebo stisknutím klávesy „Mezerník“ lze simulaci pozastavit a opět spustit. Tlačítkem „RESET“ lze simulaci zastavit a vrátit do výchozího stavu. Pravidla simulace jdou měnit v průběhu simulace zadáním pravidla do textového pole opět ve tvaru „Bxx/Syy“. Rychlost simulace lze opět nastavit pomocí posuvníku, nebo zadáním hodnoty do textového pole. Vedle nastavení rychlosti je zobrazen aktuální krok simulace a instrukce pro potvrzení pravidel a rychlosti pomocí tlačítka „Enter“. V případě nesprávně zadané rychlosti nebo pravidel se zobrazí chybová hláška a neproběhne žádná změna.

V případě, že simulátor dojde do stavu, kdy na mřížce jsou pouze oscilátory nebo stojící životy, se simulace pozastaví a objeví se hláška s počtem cyklů, které vedly k zacyklení simulace. Po opětovném spuštění simulace pokračuje cyklicky, dokud se nepozastaví manuálně, nebo není uživatelsky upravené prostředí. Maximální počet identifikovaných cyklů je 40. Jakmile se objeví cyklus, který přesahuje tuto hodnotu, nebude identifikován a simulace bude pokračovat do nekonečna.

4.4 2D simulátor s hexagonální mřížkou

Po výběru náhodného vzoru nebo prázdného pole 2D hexagonální mřížky se zobrazí obdobná možnost výběru pravidel jako u 2D ortogonálního simulátoru (viz kap 4.3). Uživatel zadá pravidla, rychlost a rozměr mřížky. Sousedství je zde pouze jedno a vždy se uvažuje všech šest sousedů buňky. Po stisknutí tlačítka „START“, nebo stisknutí klávesy „Enter“ se spustí simulace.

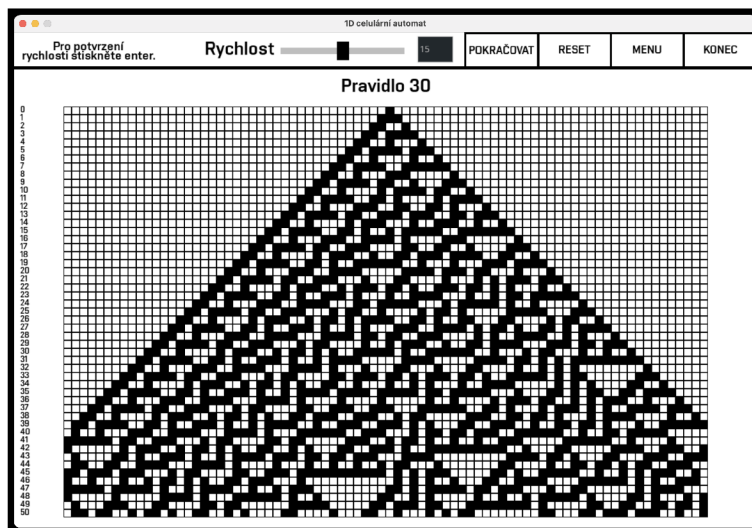


Obr. 37: 2D hexagonální simulátor, pravidlo „B2/S34“.

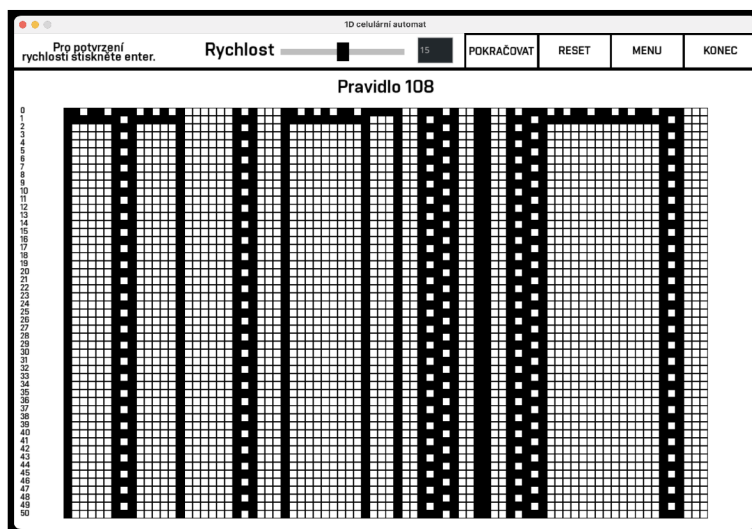
Po spuštění simulace se zobrazí hexagonální mřížka buněk, které se mění podle aktuálního pravidla a sady uživatelských nastavení, podobná jako u 2D ortogonálního simulátoru. Tlačítkem „PAUZA / POKRAČOVAT“, nebo stisknutím klávesy „Mezerník“ lze simulaci pozastavit a opět spustit. Tlačítkem „RESET“ lze simulaci zastavit a vrátit do výchozího stavu. Pravidla simulace jdou měnit v průběhu simulace zadáním pravidla do textového pole opět ve tvaru „Bxx/Syy“. Rychlost simulace lze opět nastavit pomocí posuvníku, nebo zadáním hodnoty do textového pole. Vedle nastavení rychlosti je zobrazen aktuální krok simulace a instrukce pro potvrzení pravidel a rychlosti pomocí tlačítka „Enter“. V případě nesprávně zadané rychlosti nebo pravidel se zobrazí chybová hláška a neproběhne žádná změna. Hexagonální mřížka stejně jako ostatní využívá periodické okrajové podmínky. Při zacyklení simulátoru se simulace pozastaví a objeví se hláška s počtem cyklů, které vedly k zacyklení simulace, stejně jako u ortogonálního simulátoru. Počet identifikovaných cyklů je rovněž 40.

4.5 Ukázka simulátoru

4.5.1 1D mřížka



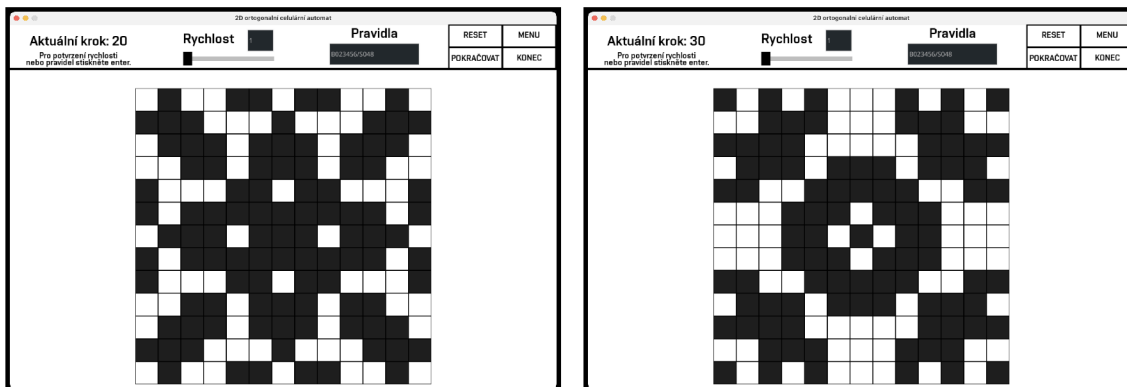
Obr. 38: Pravidlo „30“, které nemá opakující se tendenci.



Obr. 39: Pravidlo „108“, které vytváří jednoduchý vzor a spadá tak do 2. třídy.

4.5.2 2D ortogonální mřížka

Pravidlo „B023456/S048“, které z jedné živé buňky na mřížce 13×13 tvoří symetrické obrazce [30].

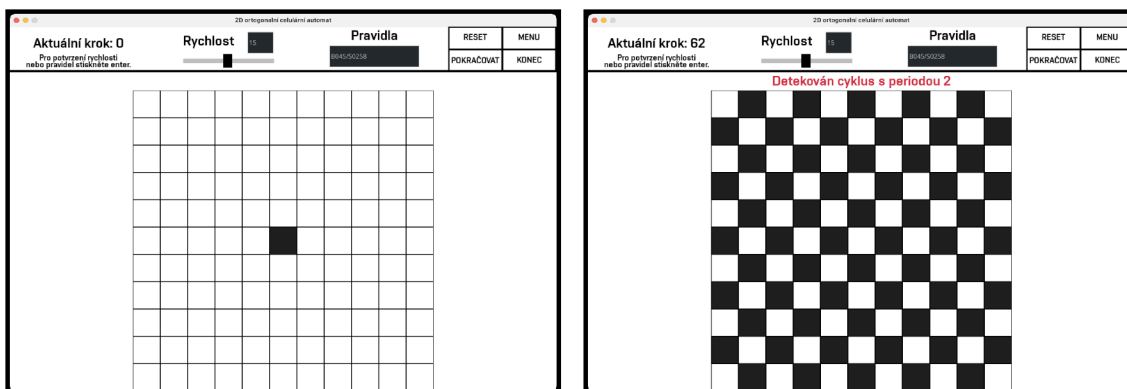


(a) Krok 20

(b) Krok 30

Obr. 40: Pravidlo „B023456/S048“.

Pravidlo „B045/S0258“, které z jedné živé buňky na mřížce 11×11 utvoří po několika iteracích šachovnicový vzor, který každý krok střídá živé a mrtvé buňky [30].



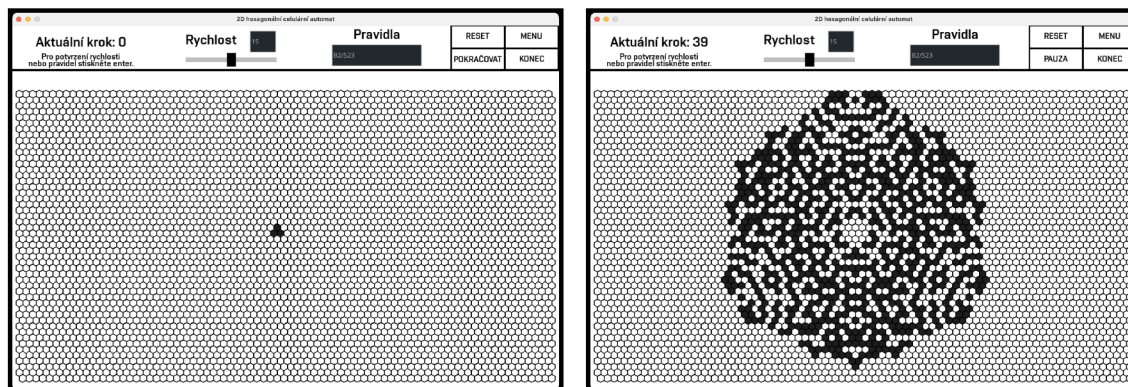
(a) Krok 0

(b) Krok 62

Obr. 41: Pravidlo „B045/S0258“.

4.6 2D hexagonální mřížka

Pravidlo „B2/S23“ z mála živých buněk na počátku tvoří vzory, které se rozrůstají do nekonečna [31].

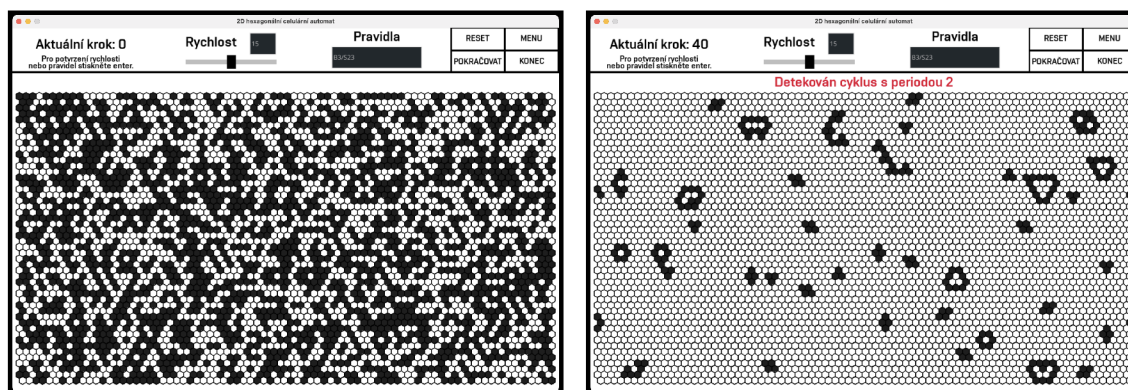


(a) Krok 0

(b) Krok 39

Obr. 42: Pravidlo „B2/S23“.

Pravidlo „B3/S23“, které v ortogonální mřížce značí Hru Života, se na hexagonální mřížce z náhodného počátečního stavu v několika málo krocích ustálí a netvoří žádné nové vzory, ani život nezmizí [31].



(a) Krok 0, náhodná inicializace mřížky

(b) Krok 40

Obr. 43: Pravidlo „B3/S23“.

5 ZÁVĚR

Cílem této bakalářské práce bylo navrhnout simulátor celulárních automatů (CA) v jednorozměrném (1D) a dvourozměrném (2D) světě. V rámci práce byly splněny všechny stanovené cíle, které zahrnovaly jak teoretické, tak praktické aspekty problematiky.

První část práce se zaměřila na teoretický úvod do problematiky celulárních automatů. Byla provedena rešerše historie celulárních automatů, jejich základní principy, klasifikace a bylo prezentováno jejich využití. Dále byly stručně popsány vybrané knihovny pro práci s celulárními automaty v programovacím jazyce Python, které jsou možnou alternativou, tato část zahrnovala knihovny jako CellPyLib, Golly, Pygame a Numpy.

V praktické části byla realizována vlastní implementace simulátoru celulárních automatů. Bylo vytvořeno uživatelsky přívětivé grafické rozhraní (GUI), které umožňuje definovat rozměry automatu, topologii světa, okolí buňky a pravidla pro simulaci. Simulátor byl navržen tak, aby byl uživatelsky přívětivý a jednoduchý na použití, zároveň ale byl flexibilní a snadno rozšiřitelný, což uživatelům umožňuje jednoduché experimentování s různými nastaveními a pravidly CA.

Pro ověření funkčnosti a ukázkou možností simulátoru byly vytvořeny konkrétní příklady zajímavých vzorů pro každý druh mřížky, další vzory tvořené pomocí simulátoru byly použity jako ilustrační obrázky v průběhu práce.

Práce poskytla nejen teoretické základy a praktické nástroje pro práci s celulárními automaty, ale také inspirovala k dalšímu zkoumání a inovacím v této oblasti.

V závěru lze konstatovat, že stanovené cíle práce byly úspěšně splněny. Realizovaný simulátor celulárních automatů představuje užitečný nástroj pro studium a vizualizaci CA a může být dále využíván v akademickém prostředí i v praktických aplikacích.

SEZNAM POUŽITÉ LITERATURY

- [1] *Cellular automaton*. 3.9.2003. Dostupné z: https://en.wikipedia.org/wiki/Cellular_automaton.
- [2] CIOFFI REVILLA, C. *Introduction to Computational Social Science: Principles and Applications*. 1. vyd. London: Springer London, 2014. ISBN 978-1-4471-5661-1.
- [3] ÖSTERLUND, T. DESIGN POSSIBILITIES OF EMERGENT ALGORITHMS FOR ADAPTIVE LIGHTING SYSTEM. *Nordic Journal of Architectural Research*. Listopad 2013, sv. 25, s. 159–184.
- [4] *Šestiúhelníkové prostředí*. Květen 2024. Dostupné z: <https://geometricolor.files.wordpress.com/2013/01/hexagonneumann.jpg>.
- [5] ADAMATZKY, A., ed. *Game of Life Cellular Automata*. London: Springer London, 2010. ISBN 9781849962162 9781849962179. Dostupné z: <https://link.springer.com/10.1007/978-1-84996-217-9>.
- [6] WOLFRAM, S. *A New Kind of Science*. Dostupné z: <https://www.wolframscience.com/nks/>.
- [7] WEISSTEIN, E. W. *Elementary Cellular Automaton*. Publisher: Wolfram Research, Inc. Dostupné z: <https://mathworld.wolfram.com/ElementaryCellularAutomaton.html>.
- [8] GHOSH, J. a ADSULE, R. One dimensional cellular automata-The totalistic approach. *At Right Angles*, 7 (2). pp. 105-116. 2018. ISSN 2582-1873. Dostupné z: http://apfstatic.s3.ap-south-1.amazonaws.com/s3fs-public/19_TOTALISTIC-CELLULAR-AOTOMATA_TECHSPACE.pdf?Qf2TywdfejQnfjSw2ko4o0v73ebFScGi.
- [9] WOLFRAM, S. Random sequence generation by cellular automata. *Advances in Applied Mathematics*. 1986, sv. 7, č. 2, s. 123–169. DOI: [https://doi.org/10.1016/0196-8858\(86\)90028-X](https://doi.org/10.1016/0196-8858(86)90028-X). ISSN 0196-8858. Dostupné z: <https://www.sciencedirect.com/science/article/pii/019688588690028X>.
- [10] WOLFRAM, S. Statistical mechanics of cellular automata. *Rev. Mod. Phys.* American Physical Society. Jul 1983, sv. 55, s. 601–644. DOI: [10.1103/RevModPhys.55.601](https://doi.org/10.1103/RevModPhys.55.601). Dostupné z: <https://link.aps.org/doi/10.1103/RevModPhys.55.601>.

- [11] VICHNIAC, G. Y. Simulating physics with cellular automata. *Physica D: Nonlinear Phenomena*. 1984, sv. 10, č. 1, s. 96–116. DOI: [https://doi.org/10.1016/0167-2789\(84\)90253-7](https://doi.org/10.1016/0167-2789(84)90253-7). ISSN 0167-2789. Dostupné z: <https://www.sciencedirect.com/science/article/pii/0167278984902537>.
- [12] WANG, Q., YU, S., DING, W. a LENG, M. Generating High-Quality Random Numbers by Cellular Automata with PSO. In: *2008 Fourth International Conference on Natural Computation*. 2008, sv. 7, s. 430–433. DOI: 10.1109/ICNC.2008.560.
- [13] PACKARD, N. H. a WOLFRAM, S. Two-dimensional cellular automata. *Journal of Statistical Physics*. březen 1985, sv. 38, č. 5, s. 901–946. DOI: 10.1007/BF01010423. ISSN 1572-9613. Dostupné z: <https://doi.org/10.1007/BF01010423>.
- [14] WÓJTOWICZ, M. *Cellular Automata rules lexicon - Life*. Dostupné z: http://www.mirekw.com/ca/rullex_life.html.
- [15] GARDNER, M. MATHEMATICAL GAMES. *Scientific American*. Scientific American, a division of Nature America, Inc. 1970, sv. 223, č. 4, s. 120–123. ISSN 00368733, 19467087. Dostupné z: <http://www.jstor.org/stable/24927642>.
- [16] ELKIES, N. D. The still-Life density problem and its generalizations. 1999. DOI: 10.48550/ARXIV.MATH/9905194. Dostupné z: <https://arxiv.org/abs/math/9905194>.
- [17] *Oscillator - LifeWiki*. Dostupné z: <https://conwaylife.com/wiki/Oscillator>.
- [18] *Conway's Game of Life*. Květen 2024. Page Version ID: 1222668696. Dostupné z: https://en.wikipedia.org/w/index.php?title=Conway%27s_Game_of_Life&oldid=1222668696.
- [19] *CPI Inflation Calculator*. Dostupné z: <https://data.bls.gov/cgi-bin/cpicalc.pl?cost1=50.00&year1=197001&year2=202402>.
- [20] *Conway's Game of Life*. Květen 2024. Page Version ID: 1222668696. Dostupné z: https://en.wikipedia.org/w/index.php?title=Conway%27s_Game_of_Life&oldid=1222668696.
- [21] MENDES, R. L., SANTOS, A. A., MARTINS, M. a VILELA, M. Cluster size distribution of cell aggregates in culture. *Physica A: Statistical Mechanics and its Applications*. 2001, sv. 298, č. 3, s. 471–487. DOI: [https://doi.org/10.1016/S0378-4371\(01\)00238-2](https://doi.org/10.1016/S0378-4371(01)00238-2). ISSN 0378-4371. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S0378437101002382>.

- [22] GHOSH, R., ADHIKARY, J. a CHEMLAL, R. Fire Spread Modeling using Probabilistic Cellular Automata. 2024. DOI: 10.48550/ARXIV.2403.08817. Dostupné z: <https://arxiv.org/abs/2403.08817>.
- [23] LING, Y., WANG, M., CHEN, Q. a MYNETT, A. Modelling Spatial-Temporal Dynamics of Cyanobacteria Abundance in Lakes by Integrating Cellular Automata and Genetic Programming. In: LOGGIA, G. L., FRENI, G., PULEO, V. a MARCHIS, M. D., ed. *HIC 2018. 13th International Conference on Hydroinformatics*. EasyChair, 2018, sv. 3, s. 1214–1223. EPiC Series in Engineering. DOI: 10.29007/dl2b. ISSN 2516-2330. Dostupné z: <https://easychair.org/publications/paper/GMP6>.
- [24] DAI, J., ZHAI, C., AI, J., MA, J., WANG, J. et al. Modeling the Spread of Epidemics Based on Cellular Automata. *Processes*. 2021, sv. 9, č. 1. DOI: 10.3390/pr9010055. ISSN 2227-9717. Dostupné z: <https://www.mdpi.com/2227-9717/9/1/55>.
- [25] *Wolfram's classification - LifeWiki*. Dostupné z: https://conwaylife.com/wiki/Wolfram%27s_classification.
- [26] ALGORITHMIC NATURE GROUP LABORES FOR THE NATURAL AND DIGITAL SCIENCES a LUVALLE, B. J. The Effects of Boundary Conditions on Cellular Automata. *Complex Systems*. březen 2019, sv. 28, č. 1, s. 97–124. DOI: 10.25088/ComplexSystems.28.1.97. ISSN 08912513. Dostupné z: https://www.complex-systems.com/abstracts/v28_i01_a05/.
- [27] ANTUNES, L. M. CellPyLib: A Python Library for working with Cellular Automata. *Journal of Open Source Software*. The Open Journal. 2021, sv. 6, č. 67, s. 3608. DOI: 10.21105/joss.03608. Dostupné z: <https://doi.org/10.21105/joss.03608>.
- [28] *Golly Help: Python Scripting*. Dostupné z: <https://golly.sourceforge.io/Help/python.html>.
- [29] *Golly (program)*. Duben 2024. Page Version ID: 1217055354. Dostupné z: [https://en.wikipedia.org/w/index.php?title=Golly_\(program\)&oldid=1217055354](https://en.wikipedia.org/w/index.php?title=Golly_(program)&oldid=1217055354).
- [30] GOLIÁŠ, D. *Celulární automaty a pravidla definovaná genetickým algoritmem*. Brno, 2024. Dostupné z: <https://www.vut.cz/studenti/zav-prace/detail/157952>.
- [31] ERDOGAN, V. 2D Hexagonal Cellular Automata: The Complexity of the Forms. 2018. DOI: 10.48550/ARXIV.1811.12387. Dostupné z: <https://arxiv.org/abs/1811.12387>.

SEZNAM PŘÍLOH

- Program simulátoru celulárních automatů
 - Soubory s třídami
 - Složka assets poskytující obrázky pro tlačítka a font