Platformata

Game Design with Automata Theory

Ahmed Shuaib

Introduction

- Automata is widely used to analyze systems, design compilers, and programming languages.
- Modern development tools are developed to optimize the development experience.
- Lets explore the use of methods and tools of automata to achieve.
    - Optimize the development experience for 2D game developers.
    - Build an framework that can be modified and built upon.

Literature Review

[1] In a paper from 2022 A procedural-level generation has been demonstrated using cellular automata.

- A genetic algorithm is used to evolve the cellular automata rules applied to generate game levels.
- A Procedural content generator designed using this method is meant to be used during the game development process rather than at runtime.

[2] In a paper from 2016 students designed an infinite runner game using mealy machines.

- The game consisted of several states such as: running, jumping, and flying.
- Games designed using automata are less prone to bugs and the development process is simplified.

[3] Conway's Game of Life

- Developed by John Conway.
- A "zero-player" game utilizing cellular automata.
- Essentially a square grid containing cells that evolve in each time step based on three rules: birth, death, and survival.
    - Dead cell will become alive -> 3 neighbouring cells are alive.
    - Cell will die -> four or more/one or fewer neighbouring cells are alive.
    - Cell survives -> two or three neighbouring cells are alive.

Game Description

A game design for a 2D Platformer.

- Reach the end most efficiently with:
  - As few attempts as possible.
  - As fast as possible.
- Each state you encounter may take time, the amount of time some state takes generally varies based on the type of state.
- When the player runs out of time, they transition back to thart for another attempt.
- As the levels go on complexity increases.
  - Try various paths
  - Implement strategies

Level 1

- Time: 36 seconds.
- Introduces: Platforms, stairs, jump pads, buttons, ghost platforms.
- Attempts (Estimate): 2

Level 2

- Time: 120 seconds.
- Features: Platforms, stairs, jump pads, buttons, ghost platforms.
- Introduces: Ghost walls and cubes.
- Attempts (Estimate): 2

Level 3

- Time: 90 seconds
- Features: Platforms, stairs, jump pads, buttons, ghost platforms, ghost walls, platforms.
- Introduces: Platform walls.
- Attempts (Estimate): 2-3

Level 4

- Time: 90 seconds.
- Features: Platforms, stairs, jump pads, buttons,ghost walls, platforms, platform walls.
- Attempts (Estimate): 2-3

Game Design

- Scalability and modularity of the game have been considered.
- If one intends to follow the game design that follows:
    - Relatively simple to implement and build on the logic of this game using modren development tools.
    - Ideas: Designing game mechanics that rely on a time-distance/time-state relationship, adding additional states, or utilizing procedural generation.
- The possibilities are endless :)

Input Alphabet

$\sum_{\text{Move}} = \{\, \mathrm{U, D, L, R} \,\}$ Defines the movement of the player.

$\sum_{\text{Action}} = \{\, \mathrm{P, J, O} \,\}$ Defines the actions the player can take.

$\sum_{\text{Game}} = \{\, \mathrm{T_n,\ TA_n,\ A_n,\ B_n,\ NB_n,\ D_n,\ ND_n} \,\}$

↑ Defines some data for the game level. Counters (Integers) and Status (Booleans). This data is reset at the start of each level.

States of the game

Time State ($\mathrm{T}_{\mathrm{n}}^{\mathrm{t}}$)

Effects: $\mathrm{T}_{\mathrm{n}} = \mathrm{t}$ $\quad$ ++TA_{n}

Platform State ($\mathrm{P}_{\mathrm{n}}^{\mathrm{t}}$)

Effect: T_{n}-= t $\quad$ ++A_{n}

Stair State ($\mathrm{S}_{\mathrm{n}}^{\mathrm{t}}$)

Effects: T_{n}-= t $\quad$ ++A_{n}

Button State ($\mathrm{B}_{\mathrm{n}}^{\mathrm{t}}$)

Effects: T_{n}-= t $\quad \mathrm{B}_{\mathrm{n}} = 1 \quad \mathrm{NB}_{\mathrm{n}} = 0$ $\quad$ ++A_{n}

Jump State ($\mathrm{J}_{\mathrm{n}}^{\mathrm{t}}$)

Effects: T_{n}-= t ++A_{n}

Hole state (H_{n})

Effect: ++A_{n}

Ghost Platform State ($\mathrm{GP}_{\mathrm{n}}^{\mathrm{t}}$)

Effects: T_{n}-= t ++A_{n}

Ghost Wall State ($\mathrm{GW}_{\mathrm{n}}^{\mathrm{t}}$)

Effects: T_{n}-= t ++A_{n}

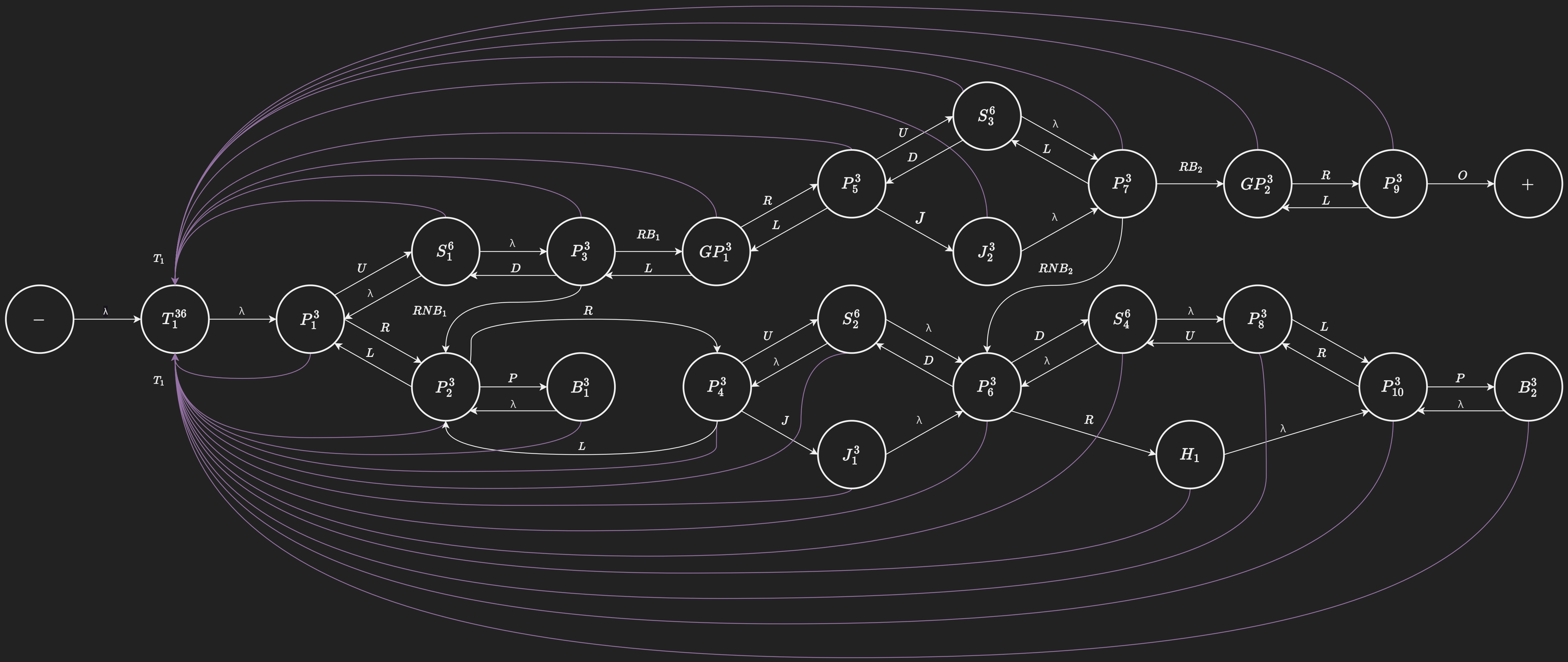
Platform Wall and Cube States ($PW_n^{h^d}$ $C_n^{h^d}$)

Effects:

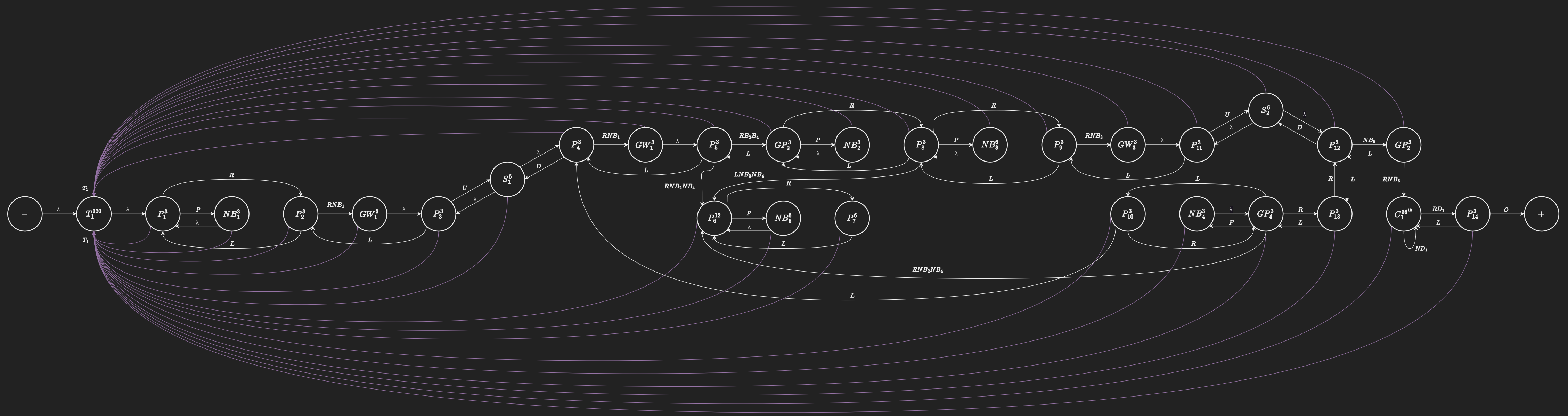
$$T_n \mathrel{-}= (h - (d \cdot (\texttt{++}A_n)))$$

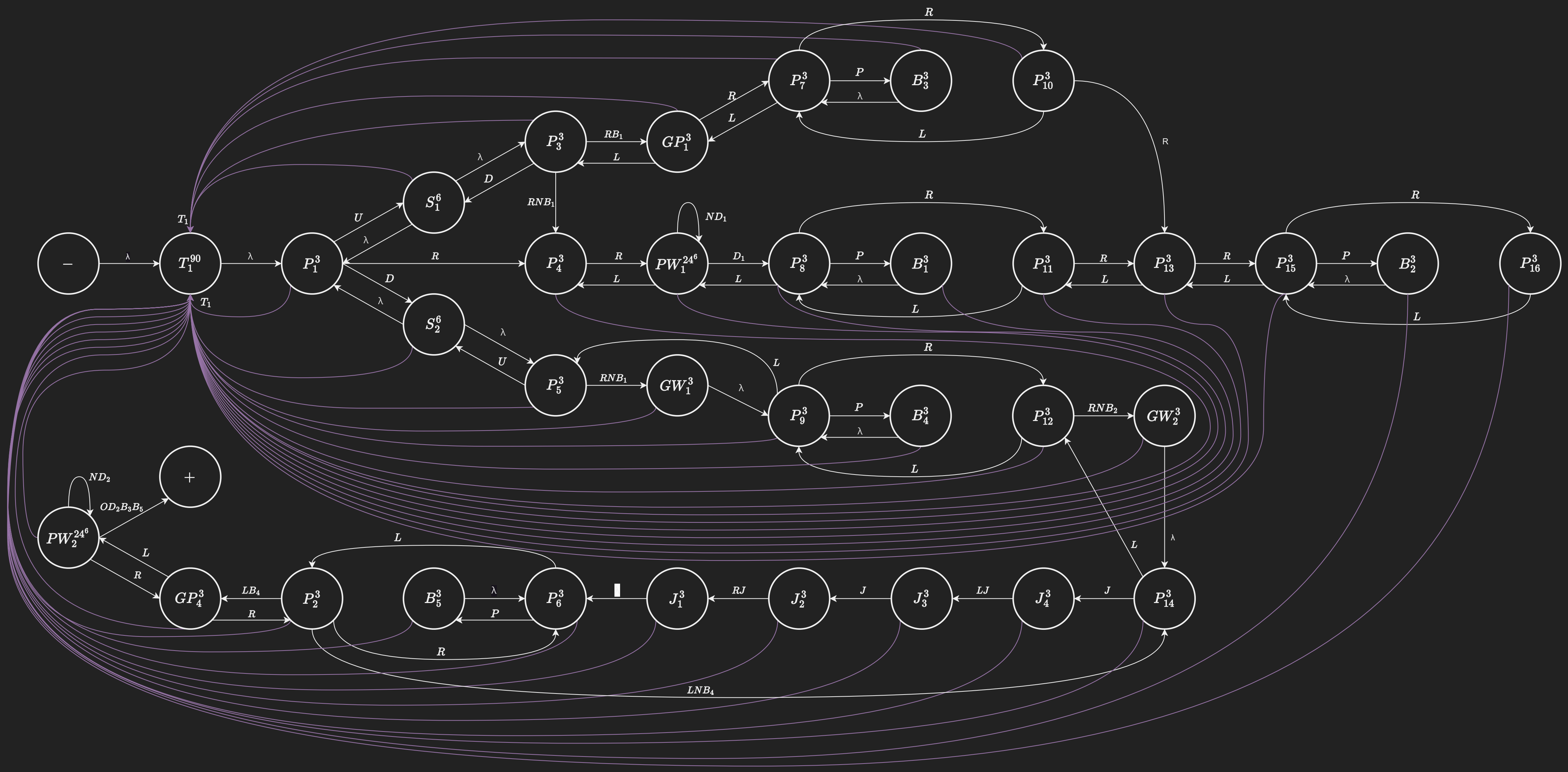
$$D_n = (h - (d \cdot A_n)) \leq 0$$

$$ND_n = (h - (d \cdot A_n)) > 0$$



$-$
T_1^{36}
P_1^3
S_1^6
P_3^3
GP_1^3
P_2^3
B_1^3
P_4^3
S_2^6
J_1^3
P_5^3
S_3^6
J_2^3
P_6^3
P_7^3
S_4^6
H_1
GP_2^3
P_8^3
P_9^3
P_{10}^3
B_2^3
$+$
T_1
RB_1
RNB_1
RB_2
RNB_2
λ
U
D
L
R
P
J
O


−
+
T_1^{120}
P_1^3
NB_1^3
P_2^3
GW_1^3
P_3^3
S_1^6
P_4^3
P_5^3
GP_2^3
NB_2^3
P_8^3
NB_3^6
P_9^3
GW_3^3
P_{11}^3
S_2^6
P_{12}^3
P_6^{12}
NB_5^6
P_7^6
P_{10}^3
NB_4^3
GP_4^3
P_{13}^3
P_{14}^3
RNB₁
RB₂B₄
RNB₂NB₄
LNB₂NB₄
NB₅
RNB₅
RD₁
ND₁


−
+
T_1^{90}
T_1
P_1^3
P_2^3
P_3^3
P_4^3
P_5^3
P_6^3
P_7^3
P_8^3
P_9^3
P_{10}^3
P_{11}^3
P_{12}^3
P_{13}^3
P_{14}^3
P_{15}^3
P_{16}^3
S_1^6
S_2^6
GP_1^3
GP_4^3
GW_1^3
GW_2^3
$PW_1^{24^6}$
$PW_2^{24^6}$
B_1^3
B_2^3
B_3^3
B_4^3
B_5^3
J_1^3
J_2^3
J_3^3
J_4^3
RB_1
RNB_1
RNB_2
LB_4
LNB_4
D_1
ND_1
ND_2
$OD_2B_3B_5$
RJ
LJ
J
U
D
R
L
P
λ

$-$
T_1^{90}
T_1
P_1^3
H_1
H_2
H_3
J_1^3
J_2^3
J_3^3
J_4^3
P_2^3
P_3^3
P_4^3
P_5^3
P_6^3
P_7^3
B_1^3
B_2^3
GW_1^3
GW_2^3
GW_3^3
GW_4^3
P_8^{12}
P_9^{12}
J_6^3
J_7^3
J_8^3
J_9^3
P_{10}^3
P_{11}^3
P_{12}^3
NB_3^3
NB_4^3
$PW_1^{24^6}$
$+$
λ
L
R
P
J
LJ
RJ
RB_3
RNB_2
RNB_3
RNB_4
LB_1
RB_4
LNB_1
OD_1
ND_1

Conclusion

- Incorporating automata into game design offers:
  - A pleasant gameplay experience.
  - Streamlines the design and development process.
- Using methods and tools of automata is a worthwhile approach to achieve an optimized development experience for game developers.

References

- [1] Adel Sabanovic and Amir Khodabakhshi, Evolved cellular automata for 2D video game level generation, 2022.
- [2] Abid Jamil, Engr. AsadUllah and Mohsin Rehman, An Infinite Runner Game Design using Automata Theory, 2016.
- [3] Kuldeep Vayadande, Ritesh Pokarne, Tanmay Patil, Mahalakshmi Phaldesai, Prachi Kumar, and Tanushri Bhuruk, Simulation of Conway's Game of Life using cellular automata, 2022.

This project would be have been difficult without the help of the following tools:

- Drawio
- $\LaTeX$ and pdfTeX
- Marp

All these tools are open source (Support open source software)

Thank you :)