Journal of Algorithms and Computations
journal homepage: http://jac.ut.ac.ir

Modeling of the Wumpus World to Evaluate the Difficulty Level of the Game using Coloured Petri Net

A. Naghash Asadi[†1], M. Pourjani[*1] and A. Mirsaeed Ghazi[‡1]

[1] Fouman Faculty of Engineering, College of Engineering, University of Tehran, Tehran, Iran

ABSTRACT

The video game industry (VGI) faces significant challenges in today's world. Game designers continually strive to create innovative scenarios that engage players and enhance their motivation to play. However, developing such scenarios is a complex task, and evaluating their difficulty during the design phase is equally challenging. In this paper, we focus on the Wumpus World game and propose a model for it using Coloured Petri Nets (CPNs). The proposed model enables designers to simulate various game scenarios and assess their difficulty levels without modifying the underlying structure by simply changing the tokens of the model. Since in most games, players need to switch between. Since, in most games, players must transition between different states to achieve their objectives, the mapping approach introduced here can also be applied to model and simulate other games.

Keywords: Wumpus world, game design, modeling and simulation, Coloured Petri net (CPN), difficulty level.

ARTICLE INFO

Article history:
Research paper
Received 06, October 2025
Accepted 08, November 2025
Available online 24, November 2025

AMS subject classification: 68M10

[†] Corresponding author: A. Naghash Asadi

Email: naghashasadi@ut.ac.ir.

Journal of Algorithms and Computation 57 issue 2, December 2025, PP. 66-79

1 Introduction

Today, the video game industry (VGI) is a leading industry and a rapidly growing business branch that has created new wide businesses [1]. The Worldwide market segment for the video games market (VGM) showcases these significant figures and trends [2] [3]:

- The VGM is projected to reach a revenue of US $282.30 billion in 2024.
- It is expected to grow at an annual rate of 8.76% between 2024 and 2027, resulting in a projected market volume of US $363.20 billion by 2027.
- The number of users in the VGM is expected to reach 1,472 million users by 2027.
- The user penetration rate is predicted to increase from 16.9% in 2024 to 18.5% by 2027.
- Among all market segments, In-game Advertising holds the largest share with a market volume of US $109.60 billion in 2024.
- In a global comparison, China is projected to generate the most revenue, with US $94,490 million in 2024.
- The average revenue per user (ARPU) in the VGM is projected to be US $215.20 in 2024.

Therefore, the VGI warrants detailed and systematic study. Unfortunately, the challenges and issues in this field often lack a solid foundation in mathematical and theoretical frameworks. In other words, conceptual models grounded in logical reasoning, mathematical formalisms, and prior knowledge are frequently overlooked, while emphasis is placed primarily on design and implementation. One of the most significant challenges in the VGI is designing new game scenarios. Designers aim to create scenarios that challenge players and enhance their motivation to play; however, this process is inherently complex. Moreover, accurately assessing the difficulty level of a scenario during its design stage is a nontrivial task. To address this, mathematical and theoretical approaches can be employed. Among the most powerful of these are Petri nets (PNs) and their extensions, such as Coloured Petri nets (CPNs). These formalisms offer robust tools for modeling complex systems and have been successfully applied in many domains of computer science and related disciplines [4] [5] [6] [7] [8] [9]. One advantage of modeling and simulating a system prior to implementation is the ability to evaluate outcomes while conserving time and budget. According to the general scheme presented in Figure 1, in this paper, we focus on the Wumpus World game and propose a CPN-based model to evaluate the difficulty levels of different scenarios. Since, in most games, players must transition from one state to another to achieve their objectives, the mapping approach described here can also be applied to model and simulate other games. The proposed model enables us to assess the difficulty of various scenarios without the need for actual implementation. The main contribution of this paper can be summarized in the four following categories:

- A game has been considered in terms of mathematical and theoretical theories.
- The movement of an agent from one state to another to reach the goal is modeled using the CPN formalism.
- The proposed CPN model can be applied to a game to evaluate and compare the difficulty level of different game scenarios without changing the model.
- Only one model is generated for a game with specific assumptions.

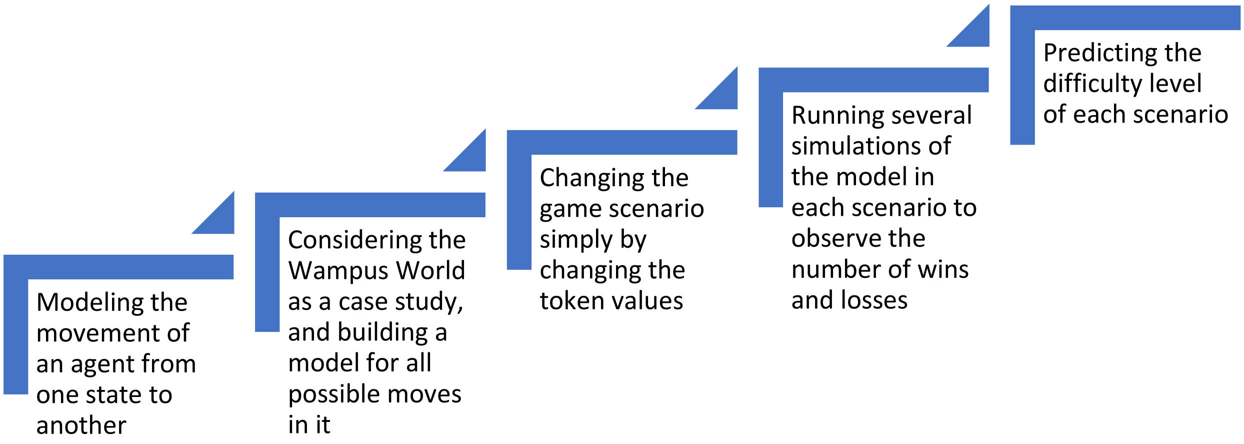


Figure 1 The general scheme of this paper

The remaining part of this paper is organized as follows. In Section 2, background information on the Wumpus world and preliminaries of CPN formalism is provided. In section 3, related works are introduced. In Section 4, the CPN model proposed for the movement of an agent from one room to another in the Wumpus world is presented. In Section 5, different scenarios for the Wumpus world are designed and their simulation results are compared to evaluate their difficulty level. Finally, Section 6 presents conclusions and points to be explored in future research.

2 Background

2.1 The Wumpus world and our assumption

A variety of worlds are being used as examples for knowledge representation, reasoning, and planning. One of them is the Wumpus world in the field of artificial intelligence [10] [11]. Examples of Wumpus world simulators are presented in [12] and [13]. According to Figure 2 [14], the Wumpus world is a world with n^2 rooms ($n \times n$), and each room is connected to others through walkways (no rooms are connected diagonally). There are some pits, a beast named Wumpus, and a treasure in this world. If the agent enters the pit, it gets stuck there. If the agent enters the Wumpus room, it is eaten there. We assume that the Wumpus cannot move between rooms. The goal of the agent that starts from room [1,1] is to take the treasure. In this paper, we assume that the agent is memoryless and forgets the path it has traversed. However, the agent has facilities to explore pits and Wumpus:

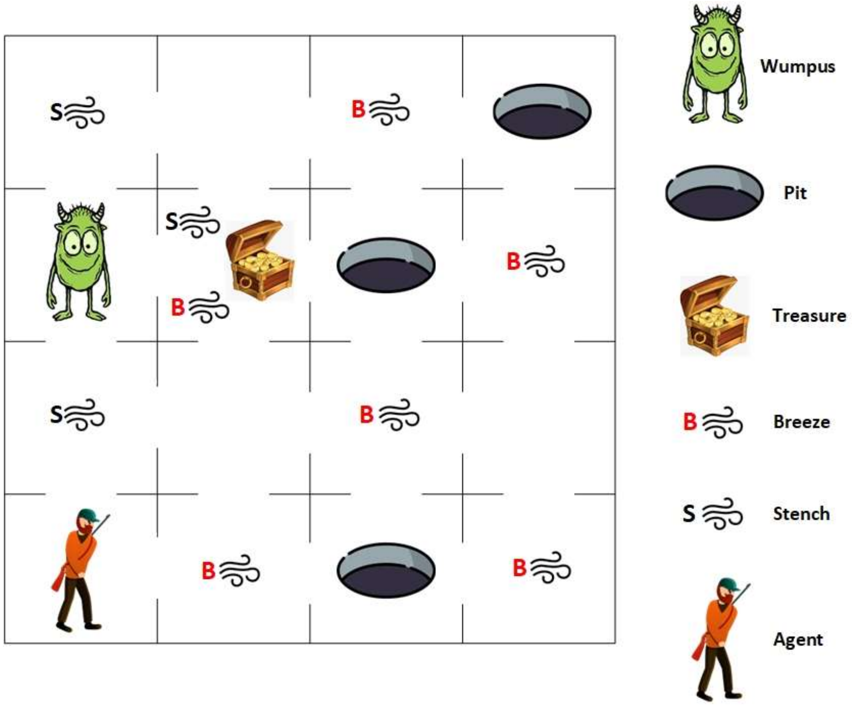


Figure 2 An example of the Wumpus world

- The pit's adjacent rooms are filled with a breeze.
- The Wumpus's adjacent rooms are stenchy.
- The agent is given one arrow. If the agent is in a room where there is a stench, it uses its arrow to kill the Wumpus before going to the next room. So, the agent may fire the arrow in the wrong direction.

2.2 The CPN Formalism and CPN Tools

There are three elements including places, transitions, and tokens in the basic definition of the PNs [15]. Places are represented by circles and can contain zero or several tokens to model the states. Transitions are represented by rectangles and used to model the events/actions that change the number of tokens inside places. The CPN [16] is an important extension of PNs to model heterogeneity that has a simple definition, and graphical representation and is supported by the CPN Tools [17]. In the CPN formalism, unlike the PN, tokens are not atomic and can have a data structure. Figure 3 shows the graphical representation of the CPN elements in the CPN Tools [4].

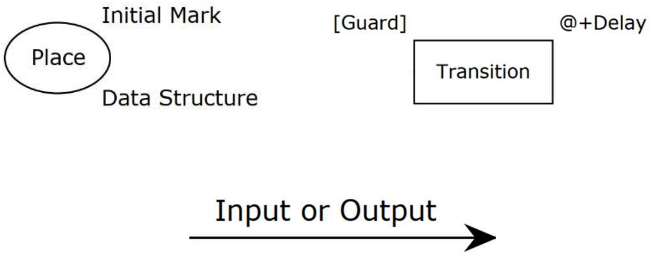


Figure 3 Graphical representation of the CPN elements in the CPN Tools [4]

There are According to Figure 3, each place in the CPN model has a name, a data structure, and an initial mark. The data structure of each place can include primary (for example, int, string, and so on) or secondary (for example, an ordered pair (string, int) which is defined as *product string*int* in the CPN Tools) data types. For example, if the data structure of a place is product string*int, it can hold the different number of tokens (N) with different values of tokens (Value) as N`Value like 2`("test", 12).

According to Figure 3, each transition in the CPN model has a name, a guard function, and a delay. The guard function of each transition is a conditional statement that if it is true, the transition is active and can fire. When the transition fires, some tokens are removed from the input places and some tokens are added to the output places. Furthermore, if the input places of a transition have time and the transition has a delay, after firing the transition and transferring tokens, the amount of the transition delay is added to the activation time of the tokens.

3 Related Work

In this section, related works that use mathematical theories, such as PNs and their different extensions, to model games are introduced. However, the movement of an agent from one state to another to reach the goal is not modeled to evaluate the difficulty level of the game scenarios in any of them. In [18], an approach has been presented and evaluated to manage the plot structure of character-based interactive narratives in games, which combines multi-agent planning with a drama management strategy based on narrative structures. A plot composition method based on situation calculus and Petri net models has been proposed in [19] to a narrative open to user co-authorship. In [20], an alternative approach with PNs has been described for the modeling of serious games and classification of motivation behavior to assess the motivation level of player ability. The PN model of motivation behavior game and the function of motivation behavior identification have been used in motivation behavior game modeling research. Furthermore, the learning vector quantization has been used to classify player's characteristics in playing games, so that the game can identify the players' motivation behavior. A language based on PNs has been presented in [21] which allows for intuitive and effective robot and multi-robot behavior design. It supports a set of features that are critical for developing robotic applications, including sensing, interrupts, and concurrency. In the domain of multi-robot path-planning problems, robots must move from their start locations to their goal locations while avoiding collisions with each other. In [22], the football dream team is evaluated on the basis of individual abilities and interplayer synergy using graph theory and vectorial distances. In [23], a black-box prediction model that can accurately predict the difficulty level of words in the popular word-filling game, Wordle, to find the deep rules in the game data. An approach has been presented in [24] based on PNs for the design process of video games to produce an estimated time that corresponds to the effective duration a player will need to complete a specific level of a game. In [25], a quest generation method depending on the player's involvement or type determined by the Bayesian network has been proposed using PNs to automatically generate game content. In other words, a system has been presented to infer from the player's involvement in the game and the fun derived and to generate the plot of a quest suitable to the player's involvement. The proposed system automatically generates a plot of a quest and allows players to enjoy new experiences from diverse quests. An authoring tool and its domain-specific language have been presented in [26] to assist designers to model learning games with PNs. In other words, a methodological framework has been presented to provide players with adaptive feedback and its core relies on modeling the learning game with a PN. Its results show that contributions help designers to build the PN in combination with classical PN editors which are still useful to visualize, to check, and to validate the PNs built. The combining the complementary strengths of PNs and serious games has been examined in [27] to produce a serious game prototype of a complex system design. In other words, the challenge of assuring accurate design of complex systems, integrating PNs and serious games to address the problem, and applying the integration in a proof-of-concept case study have been introduced. A logical PN model has been proposed in [28] to leverage the modeling advantages of Petri nets in handling batch processing and uncertainty in value passing and to integrate relevant game elements from multi-agent game processes for modeling multi-agent decision problems and resolving optimization issues in dynamic multi-agent game decision-making.

4 The proposed CPN model

As you know, in almost all games players have to move from one state to another to reach the goal, so to model a game, it is necessary to model the agent movements, under the direction of the player, to change state. In Figure 4, the CPN model proposed for the movement of an agent from one room to another in the Wumpus world has been presented. In this model, the agent can be in room *R1* and it can go to room *R2* or vice versa. In the Wumpus world, there is only one agent and it can be present in a room. According to the CPN model proposed in Figure 4, each place (*R1* or *R2*) models a room in the Wumpus world and it has a Coloured token, named *Status*, which specifies its status.

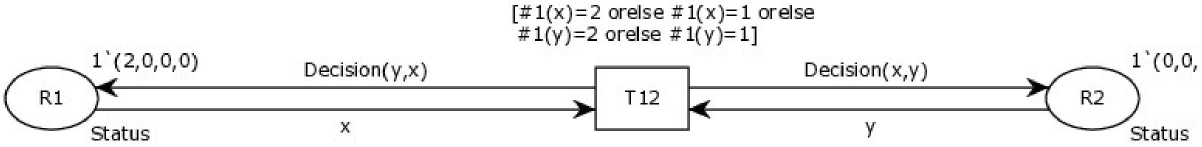


Figure 4 The CPN model proposed to the change of an agent's room in the Wumpus world

The data structure of the Coloured token is a tuple (int, int, int, int) which is defined as *product int*int*int*int* in the CPN Tools. According to Table 1, the values of the tuple of each place specify the agent's status, the Wumpus's status, the pit's status, and the treasure's status in each room.

- If the first value of the tuple is
  - zero, it means that the agent is not present in this room.
  - one, it means that the agent is present in this room but has no arrows.
  - two, it means that the agent is present in this room and has an arrow.
  - three, it means that the agent is present in this room but has stopped.
- If the second value of the tuple is
  - zero, it means that the Wumpus is not present in this room and this room is not stenchy.

    - one, it means that the Wumpus is not present in this room but this room is stenchy.
    - two, it means that the Wumpus is present in this room.
- If the third value of the tuple is
    - zero, it means that the pit is not present in this room and this room has no breeze.
    - one, it means that the pit is not present in this room but this room has a breeze.
    - two, it means that the pit is present in this room.
- If the fourth value of the tuple is
    - zero, it means that the treasure is not present in this room.
    - two, it means that the treasure is present in this room.

Table 1: The data structure of the Coloured token of each place

(int,	int,	int,	int)
Agent: 0= not present, 1= present, no arrows, 2= present, an arrow, 3= present, stop.	**Wumpus:** 0= not present, not stenchy, 1= not present, stenchy, 2= present.	**Pit:** 0= not present, not breeze, 1= not present, breeze, 2= present.	**Treasure:** 0= not present, 2= present.

According to the CPN model proposed in Figure 4, each transition (*T1_2*) models the movement of the agent from one room (*R1/R2*) to another (*R2/R1*) in the Wumpus world. For this purpose, each transition checks the values of the tokens of two related rooms. The values *x* and *y* represent the status of rooms *R1* and *R2*, respectively. The #i(x) and #i(y) specify the i_{th} value of the tuple *x* and *y*, respectively. If the agent is present in the *R1* or *R2* room, this transition (*T1_2*) will be enabled. This condition is checked by a guard function according to Eq. (1). In other words, if the agent is present in the *R1* room and has an arrow ($\#1(x) = 2$), or the agent is present in the *R1* room but has no arrows ($\#1(x) = 1$), or the agent is present in the *R2* room and has an arrow ($\#1(y) = 2$), or the agent is present in the *R2* room but has no arrows ($\#1(y) = 1$), the output of the guard function is true and the transition will be enabled. If the transition is enabled, it means that the agent is in one of the rooms (*R1* or *R2*) and it can go to the next room (*R2* or *R1*). Of course, it is possible that several transitions are enabled, in this case, one of them can fire randomly. For example, if the agent is present in the *R1* room, it might be able to go to the room *R2* or *R3*. So, in this case, transitions *T1_2* and *T1_3* will be enabled.

$$[\#1(x) = 2\ orelse\ \#1(x) = 1\ orelse\ \#1(y) = 2\ orelse\ \#1(y) = 1] \qquad (1)$$

```
fun Decision(src:Status, dst:Status)=
let in
case typeRoom(dst) of
3=>(case #1(src) of
        2=>1`(3,#2(dst),#3(dst),#4(dst))
        |1=>1`(3,#2(dst),#3(dst),#4(dst))
        |0=>1`(0,#2(dst),#3(dst),#4(dst)))
```

```
fun typeRoom(r:Status)=
let in
if #2(r)=2 then 1
else (if #3(r)=2 then 2
      else (if #4(r)=2 then 3
            else 0))
end
```

```
|2=>(case #1(src) of
        2=>1`(3,#2(dst),#3(dst),#4(dst))
        |1=>1`(3,#2(dst),#3(dst),#4(dst))
        |0=>1`(0,#2(dst),#3(dst),#4(dst)))
|1=>(case #1(src) of
        2=>1`(1,0,#3(dst),#4(dst))
        |1=>1`(3,#2(dst),#3(dst),#4(dst))
        |0=>1`(0,#2(dst),#3(dst),#4(dst)))
|0=>(case #1(src) of
        2=>(case #2(src) of
            1=>1`(1,#2(dst),#3(dst),#4(dst))
            |0=>1`(2,#2(dst),#3(dst),#4(dst)))
        |1=>1`(1,#2(dst),#3(dst),#4(dst))
        |0=>1`(0,#2(dst),#3(dst),#4(dst)))
End
```

Figure 5 The functions of Decision and typeRoom

When the transition (*T1_2*) fires, tokens of places *R1* and *R2* (*x* and *y*, respectively) are removed and then new tokens are deposited into them according to the function *Decision(src, dst)*. Figure 5 shows functions of *Decision(src, dst)* and *typeRoom(r)*. The *typeRoom(r)* function, used in the *Decision(src, dst)* function, determines the status of a room by receiving the token associated with its place. If there is:

- Wumpus in this room, the value 1 is returned.
- Pit in this room, the value 2 is returned.
- Treasure in this room, the value 3 is returned.
- Else, the value 0 is returned.

For example, if the agent is present in the *R1* room, transitions *T1_2* is enabled and can fire. As mentioned, when the transition *T1_2* fires, tokens of places *R1* and *R2* are removed and then new tokens are deposited into them according to the function *Decision(src, dst)*. The token *x* represents the status of room *R1* as the source (parameter *src* in the *Decision(src, dst)* function) of the agent's movement. Similarly, the token *y* represents the status of the room *R2* as the destination (parameter *dst* in the *Decision(src, dst)* function) of the agent's movement. According to the *Decision(src, dst)* function in Figure 5, the value of the new token to be deposited into the destination room depends on the status of the agent in the source room and the status of the destination room. For a new token to be deposited into the source room, only the status of the agent changes so that it is no longer present in the source room $(\#1(src) = 0)$. It is important to mention that the agent is not aware of the status of the destination room and can only feel the stench or the breeze in the source room. Figure 6 describes the algorithm of the *Decision(src, dst)* function. According to the *Decision(src, dst)* function:

3) If there is treasure in the destination room and the agent goes to it, the agent will be a winner and stopped.

2) If there is a pit in the destination room and the agent goes to it, the agent will get stuck there.

1) If there is Wumpus in the destination room and the agent goes to it,
    - If the agent has an arrow, it kills Wumpus.
    - If the agent does not have an arrow, it will be eaten there.

0) If there is nothing in the destination room and the agent goes to it,
    - If the agent has an arrow and there is a stench in the source room, it uses an arrow and then goes to the room.
    - If the agent has an arrow and there is no stench in the source room, it does not use an arrow and goes to the room.
    - If the agent does not have an arrow, it just goes to the room.

According to the above description, the proposed CPN model for the scenario displayed in Fig. 2 of the Wumpus world is shown in Figure 7. In this model, the red places, including *R3*, *R11*, and *R16* have a pit, the olive-colored place, *R9*, has Wumpus, and the green place, *R10*, has treasure. Furthermore, the token of each place specifies the status of its room. For example, there is a token 1\`(0,1,1,2) in place *R10* that means the absence of the agent, the presence of stench and breeze, and the presence of treasure in this room. Furthermore, it is clear how the places are connected to each other. As mentioned earlier, each room is connected to others through walkways (no rooms are connected diagonally).

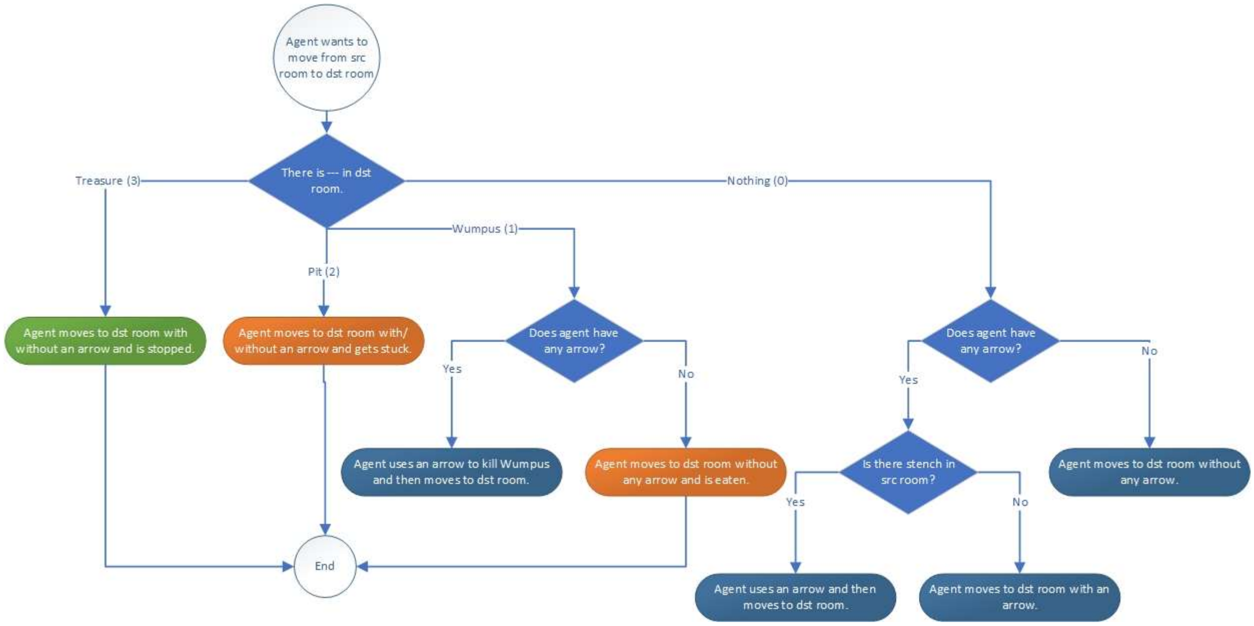


Figure 6 The algorithm of the Decision(src, dst) function presented in Fig. 5

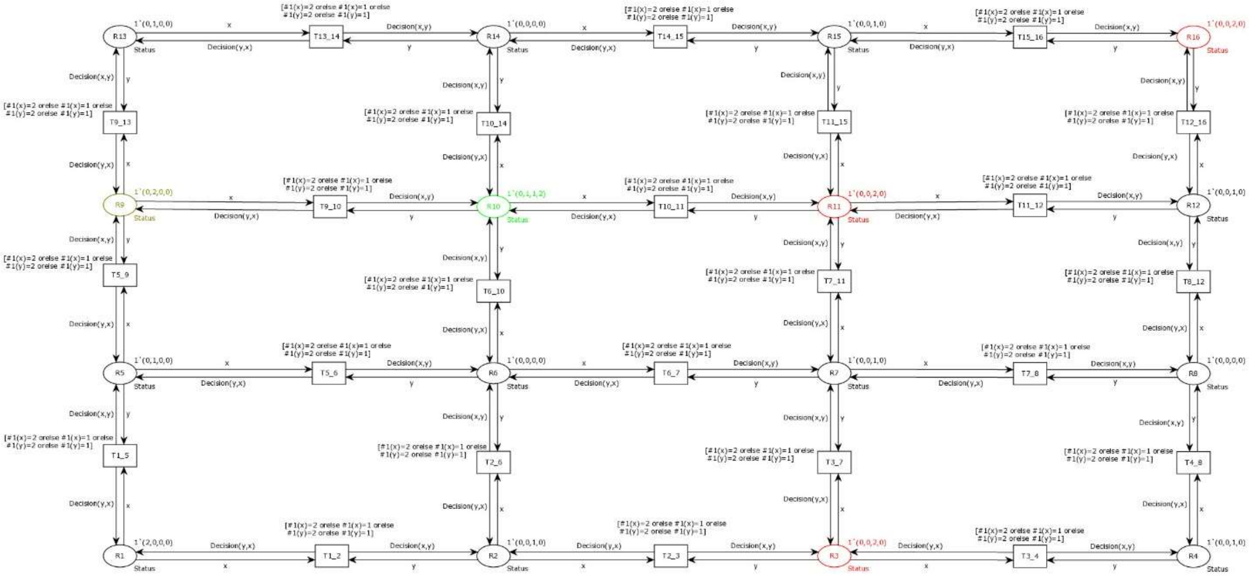


Figure 7 The CPN model proposed to the Wumpus world shown in Fig. 2

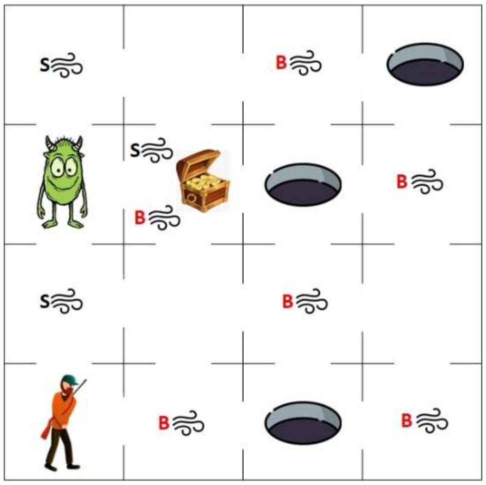
5 Evaluation

In this section, we first design several scenarios for the Wumpus World and then simulate them using the CPN model presented in Figure 7 within CPN Tools. Figure 8 illustrates six scenarios developed for this purpose. Each scenario is simulated 30 times, and the results are recorded. To apply a given scenario to the proposed CPN model, only the token values need to be modified. The resulting simulations provide measures of each scenario's difficulty, which are then compared. Table 2 presents the simulation results for the various Wumpus World scenarios. As shown, Scenario 2 has the lowest probability of winning (0.03), indicating that it is the most difficult among the six scenarios. Conversely, Scenario 6 has the highest probability of winning (0.94), making it the least difficult. Additionally, the table reports the minimum, maximum, and average numbers of steps taken in each scenario for both winning and losing outcomes.

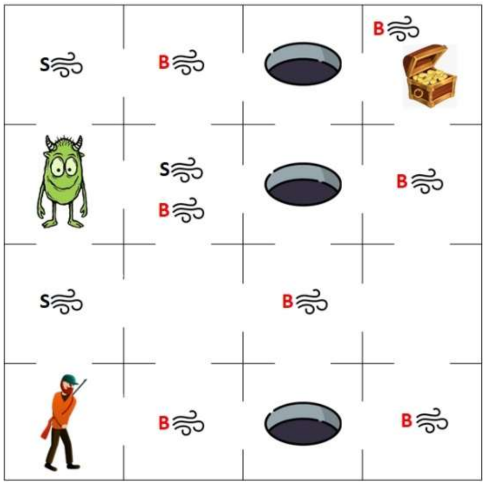
Table 1: The data structure of the Coloured token of each place

# Scenario	Winning and Losing		Number of Steps Traversed		
	Number	Probability	Min	Max	AVG
1	Win: 10 Lose: 20	Win: 0.33 Lose: 0.67	Win: 3 Lose: 2	Win: 15 Lose: 14	Win: 5.8 Lose: 7.2
2	Win: 1 Lose: 29	Win: 0.03 Lose: 0.97	Win: 12 Lose: 2	Win: 12 Lose: 15	Win: 12 Lose: 7
3	Win: 5 Lose: 25	Win: 0.14 Lose: 0.86	Win: 5 Lose: 2	Win: 23 Lose: 16	Win: 12.2 Lose: 5.68

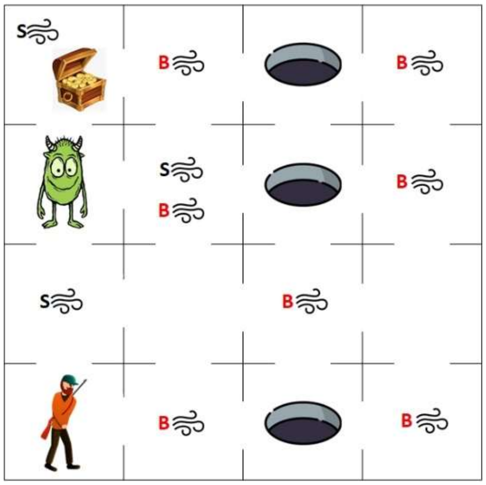
4	Win: 2 Lose: 28	Win: 0.06 Lose: 0.94	Win: 4 Lose: 2	Win: 6 Lose: 28	Win: 5 Lose: 7.36
5	Win: 13 Lose: 17	Win: 0.43 Lose: 0.57	Win: 4 Lose: 4	Win: 26 Lose: 36	Win: 9.38 Lose: 14.76
6	Win: 28 Lose: 2	Win: 0.94 Lose: 0.06	Win: 2 Lose: 8	Win: 10 Lose: 26	Win: 3.78 Lose: 17



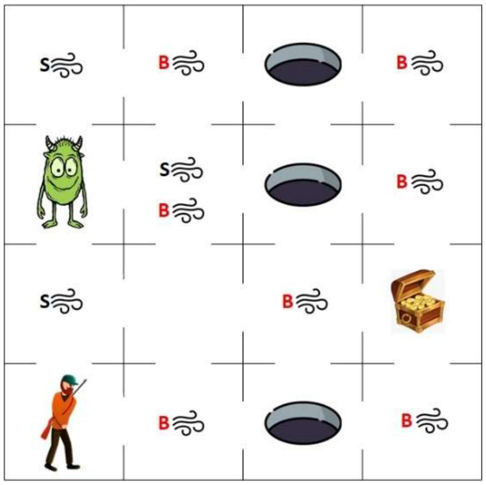
Scenario 1



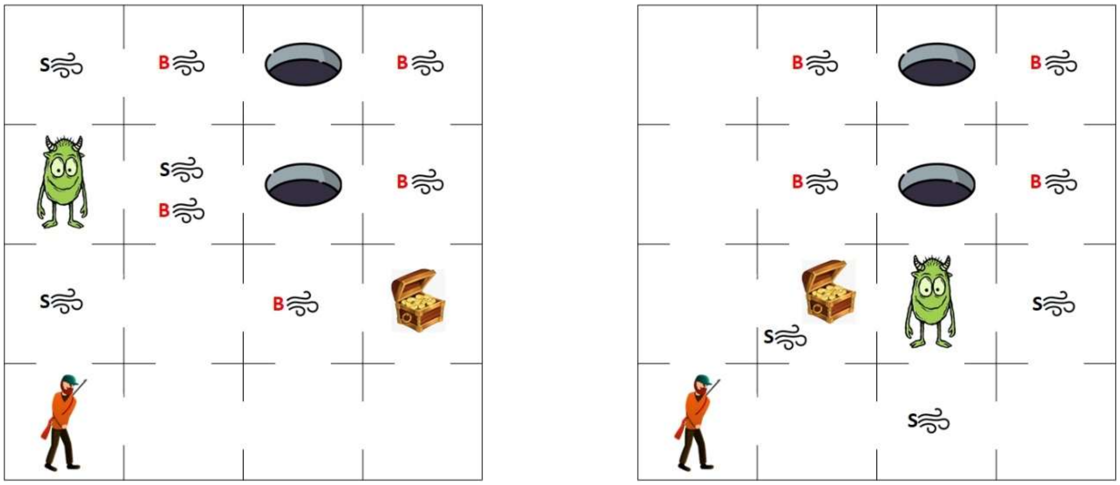
Scenario 2



Scenario 3



Scenario 4



Scenario 5 Scenario 6

Figure 8 Different scenarios of the Wumpus world

6 Conclusion and future works

The challenges faced by the VGI often lack strong mathematical and theoretical foundations, and one of its most significant challenges is the design of new game scenarios. In almost all games, players must transition from one state to another to achieve a goal; therefore, modeling a game requires accurately representing the agent's movements. To address this, we selected the Wumpus World as a case study and developed a CPN-based model to represent an agent's movement from one room to another. A key advantage of the proposed model is that, for a game with fixed assumptions, a single model can be used to design and simulate various scenarios and to evaluate their difficulty levels without modifying the underlying structure. In this study, we assumed that the agent is memoryless, forgetting the path it has previously traversed. As future work, the model could be extended to incorporate agents with memory. Similarly, while the current model assumes that the agent chooses its next move randomly, an alternative version could assign priorities to certain moves. Additional details and complexities could also be introduced to further enhance and refine the proposed model.

References

[1] R. Baltezarevic, B. Baltezarevic and V. N. Baltezarevic, "The video gaming industry: From play to revenue," *International Review,* Vols. 3-4, pp. 71-76, 2018.

[2] "Video Games - Worldwide," Statista, 2024. [Online]. Available: https://www.statista.com/outlook/dmo/digital-media/video-games/worldwide.

[3] E. Goh, O. Al-Tabbaa and Z. Khan, "Unravelling the complexity of the Video Game Industry: An integrative framework and future research directions," *Telematics and Informatics Reports,* vol. 12, p. 100100, 2023.

[4] A. Naghash Asadi, "An Approach for Modeling Trust and Distrust in Social Networks and Evaluating the Trust Level of each User using Coloured Petri Nets," *Journal of Soft Computing and Information Technology,* vol. 12, no. 3, pp. 17-29, 2023.

[5] A. Naghash Asadi, M. Abdollahi Azgomi and R. Entezari-Maleki, "Unified power and performance analysis of cloud computing infrastructure using stochastic reward netsUnified power and performance analysis of cloud computing infrastructure using stochastic reward nets," *Computer Communications,* vol. 138, pp. 67-80, 2019.

[6] A. Naghash Asadi, M. Abdollahi Azgomi and R. Entezari-Maleki, "Analytical evaluation of resource allocation algorithms and process migration methods in virtualized systems," *Sustainable Computing: Informatics and Systems,* vol. 25, pp. 1-16, 2020.

[7] A. Naghash Asadi, M. A. Azgomi and R. Entezari-Maleki, "Evaluation of the impacts of failures and resource heterogeneity on the power consumption and performance of IaaS clouds," *The Journal of Supercomputing,* vol. 75, no. 5, pp. 2837-2861, 2019.

[8] A. Naghash Asadi, M. Abdollahi Azgomi and R. Entezari-Maleki, "Model-based evaluation of the power versus performance of network routing algorithms," *Computing,* vol. 103, p. 1723–1746, 2021.

[9] A. Naghash Asadi, M. Abdollahi Azgomi and R. Entezari-Maleki, "Evaluation of the functionality of mobile wireless sensor networks using stochastic reward nets," *Scientia Iranica,* vol. 30, no. 1, pp. 91-103, 2023.

[10] R. Malathi and T. Venugopal, "The Jaina Logic Way of Wumpus World," *International Journal of Engineering Science Invention Research & Development,* vol. 2, no. 6, pp. 302-315, 2015.

[11] D. Bryce, "Wumpus World in introductory artificial intelligence," *Journal of Computing Sciences in Colleges,* vol. 27, no. 2, pp. 58-65, 2011.

[12] "Wumpus World Simulator," School of Electrical Engineering and Computer Science, Washington State University, 2011. [Online]. Available: https://eecs.wsu.edu/~holder/courses/AI/wumpus/. [Accessed 2024].

[13] T. Ferreira, E. Acar and R. S. Plaban, "Wumpus World Simulator," 2021. [Online]. Available: https://thiagodnf.github.io/wumpus-world-simulator/. [Accessed 2024].

[14] "AI | The Wumpus World Description," 27 2 2020. [Online]. Available: https://www.geeksforgeeks.org/ai-the-wumpus-world-description/.

[15] O. Diallo, J. J. Rodrigues and M. Sene, "Performances evaluation and Petri nets," in *Modeling and Simulation of Computer Networks and Systems*, Boston, Morgan Kaufmann, 2015, pp. 313-355.

[16] K. Jensen, Coloured Petri Nets; Basic Concepts, Analysis Methods and Practical Use, vol. 1, Springer, Berlin, Heidelberg, 1996.

[17] A. V. Ratzer, L. Wells, H. M. Lassen, M. Laursen, J. F. Qvortrup, M. S. Stissing, M. Westergaard, S. Christensen and K. Jensen , "CPN tools for editing, simulating, and analysing coloured Petri nets," in

Proceedings of the 24th international conference on Applications and theory of Petri nets, 2003, pp. 450-462.

[18] E. S. de Lima, B. Feijó and A. L. Furtado, "Managing the plot structure of character-based interactive narratives in games," *Entertainment Computing,* vol. 47, p. 100590, 2023.

[19] E. S. de Lima, A. L. Furtado, B. Feijó and M. A. Casanov, "Plot Composition by Mapping Situation Calculus Schemas into Petri Net Representation," in *Entertainment Computing – ICEC 2022*, 2022, pp. 61-75.

[20] M. A. Syufagi, M. Hariadi and M. H. Purnomo, "Petri Net Model for Serious Games Based on Motivation Behavior Classification," *International Journal of Computer Games Technology,* vol. 2013, no. 2, pp. 1-12, 2013.

[21] V. A. Ziparo, L. Locchi, P. U. Lima, D. Nardi and P. F. Palamar, "Petri Net Plans," *Autonomous Agents and Multi-Agent Systems,* vol. 23, pp. 344-383, 2011.

[22] A. Vyas, A. Parnami and M. R. Prusty, "Graph theory-based mathematical modeling and analysis to predict a football dream team," *Knowledge and Information Systems,* vol. 65, pp. 1523-1547, 2023.

[23] L. Shi, Y. Chen, J. Lin, X. Chen and G. Dai, "A black-box model for predicting difficulty of word puzzle games: a case study of Wordle," *Knowledge and Information Systems,* vol. 66, pp. 1729-1750, 2024.

[24] F. M. Barreto, J. C. Jeske de Freitas and S. Julia , "A Timed Petri Net Model to Specify Scenarios of Video Games," in *Information Technology - New Generations*, Cham, 2018, pp. 467-473.

[25] Y.-S. Lee and S.-B. Cho, "Context-Aware Petri Net for Dynamic Procedural Content Generation in Role-Playing Game," *IEEE Computational Intelligence Magazine,* vol. 6, no. 2, pp. 16-25, 2011.

[26] M. Muratet, T. Carron and A. Yessad, "How to assist designers to model learning games with Petri nets?," in *the 17th International Conference on the Foundations of Digital Games*, New York, NY, USA, 2022, pp. 1-10.

[27] K. Sinclair and D. Livingstone, "Accurate Complex Systems Design: Integrating Serious Games with Petri Nets," *International Journal of Serious Games,* vol. 3, no. 1, pp. 19-31, 2016.

[28] H. Byeon, C. Thingom, I. Keshta, M. Soni, S. A. Hannan and H. Surbakti, "A logic Petri net model for dynamic multi-agent game decision-making," *Decision Analytics Journal,* vol. 9, p. 100320, 2023.