4th Asia Pacific Conference on Contemporary Research (APCCR- 2018),
Bali, Indonesia
ISBN :978-0-6481172-9-2
Asia Pacific Institute of Advanced Research (APIAR)
www.apiar.org.au

RE-DESIGNING THE PACMAN GAME USING PUSH DOWN AUTOMATA

Ariana Yunita[a], Riestiya Zain Fadillah[b], Muhammad Redho Darmawan[c],
Andika Dwi Gutomo Putra[d],
[a,b]Universitas Pertamina, Jakarta, Indonesia
Corresponding Email: ariana.yunita@universitaspertamina.ac.id

Abstract

Pacman is a famous and a classic game that a player should eat dots in a maze to increase points and at the same time a pacman should avoid ghosts. Furthermore, the pacman may eat a special dot that can convert ghosts into items that can be eaten. This study purposes to redesign the game using push down automata as an alternative for business model to replace software engineering principle. By using automata, a game can be designed and developed. Push down automata, a finite state machine with stack, basically can store the inputs, so the system can memorize. In this game, push down automata stores dots to identify whether ghosts can be eaten or should be shunned. The pacman game is a desktop-based game and developed using Godot game engine. After redesigning and developing, a blackbox testing was conducted and it results that all inputs produce the same outputs as drawn in push down automata.

Keywords: Game design, Push Down Automata, Pacman.

1. Introduction

Computer games, which are usually associated with children andenjoyment, are varied and attracted persons of all ages as well as those of various backgrounds. Nowadays, the purposes of computer games are not only for entertaining, but also for learning (Yunita et al, 2017), health (Smeddinck, 2016; Wattanasoontorn et.al, 2013), simulation (Eisenack & Reckien, 2013), military (Djaoutiet al, 2011),and supporting any other activities via entertainment. Those kinds of games are commonly called serious games. Eventhough various kind of games have been developed and published, several classic games still attract persons to play, for example, Chess, Solitaire, Pacman and Mario Bross.

Designing a game is one of the essential steps in developing a game which involves multidisciplinary studies, such as cognitive psychologies, computer science, graphic design, creative writing, and any other field of studies. It is needed to view different aspects to understand what a game is (Koster, 2013). In this paper, we present how to design a game using automata theory. The case study for this research is a classic and well-known game,Pacman.The next section describes the literature review, methodology, results, conclusion and future works.

2. Literature

2.1 Related Works

Computer science has two compulsory components:1) computing models and basic computation, 2) system design engineering. Automata-and-computational theory, contributed by Alan Turing, ispart of the computing model and basic computation. Therefore, it becomes one of the novel theory that leads to the development of computer science (Hopcroft,

2013).Interestingly, some researchers have explored how to use automata theory in the game design and game industries have applied the concept to design a game (Jamil et al, 2016; Qureshi et al, 2012; Raj, 2017; Salvador-Ullauri et al, 2016).

Jamil et al (2016) developed a game called Infinite Runner Birds using mealy machine,andQureshi et al(2012) designed a roller coaster game using Non-Deterministic Finite Automata (NFA). Those references show that fewer bugs are found as compared to any other game design strategy. In addition, it is recorded that using automata theory eliminates the need of Case Diagrams and software engineering principles. Raj (2017) also design a game using NFA, i.e. Snakes and Stairs.It is shown that designing a game by utilizing NFA provides a quick method and easy reference to solve it. Furthermore, Salvador-Ullauri et al (2016)developed a serious game for supporting the teaching process. Those studies, however, use finite state automata (FSA)for a game design which is modeling the game without memory. In this paper,we proposeanother type of automata, Pushdown Automata (PDA), that is well known as FSA with stacks.

2.2 The Concept of Automata

Automata is a theory that studies the abstractions of computing instruments. There are four models or types of abstract machine, which are: finite-state, pushdown, linear bounded and Turing machine (Hopcroft, 2013).The finite-state machine is the lowest machine among others. The differences of each machine are described in the following table.

Table 1.Specification of Each Model/Type Automata

Model/Type	Input tape	Head Direction	Memory
Finite State	Read Only	One way	-
Pushdown	Read Only	One way	Stack
Linear - Bounded	R / W	Two-way direction	(bounded)
Turing Machine	R / W	Two-way direction	(unbounded)

FSA is a machine which is used to recognize a formal language called regular language. Finite automaton consists of finite states and, hence, FSA is often called Finite State Machine (FSM).Finite State Machine has properties as follows:

- The behaviour of the machine depends on the input/output tape received by the machine.
- At any time, the machine can be in a certain state and can move from that state to another state by the change in input.
- Discrete input in automata can be regarded as a language that should be recognized by automata. After the automata finishes reading the input, the automata will then make a "decision".

In this discussion, the type of automata that will be used is PDA. It is an abstract machine which has similarity with a finite automaton, but it has the ability to access unlimited memory in form of a stack (Hopcroft, 2013).

PDA consist of six components:

a. a read-only input tape,
b. input alphabet,
c. a finite state control with two head, one read only and the other one read/write,
d. a finite set of final state,
e. an initial state,

f. a stack called Pushdown Store (PDS).

In PDA, Finite State Control (FSC) has a task to read every single symbol that came with the input. PDS Stack can be operated with Last In First Out (LIFO) concept, there are two basic operations: push (insert element to the top of a stack) and pop (remove the top element of a stack). The transition between states in PDA has similarity with the transition from finite automata. Basically, PDA activity consists of reading input symbol from input tape by FSC and reading the top element of PDS stack to push or pop. The PDA model is illustrated in Figure 1.Formal definition or graphical notation for *PDA* M= (Q, Σ, Γ, δ, q_o, Z_o, F) are:

Q: finite set of state,

Σ: finite set of input symbol,

Γ: set of pushdown symbol (which can be pushed/popped),

δ: transition function,

q_o*: initial state,*

Z_o*: first element that inserted to the stack,*

F: finite set of final state.

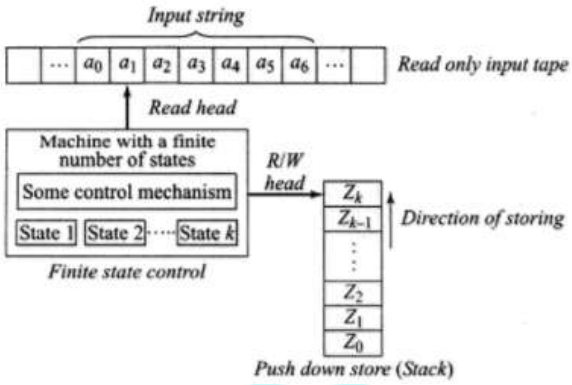

Figure 1**:** Model of *PDA* (Hopcroft, 2013)

1.3 Pacman Game

Pacman is a game board witha player-character, referred to as Pac in this paper, moves inside a maze havingfull of dots. Several objects in Pacman game are Pac, dots, ghosts, and, in special cases, bonus food. The rule in the game is Pacshould eat all dots scattered throughout the maze while avoiding the ghosts. There is a big dot whichhas a bigger size than normal dots. When Pac eats the big dots, it can eat ghosts and changes the ghosts' movement pattern that is avoiding it. The player will win a level when all dots have been eaten by the Pac. PDAis chosen in the development of Pacman because the system could memorize all dots that have been eaten by the Pac. On the other hand, it will be difficult to determine whether or not the player has already won the game if any automata-without-memory is used.

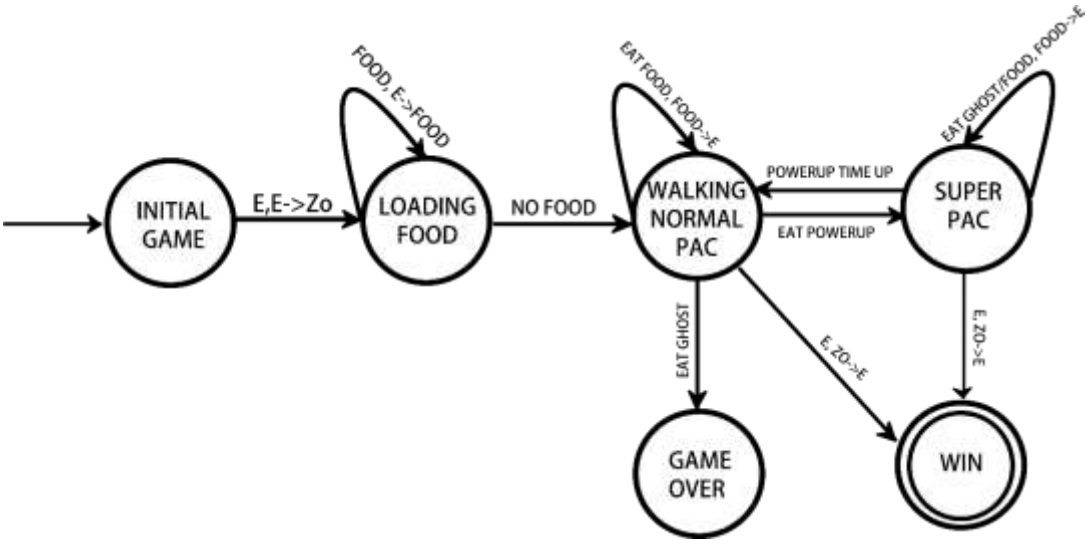

Figure 2:*PDA* for Pacman, User/Pac's Point of View

2. Design and Methodology

Figure 2 shows PDA for Pacman from the point of view of the player, i.e. the Pac. There are 6 states in the Pacman design with PDA as follow.

1. Initial Game
   This is the beginning of the game, including assets loading (maze, ghost, and default score = 0).
2. Loading Food
   This is the state where all dots are placed inside the maze, i.e., each level could have different shapes of the maze, thus the dots number could increase or decrease.
3. Walking Normal Pac
   This is the state where Pacenters the maze and eats the dots.
4. Super Pac
   This state occurs after Pac eats the big dots and hence it can eat the ghosts.
5. Game Over
   This is the state where Pac is in the Normal state after being eaten by the ghost.
6. Win
   This is the state where all dots are successfully eaten by Pac.

3. Results

We design the Pacman by utilizingPDA.The game is then developed using Godot game engine that is shown in Figure 3.

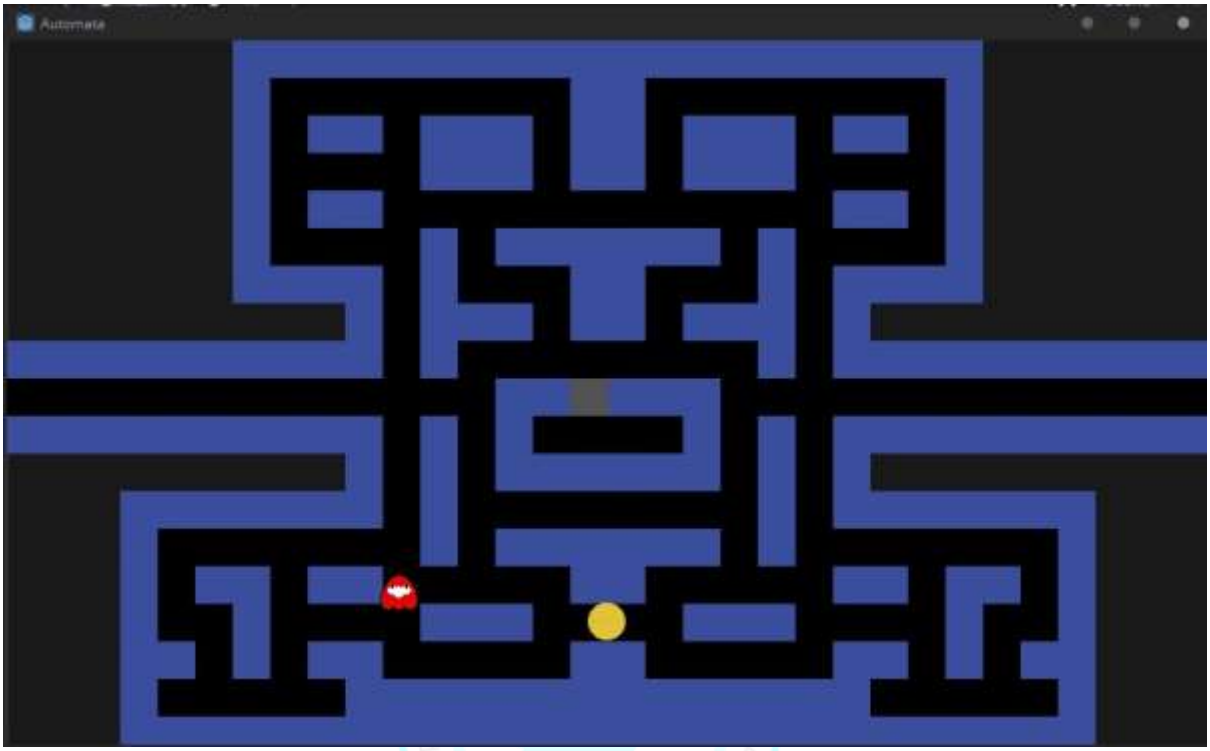

Figure 3: User Interface of Pacman Game

Based on the development phase, we found that designing the automata prior to developing simplifies the shifting between one state to another and providessimpleinputs and outputs. Furthermore, It leads to a more systematic and organized code for the game. In addition, the each game class inside the code represents states in the PDA design.

Afterward, blackbox testing is conducted to check whether or not inputs from users are the same as expected conditions. Below isthe table that contains the results of blackbox testing.

Table2. The results of blackbox testing

No	Input	Expected Conditions/Output according to PDA	Does the expected output the same as output in the application?
1	No inputs	Ghost eats Pac	Yes
2	Pac moves right/left/up/down and eats dots	Point increases	Yes
3	Pac moves right/left/up/down and eaten by ghost	Game over	Yes
4	Pac eats special food	Pac can eat ghost	Yes
5	Pac eats ghosts	Point increases	Yes

Conclusion

PDA has been proposed to design the Pacman game by utilizing a finite-state machine with a stack to store the inputs such that the system can memorize. It is shown that there is consistency between the inputs and the expected outputs. Furthermore, we found that designing a game using Automata is the best practice by which a game director without having any background in programming can design a game with visualization. Therefore, it is a helping tool to explain the story and business process of a game. Moreover, PDA helps the coding process to be more systematic and organized. In further work, the game design from the ghost's point of view should be conducted. Also, comparing a game design between utilizing Use Case Diagram and Automata Theory is a very interesting work.

References

i. Djaouti, D., Alvarez, J., Jessel, J.P. and Rampnoux, O., 2011. *Origins of serious games. In Serious games and edutainment applications* (pp. 25-43). London: Springer.

ii. Eisenack, K. and Reckien, D., 2013. Climate change and simulation/gaming.

iii. Hopcroft, J.E., 2013. *Introduction to Automata Theory, Languages and Computation: For VTU*. 3rd ed. S.l.: Pearson Education India.

iv. Jamil, A., Ullah, A. and Rehman, M., 2016. An Infinite Runner Game Design using Automata Theory. *International Journal of Computer Science and Software Engineering*. Vol. 5, no. 7, p.119.

v. Koster, R., 2013. *Theory of fun for game design*. S.l.:" O'Reilly Media, Inc.".

vi. Qureshi, N.S., Abbas, Z., Sohaib, M., Arshad, M., Sabir, R.A. and Maqsood, A., 2012. A Roller Coaster Game Design using Automata Theory. *International Journal Of Multidisciplinary Sciences And Engineering,*vol.3, no. 5, pp.40-45.

vii. Qureshi, N.S., Mushtaq, H., Aslam, M.S., Ahsan, M., Ali, M. and Atta, M.A., 2012. Computing game design with automata theory. *International Journal of Multidisciplinary Sciences and Engineering*, vol. 3, no. 5, p.13021.

viii. Salvador-Ullauri, L., Luján-Mora, S. and Acosta-Vargas, P., 2016. Development of serious games using automata theory as support in teaching people with cognitive disabilities. *In Proceedings of ICERI2016: the International Conference of Education, Research and Innovation Conference, 14th–16th of November 2016,* p. 4508.

ix. Smeddinck, J.D., 2016. *Games for Health. In Entertainment Computing and Serious Games, pp. 212-264*. Springer, Cham.

x. Wattanasoontorn, V., Boada, I., García, R. and Sbert, M., 2013. Serious games for health. *Entertainment Computing,*vol.4, no. 4, pp.231-247.

xi. Yunita, A., Moore, A. and Losada, J.A.G., 2017, October. Gamification for learning science: ELISA (Enzyme linked immuno sorbent assay) game study case. *In Information & Communication Technology and System (ICTS), 2017 11th International Conference on (pp. 297-302)*. IEEE.