Original Article

Expert System For Improved Response In Greenfoot Game To The Machine Learning Implementation

Mohamad Nurkamal Fauzan[1], Nyi Mekar Saptarini[2*]

[1] Applied Bachelor Program of Informatics Engineering, Pos Indonesia Polytechnic, Bandung, Indonesia
[2*]Department of Pharmaceutical Analysis and Medicinal Chemistry, Faculty of Pharmacy, Universitas Padjadjaran, Jatinangor, Indonesia. nyi.mekar@unpad.ac.id
DOI: 10.47750/pnr.2022.13.04.278

Abstract

The game is the first semester material in the Algorithm course at the Pos Indonesia Polytechnic which was adopted from Oracle Academy. The output of this course is a simple 2D Greenfoot game. In 2020, around 70% of students made games, which move randomly. It is due to the computer responses that are not like human responses when the game is running and the rest uses a bouncing technique, so the computer response is very predictable. This study applied an expert system in the form of Naive Bayes and forward chaining with training data from student correspondence for the computer movements. The purpose of this study was to compare the random, bounce, and expert system techniques. This study contributes to providing concrete innovations in the form of course implementation in semester 1 (using Greenfoot) and semester 6 (using WEKA as a learning tool in artificial intelligence, data science, expert system courses) so that both are integrated. This can be a reference for teachers who use the Oracle Academy (Greenfoot) curriculum to be able to implement other machine learning/expert system models.

Keywords: Expert system, Forward chaining, Greenfoot, Naïve Bayes, Response.

INTRODUCTION

The ability of the hardware to perform computing is increasing. This makes it possible to embed Artificial Intelligence (AI) into a wider range of devices. In the past, a company made telephone system automation for example to contact the complaint service press 1, for language translation press 2, etc. Now, with the implementation of AI, the telephone system can use the Natural Language Processing method so that it is as if interacting with humans as well as short message services. AI through machine learning makes the system smarter over time because it can recognize and create its patterns. In the case of a refrigerator, a refrigerator with a camera and AI can recognize the type of food, estimate the stock content of a type of food and make recommendations. An autonomous vehicle (AV) or a self-driving car is a vehicle that is capable of sensing its environment with several sensors, such as radar, lidar, GPS, sonar, and inertial measurement units and moving safely with little or no human input, results from the sensor readings processed by AI then used for navigation and response to the environment. In the field of chemistry, with the current machine learning-based, the deep learning model is employed to capture the constitution of 541,555 bioactive small molecules retrieved from ChEMBL [1]. Android smartphone devices provide Google Assistant, which can be the personal assistant. Examples of AI in games such as The Last of Us, F.E.A.R, Tom Clancy's Splinter Cell: Blacklist, etc.

The industry is trying to implement and benefit from AI, and the gaming industry is no exception. As reported, the video game industry will grow to $ 300 billion by 2025. Mobile and cloud games are expected to grow, as mobile games outnumber PC and console revenues in 2018, which reached $ 130 billion [2]. There is a list of video game genres such as action, adventure, role-playing, etc. In the action type, there are several sub-platforms, such as platform, shooter, fighting, etc. Examples of platformer games like Mario Bros, Sonic the hedgehog, etc. According to the author, the game Mario Bros, which was popular around the 90s, is a legendary platform type game, in this game there are several levels, challenging obstacles, players can jump, run and shoot. Until now, the

game industry continues to produce platform types using the latest technological capabilities, for example on the PS4 console there is Sonic Mania which was developed by Sega or can be created a platformer game with the Unity3D engine which can use Key AI features which provide a machine learning agents toolkit for game maker. To make a game requires knowledge, skills, art, and programming. In general, the stages of making a game start from an idea then create a storyboard, scenario, and finally build a game. When creating a game, assets in the form of images, sound effects, and background music must be considered. Assets can be created using free and open-source software, such as LMMS and GIMP. LMMS is used to create sound assets. With LMMS, can easily generate, play and arrange sounds and tones into complete tracks. Sound effects or music can be saved in mp3 or Wav format. GIMP is a cross-platform image editor, which can create image assets with the output of Portable Network Graphics (png) for images that require transparency or jpg for compressed images. Programming is needed to create a set of instruction sets or algorithms that are understood by machines so that the machine can do its job. There are various programming languages such as Java, Kotlin, C #, Python, PHP, C ++, JavaScript, Golang, Objective-C, etc. With programming, can also create AI and its derivatives such as machine learning, expert systems, etc.

The game is one of the Algorithm course materials adopted from Oracle Academy [3], with a simple two-dimensional game as an output. The Oracle Academy curriculum contains Java fundamental material. Java fundamentals are divided into three main subs, i.e. (a) Alice 3, introducing object-oriented programming with drag and drop techniques, students getting to know classes, making the object, methods in the form of procedures and functions, and making interactions; (b) Greenfoot, students begin to be introduced to writing Java programs with the ultimate goal of making games, in this case, students can create their classes, create scenarios and simple algorithms to adjust the target game with the Greenfoot framework; (c) Java, students create Java programs then run them in the terminal with a focus on using the Eclipse IDE for integrated debugging and compilation [3].

In Greenfoot, students write in a Java program so which requires following the program syntax writing pattern. Students also begin to understand the concepts and basics of programming, so that when creating solutions or algorithms, errors are often encountered but this is a common thing in learning. Greenfoot IDE also helps students where the error lies or also provides estimates. Likewise, the available materials and API documentation from Greenfoot are sufficient. So with the final project of making a game with Greenfoot, the target is that students can design scenarios then write their programs, logic, algorithms until a game can be played. So making a game with Greenfoot is a complete package for the final project.

The Greenfoot game is the final project of the 2020 semester 1 algorithm course at the Pos Indonesia Polytechnic. Previous data showed that out of 60 students, who make games on average implement about 70% of randomly moving enemies or use invisible objects, 10% of the enemy can bounce to the right/left, or 20% of the enemy can see the edge of the platformer object on the platformer type, so the game very easy to predict and win over. So the problem of this game is how to make the Greenfoot game more difficult and the computer has a human-like response. This is the target of algorithm courses such as teaching Object-Oriented Programming and simple program logic, then the implementation of expert systems will be tested, so that computer responses can resemble humans with algorithm implementation [4,5], such as expert systems [6,7]. A similar study includes incorporating AI into the game [8] and the use of WEKA's Multi-Layer Perceptron, J48, and Random Forest classifiers to create game-level tiers [9]. This study innovation and solution is to align algorithm courses in semester 1 and expert systems in semester 6 using WEKA tools because the same programming language is Java. Real case studies, implementation of the game are expected to attract more interest, innovative exploration of Greenfoot and WEKA. The objective of this study was to compare three scenarios, i.e. random response, platformer response using object invisibility, and response using expert system algorithm, and the result was the chance of winning from the three scenarios. The implementation of an expert system in games is not the first time but, this is the first study to discuss an expert system in Greenfoot throughout the literacy study carried out until the mid of 2021.

Materials and Methods

Materials

Greenfoot is an Integrated Development Environment (IDE) with the Java or Stride language, which is designed for educational purposes. Greenfoot allows easy development of two-dimensional graphics applications, such as simulations and interactive games. Greenfoot was developed by King's College London, with Oracle support. This is free software, released under the GPL license. Greenfoot is available for Windows, macOS, Linux, Solaris, and

the latest JVM [10]. This study uses Java as a programming language. Java is a programming language and computing platform first released by Sun Microsystems in 1995. Java is a fast, secure, and reliable language for laptops, data centers, game consoles, supercomputers, cell phones, and the internet [11].

Greenfoot game is designed for algorithm implementation without high computational resources, with the consideration that the game rating is easy, feasible, and the computer responds to players like humans. Wang tested the prediction of the Naive Bayes classifier from the respective winning probability of the two parties in the DOTA2 game [12]. The results showed that the Naive Bayes classifier is a practical tool for analyzing lineups and predicting outcomes based on player choices. Anwar used the Naive Bayes classifiers to classify players based on data from the EEG headset and gave an accuracy value of 88% [13]. Al-Bow *et al*. concluded that Greenfoot successfully taught basic programming concepts to novice programmers [14]. Fuzzy logic systems are applied to a serious game strategy to each consumer [15] and this technique is also applied to other systems with the application of fuzzy logic Mamdani [16]. This study used Naive Bayes and forward chaining to move the enemy, because of the results of other studies and relatively easy to program. For examples of games that have been created using Greenfoot, students can access the site greenfoot.org or Github with the keyword search for Greenfoot. GitHub is a web-based software, which used as a collaborative medium in developing application software projects. The software of the engineering collaborative learning model uses Github for informatics students because it can simplify the learning and collaboration process [17].

Methods

In this study, of the 60 students, only 20% make games with the platformer type but the logic of the game is not too simple when compared to shooter-type games or games that run randomly. In this type of platformer, it should be checked for collisions on the platform/floor object when the player jumps, falls, and walks, as well as the enemy object that becomes the obstacle. The Greenfoot Game was designed to contain 1 world and 3 actors. This is the minimum number of classes required in a platformer game so that the enemy and player classes will only be made into one object, while the floor objects of the Ground class are three. The world contained the MyWorld class which defines a 900x600 pixel screen. MyWorld class was made with 3 floor platforms for players or enemies, the positions were on a scale of 1: 100, i.e. (x1): 1 to 9 (y1): 6; (x2): 2 to 4 (y2): 4; and (x3): 6 to 9 (y3): 3. Actions can be taken by players and bots were moving left/right, up/dodging, and shooting, with 10 lives. Players and bots are always on the floor and as a guideline, the world map shows the position of the region of MyWorld which by default is not displayed visually to end-users **(Figure 1)**.

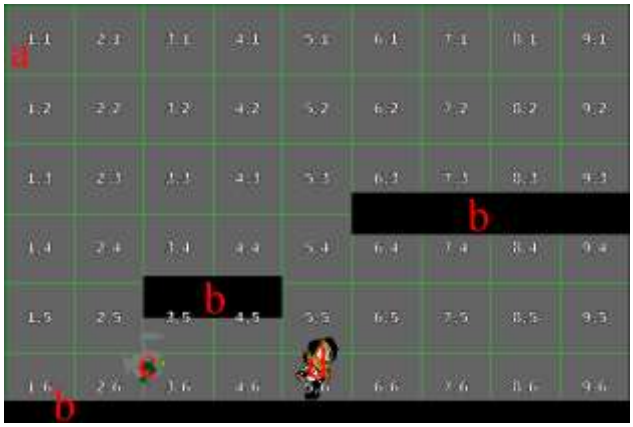

Figure 1. World map, a=x, y position, b=floor, c=bot enemy, d=player

The base class was a gaming platform. The trial was carried out with three scenarios, i.e. random responses, platformer responses using invisible objects, and responses using expert system algorithms.

In enemy bot random response scenarios, with pseudocode:

predAction = Greenfoot.getRandomNumber(3)
If predAction = 0
a = true
If predAction = 1
d = true

```
If predAction = 2
space = true
```
Greenfoot.getRandomNumber(3) will return a value of 0, 1, 2.
In the response scenario the bot moves to the left/right bounce until it hits an invisible object, with pseudocode:
```
If dir = 0
    d = true
Else
    a = true
If getOneObjectAtOffset(0,0,Kanan.class) != null
    dir = 0
If getOneObjectAtOffset(0,0,Kiri.class)!=null
    dir = 1
```

In the invisible object pseudocode built from the Kiri and Kanan classes. The getOneObjectAtOffset function will check the object's value at the specified position from the bot side.
In an expert system implementation scenario, the PlayerBot class responds to keyboard input when the main player moves. The projectile class was created to create bullets in MyWorld. The enemy class contains a bot or Non-Player Character (NPC) [18]. This bot is an implementation of an expert system, i.e. Naive Bayes and Forward Chaining. Naive Bayes is an algorithm with good performance and fast learning in a variety of supervised classification problems [19] and the successful implementation of game strategies [18]. Enemy movements were left, right, and jump using Naive Bayes, while dodging and shooting using forward chaining.
Naive Bayes classification is a classification technique of calculating the degree of fit. If the degree of fit is high, then the data are classified into the corresponding class. If the Vnj classification has attributes a1, a2, ... an, then the result of Vnj can be calculated by Eq. (1) [20].

$$V_{nj} = P(V_j) \prod_{i=1}^{n} P(a_j \vee V_j) \quad (1)$$

According to Eq. (1), each attribute will produce a conditional probability value for each attribute, so that it can be described in Eq. (2) [20].

$$V_{nj} = P(V_1)(P(a_1 | V_j) * P(a_1 | V_j) * \dots \dots .* P(a_n | V_j) \quad (2)$$

V_{nj} = the value of the probability classification result of the j target class
$P(V_j)$ = j chance of occurrence of target class v
$P(a_i \mid V_j)$ = the conditional probability of occurrence of attribute a_i in class v_j
The modeling stages, ground truth data, and the game screen were mapped into a 2x2 matrix with 9x6 dimensions. This is obtained at a scale of 1: 100 pixels from the 900x600 screen. The scale mapping technique was calculated by Eq. (3).

$$meX = \frac{getX}{100} + 1 \quad (3)$$

The getX function returns the x coordinate of the actor's current location with an integer data type. Four attributes were mapped, i.e. enemy x position, enemy y position, player x position, player y position, and class classification. The model has three class choices, i.e. a, d, and space with 2380 training data sets. Class a for bot moving left. Class d for bot moving right. Class space for bot jumping or dodging.The data set was inputted manually for this scenario, then processed through the WEKA application [21]. Attributes x, y are nominal types 1-9, target_x and target_y attributes were nominal types 1-6. Training data were obtained from four pairs of human players for 11 minutes by recording each position state between the player and the enemy **(Table 1)**. The enemy was moved manually by humans, so in the training process, there were 2 players, 1 main player, and 1 enemy.

Table 1. Collection of Data Training

Team player	Collect data	Time (min)
1	680	10
2	451	10
3	615	15
4	634	10

average	595	11 min 15 sec

The test data collection was done only once, with the results of 142 data in 5 minutes. The training was carried out in parallel and file contributions using the BufferedWriter class with output in CSV format. Human labeling is done when there is a change in the position of a player or enemy bot with the pseudocode of recording.

The test data collection was done only once, with the results of 142 data in 5 minutes. The training was carried out in parallel and file contributions using the BufferedWriter class with output in CSV format. Human labeling is done when there is a change in the position of a player or enemy bot with the pseudocode of recording.

If tempPosition != position
tempPosition = position
Writetocsv

This technique is like the Matlab ground truth Labeler [22], which provides for human intervention. Example data: 1,6,1,6, a. The results of Naive Bayes modeling will be saved into a serial file for the next classification process. Forward chaining was a fact to get a conclusion from the previous facts. This reasoning was based on existing facts (data-driven). Tracking started from inputting information, then tried to make conclusions. The facts were obtained from the input x, target x, and the x position of the projectile object. The rules were made from logic to the x and y target positions of the enemy. Rule 1 read the facts from the target position y with the output shoot (s) of boolean type. Rule 2 read the facts, that the target x position was distanced from the enemy with a boolean type dodge output.

The Forward chaining as follows:
Fact 1: this.getPlayerbot().getY() lower than this.getY() + 20
Fact 2: this.getplayerbot().getY() greater than this.getY() - 50
Fact 3: this.getObjectInRange(150, Projectile.class)
Decision 1: this.s is true
Decision 2: this.space is true
Rule1: Fact 1 AND Fact 2 => Decision 1
Rule2: Fact3 => Decision 2

Pseudocode:
predAction = classify(meX,meY,playerBotX,playerBotY)
if predAction = 0
a = true
if predAction = 1
d = true
if predAction = 2
space = true

The classify function will process input in the form of human player positions and bot enemies at x, y and process it through a previously created model. This will return the classification result value. Other supporting sources are available from GitHub. The examination was an experiment on respondents. The student survey was conducted with a questionnaire containing total play and total winners against three-game scenarios. The scenarios were played by students from scenarios 1, 2, and 3, then the results were compared between the three scenarios. The questionnaire target was 80 students.

To achieve the Greenfoot game more difficult and the computer has a human-like response, the Naive Bayes classification method and Forward chaining were applied. Naive Bayes modeling was obtained by collecting player responses when playing games, in this case, the bot actor was played by the player/student and the Player actor was played by other players/students. The collected data were entered into training data and test data. The Naive Bayes model that formed, then stored to respond to the position of the bot actor against the Player actor. The Forward chaining method contains a fact set of rules designed to attack Player actors. These two methods are then implemented in the game on the bot actor. For testing the three scenarios, the player must play the game with each scenario. Players then fill out a questionnaire. The questionnaire was then interpreted. The software requirements for testing are Greenfoot 3.6.0 software onwards and the recent Java platform on Intel/AMD processors requires a 64-bit CPU and 64-bit operating system.

Results and Discussion

Based on the research design that has been carried out, scenario 1 will be explained and discussed what was found. Enemy bots move randomly, for example, when the game is running, the bot on the 1st-floor moves to the right, jumps to the 2nd floor, advances to the left jumps and shoots while the player does not take action on the computer via the keyboard, players can easily attack enemy bots by moving left, locating and shooting.

In the scenario 2, the enemy bot will move to the right and when it touches an object from the class kiri (ki) then the direction of motion of the enemy bot moves to the left until it touches object kanan (ka) and so on, this causes the movement of predictable enemy bots, although coupled with random action to jump and shoot, so that enemy bots are easy to defeat.

Figure 2 showed the position of the mapped area, the value information of the variable can also be displayed during the game development process for debugging purposes. The information about the position of enemy bots, players, and motion classifications of enemy bots according to Naive Bayes and forward chaining modeling. The enemy bot moves towards the player. In the scanario 3, Naive Bayes and forward chaining are used in this study to improve the response of the enemy to players, according to the author, this classification probability method is easy to implement with a small number of classes and can be trained with a small dataset [23], as well as forward chaining, namely simple rule application. This study measurement only involved 1 enemy and 1 player.

By default info regarding the position and value of the variable is not displayed, when the game is being played by the end-user. When the player is on the 2nd floor, the enemy bot will chase to the same position and at a certain distance the enemy bot will shoot. Enemy bot movement was the result of modeling, this makes it difficult for players to attack or dodge. For example, in **Figure 2**, Player and Bot actors were at positions x=7 and y=6. **Figure 3** showed the performance of the Naive Bayes model.

Figure 2. Implementation of Naive Bayes and Forward Chaining, info shown

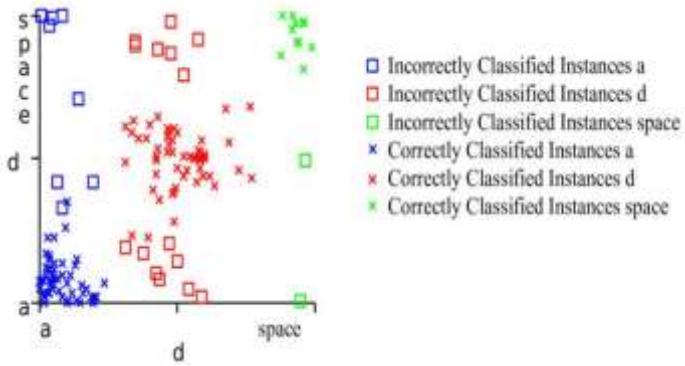

Figure 3. Class vs predicted class

In the expert system implementation scenario, the modeling process from training data and test data has previously

been carried out. Table 2 showed that of correctly classified and incorrectly classified as the result of performance summary of Naive Bayes modeling. Models are stored in serial files for use by the game. In the expert system implementation scenario, the modeling process from training data and test data has previously been carried out. Models are stored in serial files for use by the game. There were 2380 data train sets, 142 data test sets and the results of the confusion matrix **(Table 2)** showed that of correctly classified instances were 117 and incorrectly classified instances was 25. So, the level of accuracy was 82.3944% when implemented in the game, then the enemy can jump on ground 4 and 3.

Table 2. Confusion Matrix

		True class		
		a	**d**	**space**
Predicted class	a	51	4	4
	d	8	55	7
	space	1	1	11

Table 2 showed the confusion matrix. The results of Forward chaining with the 2 rules applied to the shoot and dodge method produce an accurate response to the dodge or projectile motion. As result, if the value of s is true, then the enemy will shoot and if the space is true then the enemy will jump. Questionnaire results of 80 students from three scenarios, scenario 1 random, scenario 2 bounce, scenario 3 implementation of the Naive Bayes expert system and forward chaining **(Table 3). Table 3** was the results of the implementation of the test with the three scenarios. Variable Main to accommodate the number of times played, variable Menang to accommodate the number of times won the game. For example in the Random scenario, the main Random shows the number of times the player plays in the Bot actor scenario moving randomly, menang Random shows the number of times the player can win the game by defeating the Bot actor who acts randomly.

Table 3. Main and Menang Results for Random, Bounce, and Nbfc Scenarios

	main Random	**menang Random**	**main Bouncing**	**menang Bouncing**	**main Nbfc**	**Menang Nbfc**
Total	764	668	745	647	765	18
Average	9.55	8.35	9.3125	8.0875	9.5625	0.225

After implementing the expert system, the game was 37 times more difficult to win than the random technique **(Table 4)**. According to our interpretation, the expert system method is superior to the random method, this is because the modeling techniques and class labels were recorded when the Bot actor is played by the player/human/student-run well. When players play Bot actors to train, this is done seriously by student respondents. Game response resembles a human response. Expert System has improved better game response in a Greenfoot game.

Table 4. Comparison of Menang Probability between Scenarios

scenario	**1**	**2**	**3**
1	1	1.0068	37.1597
2	0.9933	1	36.9094
3	0.0269	0.0271	1

As additional discussion material, if there is more than one enemy, it should be noticed how the enemy object responds to one another because in a certain condition/position the enemy will run uniformly, this requires other techniques so that the enemy can respond to enemies other than the main player. This is also a concern about how the power in computing if the Enemy/Bot object is more than 200 in one world on the Greenfoot, therefore a profiling program is needed to measure the space (memory) or time complexity of a program, the usage of particular instructions, or the frequency and duration of function calls so that it can be analyzed further to find the best optimization method. Comparison of a screen scale of 1: 100 to training data can be more precise, for example, 1: 1000, then Classification on Naive Bayes requires work to define/label more classes so that suggestions for further development can be implemented by the Generative Adversarial Network (GAN) method [24] as done by NVIDIA Researchers on Pac-man games. The Grenfoot game engine can be further developed so that the rendering results become smoother and can be performed on android mobile. For the general case in Greenfoot, if ported to low-end android mobile, scenarios that have many actors (eg more than 30) will cause a decrease in rendering and computing performance. This study has succeeded in producing examples of integration of semester 1 courses that focus on the introduction to programming through the game Greenfoot (Oracle curriculum) and semester 6 courses on expert systems.

Conclusion

Naive Bayes modeling produces an accuracy rate of 82.3944% from 2380 training data and 142 test data so that the game response resembles a human response. The implementation of an expert system made the Greenfoot game was 37 times more difficult to win than the random technique. **Figure 1** as previously explained, Greenfoot is used by the Oracle curriculum for beginners in learning programming. This study contributes to providing concrete innovations in course implementation in semester 1 (using Greenfoot, algorithm 1 course) and semester 6 (using WEKA as a learning tool in machine learning/data science/expert system courses) so that both are integrated. This can be a reference for teachers who use the Oracle curriculum, especially those who use Greenfoot to be able to implement other machine learning/expert system models. This study has taken the example of Greenfoot in the form of games, in other cases, it can also be implemented in the form of learning simulations with various disciplines.

Acknowledgments: The authors thank all the students who participate in this study.
Conflict of interest: None
Financial support: None
Ethical statements: None.

REFERENCES

1. Merk D, Grisoni F, Friedrich L, Schneider G. Tuning artificial intelligence on the de novo design of natural-product-inspired retinoid X receptor modulators. Commun Chemistry [Internet]. 2018 [cited 2021 Jun 12];1(68):1-9, Available from: https://www.nature.com/articles/s42004-018-0068-1 DOI: 10.1038/s42004-018-0068-1
2. Lanier L. Video Games Could Be a $300 Billion Industry by 2025 (Report) [Internet]. 2019 [cited 2021 Jun 9]. Available from: https://variety.com/2019/gaming/news/video-games-300-billion-industry-2025-report-1203202672/
3. Gooris D. Oracle Academy, Take The Future In Your Own Hands. INTED 2016 Proceedings. 2016 [cited 2021 Jun 11];1:5119 Available from: http://dx.doi.org/10.21125/inted.2016.2304
4. Hill RK. What an Algorithm is? Philosophy and Technology [Internet]. 2016 [cited 2021 Jun 2];29(1):35-59, Available from: https://link.springer.com/article/10.1007/s13347-014-0184-5 DOI: 10.1007/s13347-014-0184-5
5. Vardi MY. What is an algorithm? Communications of the ACM [Internet]. 2012 [cited 2021 Jun 3];55(3):5-9, Available at: https://dl.acm.org/doi/10.1145/2093548.2093549 DOI: 10.1145/2093548.2093549
6. Hingole RS. Fundamentals of expert system. In: Advances in Metal Forming, Springer Series in Materials Science, vol. 206 [Internet]. Switzerland: Springer; 2015 [cited 2021 Jun 5]. Available from: https://link.springer.com/chapter/10.1007%2F978-3-662-44497-9_5 DOI: 10.1007/978-3-662-44497-9_3.
7. Tan CF, Wahidin LS, Khalil SN, Tamaldin N, Hu J, Rauterberg GWM. The application of expert system: A review of research and applications. ARPN Journal of Engineering and Applied Sciences. Asian Res Publ Network [Internet]. 2016; 11(4):2448-53.
8. Cruz CA, Uresti JAR. Player-centered game AI from a flow perspective: Towards a better understanding of past trends and future directions. Entertain Comput [Internet]. 2017 [cited 2021 Jun 20];20:11-24. Available from: https://www.sciencedirect.com/science/article/pii/S1875952117300095 DOI: 10.1016/j.entcom.2017.02.003
9. Wheat D, Masek M, Lam CP, Hingston P. Modeling perceived difficulty in-game levels. In ACM International Conference Proceeding Series. 2016 [cited 2021 Jun 20];1:78-81. Available from: https://dl.acm.org/doi/10.1145/2843043.2843478 DOI: 10.1145/2843043.2843478
10. Kölling M. The Greenfoot Programming Environment. ACM Transactions on Computing Education [Internet]. 2010 [cited 2021 Jun 5]:10(4):1-21. Available from: https://dl.acm.org/doi/10.1145/1868358.1868361 DOI: 10.1145/1868358.1868361

11. Oracle [Internet]. What is Java and why do I need it? 2018 [cited 2021 Jun 2]. Available from: https://java.com/en/download/faq/whatis_java.xml
12. Wang K, Shang W. Outcome prediction of DOTA2 based on Naïve Bayes classifier. In Proceedings - 16th IEEE/ACIS International Conference on Computer and Information Science [conference proceedings on the internet]; 2017 May 24-26; Wuhan, China. ICIS; 2017 [cited 2021 Jun 2]. p. 591-93. Available from: https://ieeexplore.ieee.org/document/7960061 DOI: 10.1109/ICIS.2017.7960061
13. Anwar SM, Saeed SMU, Majid M, Usman S, Mehmood CA, Liu W. A Game Player Expertise Level Classification System Using Electroencephalography (EEG). Appl Sci [Internet]. 2018 [cited 2021 Jun 2];**8**(1):18-24. Available from: https://core.ac.uk/display/194095006 DOI: 10.3390/app8010018
14. Al-Bow M, Autin D, Edgington J, Fajardo R, Fishburn J, Lara C, Leutenegger S, Meyer S. Using Greenfoot and games to teach rising 9th and 10th-grade novice programmers. Proceedings of Sandbox 2008: An ACM SIGGRAPH Videogame Symposium, Sandbox'08. 2008:55-60. Available from: https://dl.acm.org/doi/10.1145/1401843.1401856 DOI: 10.1145/1401843.1401853
15. Ponce P, Meier A, Méndez JI, Peffer T, Molina A, Mata O. Tailored gamification and serious game framework based on fuzzy logic for saving energy in connected thermostats. J Clean Prod [Internet]. 2020 [cited 2021 Jun 2]:262;1-39 Available from: https://www.sciencedirect.com/journal/journal-of-cleaner-production/vol/262/suppl/C? DOI: 10.1016/j.jclepro.2020.121167.
16. Fauzan MN, Saptarini NM. Implementation of fuzzy logic controllers to maintain water temperature in hydroponics nft for lollo verde lettuce (Lactuca sativa L.). Int J Appl Pharm [Internet]. 2021 [cited 2021 Jun 2];13(3):23-27. Available from: https://innovareacademics.in/journals/index.php/ijap/article/view/41414/24572 DOI: 10.22159/ijap.2021.v13s3.04
17. Zakiah A, Fauzan MN. Collaborative Learning Model of Software Engineering using Github for informatics students. In: Proceedings of 4th International Conference on Cyber and IT Service Management, CITSM 2016 [conference proceedings on the internet]; 2016 Apr 26-27; Bandung, Indonesia. New York: IEEE; 2016 [cited 2021 Jun 2]. Available from: https://ieeexplore.ieee.org/abstract/document/7577521 DOI: 10.1109/CITSM.2016.7577521u4444a
18. Machado AFV, Clua EW, Zadrozny B. A method for generating emergent behaviors using machine learning to strategy games. Brazilian Symposium on Games and Digital Entertainment. SBGames; 2016 [cited 2021 Jun 3]. Available from: https://www.infona.pl/resource/bwmeta1.element.ieee-art-000005772267 DOI:10.1109/SBGAMES.2010.21
19. Pradeep KR, Naveen NC. Lung Cancer Survivability Prediction based on Performance Using Classification Techniques of Support Vector Machines, C4.5 and Naive Bayes Algorithms for Healthcare Analytics. In: Procedia Computer Science, Elsevier B.V. 2018 [cited 2021 Jun 2];132,pp. 412-20, Available from: https://www.sciencedirect.com/science/article/pii/S1877050918308962 DOI: 10.1016/j.procs.2018.05.162.
20. Patil TR, Sherekar SS. Performance Analysis of Naive Bayes and J48 Classification Algorithm for Data Classification. Int J Comp Sci Appl [Internet]. 2016 [cited 2021 Jun 6];6(2):256-61.
21. Witten IH, Frank E, Hall MA, Pal CJ. Data Mining: Practical Machine Learning Tools and Techniques. Data Mining: Practical Machine Learning Tools and Techniques [Internet]. Singapore: Elsevier Inc.; 2016 [cited 2021 Jun 2].621 p. Available from: https://dx.doi.org/10.1016/c2009-0-19715-5.
22. Jayaraman A, Kurtz N, Ragunanthan B, Aldrich R. LiDAR Based Sensor Verification. SAE Technical Papers [Internet]. 2018 [cited 2021 Jun 2]; 43 p. Available from: https://www.sae.org/publications/technical-papers/content/2018-01-0043 DOI: 10.4271/2018-01-0043.
23. Park K, Mott BW, Min W, Boyer KE, Wiebe EN, Lester JC. Generating educational game levels with multistep deep convolutional generative adversarial networks. In: Proceedings of the IEEE Conference on Computational Intelligence and Games [conference proceeding on the internet]; 2019 Sep 26; London, UK. New York: IEEE; 2019 [cited 2021 Jun 2]. Available from: https://ieeexplore.ieee.org/document/8848085 DOI: 10.1109/CIG.2019.8848085
24. Kumar I, Dogra K, Utreja C, Yadav PA. A Comparative Study of Supervised Machine Learning Algorithms for Stock Market Trend Prediction. In: Proceedings of the International Conference on Inventive Communication and Computational Technologies [conference proceeding on the internet]; 2018 Apr 20-21; Coimbatore, India. New York: IEEE; 2018 [cited 2021 Jun 2]. p. 1003-7. Available from: https://ieeexplore.ieee.org/document/8473214 DOI: 10.1109/ICICCT.2018.8473214