International Journal of Advanced Trends in Computer Applications
www.ijatca.com

Framework for An Anthill Invader Game: Knowledge Encoding In-Game Mechanics

[1]Obe O. Olumide
[1]Department of Computer Science
Federal University of Technology, Akure, Nigeria
[1]*oluobes@gmail.com*

Abstract: *The use of games and games technology has been in use in recent times to investigate their possible impact in teaching abstracts concepts. The underlying hypothesis is that the motivating qualities of games may be harnessed and embedded in a game-based learning system (GLS) to accelerate the comprehension of abstract concepts. In this paper, an AntHill Invader Game (AIG) framework is presented to guide in the design of gamified Ant social collaborative behaviour in an Ant Hill Colony. A detailed workflow illustration and mathematical model for the use of AIG in the development of GLS is presented in this paper.*

Keywords: Game, Learning, agent, Animations, Physics Mechanics, Graphics.

I. INTRODUCTION

A scientific theory tries to abstract the most essential aspects of a given situation, analyse them using certain assumptions and procedures, and at the end derive some general principles and predictions that can be applied to individual instances [10]. Game Theory is an interesting sub-branch of Artificial Intelligence (AI) that is focused on creating a form of entertainment and recreation for computer users combining some principles of scientific theories in domains such as computer graphics, computer programming, and artificial intelligence to create exciting video games [1] [2]. The subject matter of game theory is exactly those interactions within a group of individuals (or governments, firms, etc.) where the actions of each individual affect the outcome that is of interest to all [10].

Learning can be seen as a process of assimilating a new concept. However, this can be a problem in a situation where the subject is abstract and students have no previous personal experience about the subject. If students can have enough personal practical experience (either physically or virtually) with the new subject, it will greatly increase their rate of assimilation.

Computer video games have found applications in training human abilities, such as the cognitive flexibility trait [11], spatial-visual attention [12], and spatial resolution [13]. Game-based learning has led to the development of serious games designed to educate players in a broad variety of topics like genetics [14], biological consequences of alcohol abuse, teach about a specific knowledge, and also motivate players to consider a science career [15].

An Anthill is the natural habitat of most terrestrial ants where they live together as a colony. It is characterized by strong teamwork covering general and survival activities like food searching and provisions which is being handled by the worker; adequate security against any form of an intruder be it a man or other terrestrial animals by soldiers and the procreation of new generation ants which is mutually carried out by queens and kings of the colony.

An Ant Hill Invader game (AIG) or Ant Colony game (AC) is a game that is meant to reflect and model the human scenario of invading an ant colony for the sole purpose of capturing the queen ant(s). The game is a video game that draws relevance in Computer Science (like Graphics, Artificial Intelligence, and Programming) and Physical Sciences [4].

The work proposes the implementation of the anthill algorithm and anthill modeled behaviors in an anthill colony and rendering it in a Two-Dimensional (2D) computer video. The game design is challenging enough for the game player to be overwhelmed by the challenge and stimulating enough for the player to have enough interest in playing the game. Also, the game features a

systematic increment in difficulty level as the player moves from one level of the game to another.
The game is modeled such that the defender behaves intelligently for the achievement of its goals.
The goals are to:

i. intelligently locate and navigate their way to the current position of the player agent
ii. be able to attack the player agent with an optimal chance of defeating him in the game
iii. demonstrate randomness in characters, behavior, and movement which is natural in the ant colony (AC).

II. LITERATURE REVIEW

In [16], it was observed that the growing appreciation that the conventional approach to the process of teaching does not address the social, emotional, mental, and motivational needs of the new generation. The educational system in many institutions cannot meet the demands of the present digital age ([17]; [18]). Digital technologies are described as tools that will enhance collaboration and motivate learners to reengage with education [19] and enable them to develop the new multimodal literacy skills required for today's knowledge economy [20].

The exploitation of Information Technology as a tool to take the real entertainment industrial advantages in fostering education cannot be overemphasized ([21],[22], [23],[24]). In many cases, digital games have proven to be powerful education tools not inherently, by their design, but mainly through their effective classroom implementation [25].

Games can make learning more engaging and satisfying, offering at the same time, the possibility to expose learners to experiences that would be impossible, unsafe, or at least impractical to reproduce in the real world [26]. Studies show that even young children under the age of 8 years are frequent users of digital games and applications ([27], [28], [21], [23]). The popularity of the game in the dominant culture of the new generation has spurred the interest of the educational community, with several educators and researchers to seek different approaches, in using digital games in the classroom environment ([26],[29],[22])

Digital games are gaining wide recognition as an effective way to create socially interactive and constructivist learning environments. Studies indicate that playing video games gives learners a 'mental workout' and the structure of activities embedded in computer games develops several cognitive skills [30].

Section III of this paper describes the method for the design of the AIG framework. Section IV discusses the expected result while section V and VI are about the conclusion and future direction for the implementation and evaluation of the framework discussed in this paper respectively.

III. METHODS

3.1 The AC System Architecture

The diagrammatic representation of the system and each component that makes up the system are presented in Figure 1. The System architecture consists of three major components which are the AC:
1. Game Entities
2. Game Logic
3. Game Look and Feel

The development of each of these components entails several sub-components (sub-modules) but to avoid unnecessary complexity in the architectural layout of the system, these sub-components are not captured in the system's overall architecture. The choice of technical requirements and development tools used for the app for each sub-component is iterated below.

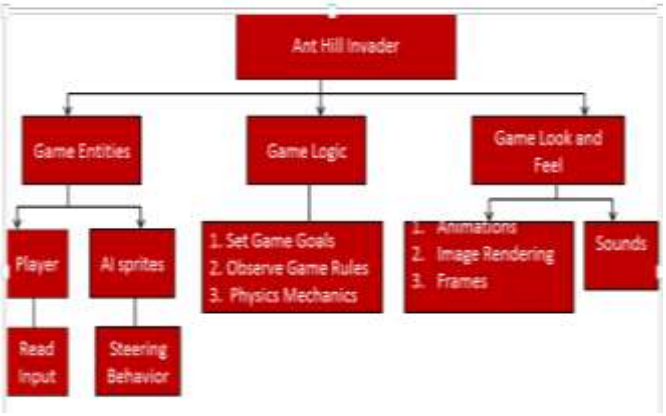

Figure 1: The AC System Architecture (Adapted from Steven, 2016)

1. The AC Game Entities

These are the basic entities that will be participating in the game world. They are majorly
a) Player: The player is the digital representation of the user who is playing the computer game. It is the avatar of the user in the game world. The basic actions of the Player entity are as follows:

i. Navigate through the game world
ii. Achieving goals: At every stage in the game, the goal of the stage will be presented in detail for the player

iii. Gathering Crisp Unit: The player will be expected to gather crisps in form of coins in the game world; this will enable it to enrich its pocket to avoid stronger ammunition that will improve its attacking mechanism, health improvement, bullet refilling, and many other actions that will involve it buying with its crisp unit.

iv. Defend against AI Sprites attacks: It will be solely responsible to defend itself against any form of attack that the AI Sprites might launch against it.

v. Health: The player will have health as its basic living requirements. The health at the beginning of a new game will be very healthy (100%).

b) Artificial Intelligence (AI) Sprites: The AI Sprites are programming controlled entities in the game world. They are to create a sizeable opposition for the player in proportion to the increase in the stage. The AI Sprites are to be generated by the use of the Genetic Algorithm (the individual chromosome or combination of chromosomes implicitly generate the values at which each of its resources function) to ensure that preceding stage sprites are stronger and create tougher opposition for the player in the game world. In the game world, the actions to be carried out by AI Sprites are as follows:

-To navigate to the current position of the player in the game world: This will be achieved for the AI Sprite by individual built-in steering behaviour.

-Launch an attack against the player: They will be expected to launch an attack against the player to distract it from pursuing its goal, rob it of its gift that has limited elapsed time before being lost, and primarily, to alienate the player in the game world by depleting its health status to zero.

The resources that are provided for the sprites in the game world are as follows:

i. Health: The AI sprite will have a resource and health. This resource is the basic living factor that they have

ii. Ammunition: basically, ant mouth will be used for their basic attack ammunition. They will be able to sting the player. Each successful attack against the player will cost the player a predefined unit of decrement in its health. However, the ammunition of the Sprites has unlimited usage in the game.

2. The AC Game Logic

This is the core control of the game and it is where all the rules of the game are realized and organized to favour the easy flow of the game. It is also where the states and activities of both the player and Sprites are being updated and noted respectively. Every game logic portion of the system runs for each frame such that the game logic rules and goals are always being observed and become the input to render next frame graphics. This portion is well optimized in time and memory usage as the evaluation reveals.

In general, in in-game development, game logics have a high overhead cost due to its complexity and their high importance [3]. They are generally meant to be very critical in the overall performance of the game system. The game logic includes:

- **Set Game Goals**

This game logic portion is the module that is concerned with the setting of the game goals for the user at the very beginning of the game and stages in the game. The goals of the game are directly proportional to the current game stage of the user.

It is a dynamic module as it is created with its logic. The game goal module ensures that the challenge presented for the player is realistic and increases as the player proceeds in the game. However, it is not modeled to be explicitly the same for every game instance launched and play by the human player. This is to ensure unpredictability and unnecessary familiarity of the player to the game world which is its true self is another mechanism to ensure that the game world presents a real challenge to the player.

- **Observe Game Rules**

Game rules can be defined as the set of logical principles in the game that ensures that activities, events, and actions of both the player and Sprites in the game have a predefined and predictable consequence on both the actor and the receiver of actions in the game world. Also, game rules are extremely important in validating the game world after a series of activities have taken place. Game rules ensure the proper updates of various entities that are represented in the game world and consequently, influence how the human player views the game world via the game user interface.

In this work, the game rules module will be called at a specific time from the commencement of a stage to the completion of it. It is here that the overhead cost in memory and CPU usage is meant to be optimized. As it uses to be, all the game rule modules must be run always to avoid the risk of not taking any event with any slight chances of occurrence no matter how little. In game programming, the more the game world models reality, the higher the complexity of the game rule and hence, the higher the demands of the game software on the usage of system resources which includes memory, CPU time, VGA, etc.

Also, the game rule is a superset for the physics mechanics but in this architecture, Physics Mechanics is given a stand-alone unit in the architecture because of its great importance in modeling reality in the game development. So, at every call to the game rule module, the module that maintains the physical mechanics has

been implicitly called in it to ensure that the virtual world obeys the physical world.

- **Physics Mechanics**

The physics mechanics of the game will be implemented to mimic reality and observed the universal law that is guiding our physical law. Some of these laws that will be recognized and observed in the game are as follow:

i. Collision Detection: In the physical world, it is a general phenomenon that matters generally occupied unique space in their environment. This implies that no two physical entities can logically occupy the same place simultaneously in the physical world. The basic collision detection rule that will be kept and observed are:

a) No two Sprites will be overlapped in the game world

b) Player actions will be monitored and the consequences of a collision with any of the game entities will be observed.

c) All the game entities will have a unique vector space at every instance of the game. Otherwise, will trigger an event to observe the rule e.g. collision of sprite and the bullet should result in hitting off the sprite among others

ii. Law of Motion and Inertia: The basic laws of motion states that it will always require a force either internal or external force before a move is to be produced. Also, any object in motion is being influenced by many forces and thus have no predictable geometry as it navigates in the game world.

iii. Velocity: Each game entity will have a different velocity in the game world to mimic reality. This will be observed in the game world by the creation of different displacement rate of each of the game entity in response to its velocity.

iv. Law of Gravitation: Each entity of the game is to conform to the law of gravitation that all objects are to be grounded. Though the game world will be presented on x, y plane and there will be no physical movement along the z-plane. This is where the law of gravitational pull will be observed.

v. Relative Size: The size of each object should be relatively similar to the physical world. That is, Bullets are smaller in size while ants are relatively small to the predators in the game world. This is to model the reality in the game. However, the ratio of their conformation might not be observed. This is being handled in the graphical production of entity images and the dimension of the length by width pixels handles their sizes in the game world.

3. The Game Look and Feel

The AC game will be rendered in 3D and will provide excellent Human-Computer Interaction features. It will offer all features provided by other computer games.

3.2 Mathematical Model

Let the cost of a bullet be given as Bc; Number of targeted ants be na,

Thus, for a given Stage, S, the cost of a bullet is Bc * na and for a complete mission, the cost of a bullet is:

Where: Bci is the cost of a bullet for an ant in stage i,
nai is the net aggregate of minimum ants to kill in stage i
For the Life meter at any given time t,

The Life meter of the attacker experienced linear decrement when sting by soldiers in the colony

$L(f) = \min\{0, L(i) - P\}$ (L(f) has a lower bound of zero in which case the attacker is considered dead and fail the stage challenge)
where L(f) is the new value of life meter after sting
L(i) is the initial value of the life meter before the sting
P is the poison cost of the sting on health ($1 <= P <= 10$)

The Life meter of the attacker experienced linear increment when receiving a life booster as part of a random gift in the game stage with the following relationship:

$L(f) = \max\{100, L(i) + B\}$
(L(f) has an upper bound of 100 in which case the attacker is considered most healthy)
where L(f) is the new value of life meter after health booster
L(i) is the initial value of the life meter before receiving a boost
B is the health booster benefit gain ($1 =< P <= 20$)
The maximum booster in a given stage is 2

However, $0<L(t) <=100$ for any given time, t, during the mission.

The expected duration of the time is given as, $T(t) = K - t$
where T(t) = left over time after time, t; K= expected time cost of mission and t is the time spent in the mission.

1) Collision: Collision detection is the mechanism to determine when two or more objects in the space are occupying the same location in 2D or 3D space. The mathematical model for collision detection in this work is as follow:

For any two arbitrary objects A and B Where A is defined by a vector (a1, a2, a3, a4) and B with a vector (b1, b2, b3, b4) with

a1, b1= x-coordinate of A and B in the space
a2, b2= y-coordinate of A and B in the space
a3, b3 = width of A and B
a4, b4= height of A and B
A is said to have collided with B if:
(b1 < a1 + a3 < b1+ b3) or (b2 < a2 + a4 < b2 + b4)

2) Health Status: Let f(x) define a function for state of health of Player (i.e health status)
Let Y= {"Very Healthy", "Healthy", "Weak", "Very Weak", "Died"}
Y= f(x) where y is the discrete state of a player in the game world.
X<=0 => Y= Died
0<=x<=20 => Y=Very Weak
21<=x<=40 => Y = Weak
41<=x<=80 => Y=Healthy
81<=x<=100 => Y=Very Healthy
Such that. 0<=x<=100

3) Sting Cost: A player experienced a string that deprecates its health status by F(S) = X-S
where X= Initial health level of player
S= Penalty cost of the sting
Such that S > 0

4) Scoring: The game score for a game level is cumulative obtain until the stage is over or the user loses the stage due
to some difficulty. It depends on several factors in a game level.
Score = 20(t) + 200(c) + 200(s) + 2000(g)
Where:
t = time spent in seconds in a game stage
c = number of coin gathered in a game stage
s = number of ant killed in a game stage
g = number of game gift collected

IV. EXPECTED RESULTS

The components design of the AC will be implemented and integrated. The program will be run for a long interval of time and the record of the heap dump every 3 minutes of running will be taken to observe Java Virtual Machine memory consumption and CPU usage. The system will be evaluated for a good performance in CPU utilization and Memory consumption. The game would be tested by some volunteer students for user experience. Knowledge of the Ant Colony teaching/learning concept experience in a traditional classroom environment would be evaluated vis-a-vis the knowledge acquired by the same student volunteers of the game tool.

V. CONCLUSION

In this paper, an AntHill Invader Game (AIG) framework; a participatory design-oriented process for gamifying collaborative learning behaviours in an AntHill colony is presented. A detailed work-flow illustration and mathematical model for the use of AIG in the development of GLS is presented.

VI. FUTURE DIRECTION

The framework will be implemented and evaluated for computer resource utilization and the effect of a game-based learning tool for effectively aiding the understanding of abstract concepts

REFERENCES

[1] Beginning of Game Programming: www.coursera.com Date accessed: 21/12/2019)
[2] K. Charles (2012). Programming 2D Games. CRC Press Taylor & Francis Group 6000 Broken Sound Parkway NW; Suite 300 Boca Raton; (1992) 327-342
[3] JavaGameTuturial: http://zetcode.com/tutorials/javagamestutorial/animation, (2016)
[4] O. Steven (2016) Introduction to Game Programming in C++;www.cs.cmu.edu;
[5] B. D. Glass, W. T. Maddox, and B. C. Love (2013) "Real-time strategy game training: the emergence of a cognitive flexibility trait," PLoS ONE, vol. 8, no. 8
[6] C. S. Green and D. Bavelier (2003) "Action video game modify visual selective attention," Nature, vol. 423, no. 6939, pp. 534–537
[7] C. S. Green and D. Bavelier (2007). "Action-video-game experience alters the spatial resolution of vision," Psychological Science, vol.18, no. 1, 88–94
[8] L. A. Annetta, J. Minogue, S. Y. Holmes, and M. T. Cheng (2009), "Investigating the impact of video games on high school students' engagement and learning about genetics," Computers & Education, vol. 53, no. 1, pp. 74–85
[9] L. M. Miller, C. I. Chang, S. Wang, M. E. Beier, and Y. Klisch (2011) "Learning and motivational impacts of a multimedia Science game," Computers & Education, vol. 57, no. 1, (2011)1425–1433
[10] L. Kockesen and E. A. Ok (2007) An Introduction to Game Theory: http://home.ku.edu.tr/~lkockesen/teaching/econ333/lectnotes/uggame.pdf
[11] M. E. DeRosier and J. M. Thomas (2018) "Video games and their impact on teens mental health," in Proceedings of the Technology and adolescent mental health, M. Moreno and A. Radovic, Eds., pp. 237–253, Springer International Publishing, 2018.
[12] J. J. Horan (1996) "Effects of Computer-Based Cognitive Restructuring on Rationally Mediated Self-Esteem," Journal of Counseling Psychology, vol. 43, no. 4, pp. 371–375
[13] D. H. Goh., R. P. Ang, and H. C. Tan (2008) "Strategies for designing effective psychotherapeutic gaming

interventions for Children and adolescents," Computers in Human Behavior, vol. 24, no. 5, pp. 2217–2235

[14] D. S. McNamara, G. T. Jackson, and A. Graesser (2010) "Intelligent tutoring and games (ITaG)," in Gaming for classroom-based learning: Digital role-playing as a motivator of study, Y. Baek, Ed., pp. 44–65, formation Science Reference, Hershey, Pa, USA

[15] K. Fenstermacher, D. Olympia, and S. M. Sheridan (2006) "Effectiveness of a computer-facilitated, interactive social skills training program for boys with attention deficit hyperactivity disorder," School Psychology Quarterly, vol. 21, no. 2, pp. 197-224.

[16] S. Papadakis (2018). The use of computer games in a classroom environment. International Journal of Teaching and Case Studies. 9. 1. 10.1504/IJTCS.2018.10011113.

[17] K. Becker and E. Gopin (2016) 'Selection criteria for using commercial off the shelf games (COTS) for learning', in Learning, Education and Games, Ch. 2, Vol. 2 pp.43–60, Bringing Games into Educational Contexts, ETC Pres.

[18] J. Palfrey and U. Gasser (2013) Born Digital: Understanding the first Generation of Digital Natives, Basic Books, 387 Park Avenue South, New York, NY 10016-8810, USA.

[19] S. Papadakis (2016) 'Creativity and innovation in European education. 10 years eTwinning. Past, present and the future', International Journal of Technology Enhanced Learning, Vol. 8, Nos. 3/4, pp.279–296.

[20] P. Lacasa (2013) Learning in Real and Virtual Worlds. Commercial Video Games as Educational Tools, Palgrave Macmillan, New York.

[21] S. Papadakis, M. Kalogiannakis and N. Zaranis. (2016a) 'Comparing tablets and PCs in teaching mathematics: an attempt to improve mathematics competence in early childhood education', Preschool and Primary Education, Vol. 4, No. 2, pp.241–253.

[22] S. Papadakis, M. Kalogiannakis, V. Orfanakis and N. Zaranis. (2016b) 'Using scratch and app inventor for teaching introductory programming in secondary education. A case study', International Journal of Technology Enhanced Learning, Vol. 8, Nos. 3/4, pp.217–233.

[23] S. T. Papadakis, M. Kalogiannakis, N. Zaranis (2016c) 'Developing fundamental programming concepts and computational thinking with scratch jr in preschool education, a case study', International Journal of Mobile Learning and Organisation, Vol. 10, No. 3, pp.187–202.

[24] M. Kalogiannakis and S. T. Papadakis (2017) 'Combining mobile technologies in environmental education: a Greek case study', International Journal of Mobile Learning and Organisation, Vol. 11, No. 2, pp.108–130.

[25] J. Groff, C. Howells and S. Cranmer (2010) The Impact of Console Games in the Classroom: Evidence from Schools in Scotland, Futurelab, UK.

[26] M. B. Carvalho (2016) Serious Games for Learning: A Model and a Reference Architecture for Efficient Game Development, Technische Universiteit, Eindhoven.

[27] S. Judge, K. Floyd and T. Jeffs (2015) Using mobile media devices and apps to promote young children's learning', in Heider, K.L. and Jalongo, M.R. (Eds.): Young Children and Families in the Information Age, pp.117–131, Springer Science + BUsiness Media, Dordrecht, DOI: 10.1007/978-94-017-9184-7.

[28] V. Orfanakis and S. T. Papadakis (2014) 'A new programming environment for teaching programming. A first acquaintance with enchanting', The 2nd International Virtual Scientific Conference – Scieconf, EDIS – University of Zilina, Slovakia, pp.268–273.

[29] V. Orfanakis and S. T. Papadakis, St. (2016) 'Teaching basic programming concepts to novice programmers in secondary education using Twitter, Python, Ardruino and a coffee machine', Hellenic Conference on Innovating STEM Education (HISTEM), 16–18 December 2016 University of Athens, Greece

[30] M. N. Giannakos (2013) 'Enjoy and learn with educational games: examining factors affecting learning performance', Computers and Education, Vol. 68, pp.429–439, ISSN 0360-1315.

Author Profile

Obe O. Olumide received the B.Tech and MSc. degrees in Computer Science from the Federal University of technology, Akure, Nigeria and the University of Lagos, Akoka, Yaba, Nigeria in 2000 and 2003, respectively. He obtained his PhD in System Engineering from the Politehnica University of Bucharest in 2010. He is a facaulty at the Department of Computer Science, Federal University of Technology, Akure, Nigeria.