

AN SIMULATION AND SURVIVAL BASED MOBILE GAME FOR PLAYERS DEVELOPING A SENSE OF ENVIRONMENTAL PRESERVATION USING MODELING AND MANUAL MANIPULATION PROGRAMMING

Yiwen Liu¹ and Yu Sun²

¹The high school Affiliated to Renmin University, Beijing,
Haidian, Zhong guan cun street, 37

²California State Polytechnic University,
Pomona, CA, 91768, Irvine, CA 92620

ABSTRACT

The citizens nowadays are often born and raised in well developed urban areas and have rarely, even never experienced the difficulties that wildlife are suffering from due to human actions. They are likely to possess sympathy, but never do these individuals are aware of how their deeds may affect the living of other species on planet Earth [1]. However, if someone reveals the bloody truth to the people, they are likely to change for the greater good.

In this paper, we mainly used unity modeling and java programming skills to develop an animal simulation game in order to show the damage done by mankind and resonate the feeling of empathy so the players may alter their actions to preserve the environment [2][3]. The player starts the game as an animal figure in a randomly generated map. The player will control the animal to move around to consume water and food for survival. Meanwhile, the animal must avoid the invasion of human poison lands closing in on the habitats for this figure. Eventually, the player will starve or be poisoned and fail to survive. By setting this result, we hope to arouse the sympathy in hearts and lead to some alteration to a person's habits.

KEYWORDS

Simulator, self-manipulation, protection, ecosystem.

1. INTRODUCTION

This simulation game is developed mainly to raise the alertness for environmental protection of individuals, also leading to changes in lifestyle to benefit nature. People living in the wealthy areas of the world has never been to the developing areas where humans have to invade the territory of wild beasts, driving them to further lands or maybe blind alleys. However, this damages the ecological balance significantly because the distinction rate of endangered animal species is higher than ever [4]. This topic has been widely discussed ever since humans entered the era of technology, people are struggling to find the balance of sustainable development in life to live in peace with other species and maintain an appropriate biodiversity [5][6][7]. This game has a significant benefit when compared with other products or slogans in society nowadays. This game can provide a first person perspective, an immersed experience for individuals that have

little chance to actually feel the negative effects we have done to the nature kingdom. The commercials, slogans and products in the market currently cannot let the people directly soak themselves in the situation of wild lives. Consequently, while playing the game, the character that that player represent will slowly fall into the situation of lacking food, water and shelter, as the player helplessly watch the hp figure decreases to 0, they will have a similar emotion with animals that actually strives to live under this state. By receiving this sense of substitution, the individuals are more likely to make actual adjustments to life.

Some of Unity techniques and systems that have been proposed to model the creature figures and the map used in our simulation game, which allows the user to operate the character freely by the keyboard. Also, Java scripts are written to allow the health bar and poison bar to elevate and drop, the camera to always follow the player and be able to rotate according to the movement of the mouse, the resource area to randomly generate and etc. However, these proposals assume deliberate methods of modeling in Unity, which is rarely the case in practice. Their implementations are also limited in functions, with samples given for the inability for C# code itself to assort a tatted map that meets the demands of containing all kind of landscapes and has precise height demands [8]. Other techniques, such as animal building, may not be accurately done in Unity, in the “create object” options, there are mostly basic shapes like sphere and cube to choose, but we wish for the player to recognize the animal at first sight for better game experience and a deeper substitute into reality. Consequently, a new application is put into practice for character establishment. Because Maya has a high degree of finesse, when modeling, we are able to access the angle and shape freely and fabricate an ideal figure. Therefore the method can grow more sophisticated and results in hypothetical results. A new issue comes up as Maya is actually applied, the flips and twists of models are easily misunderstood and we often fail to solve the math involved in this process. A second practical problem is that some users find it hard to understand the rules and goal of the game, some players may accuse the game for not owing the method to survive and win, questioning the play-ability of the simulator.

In this paper, in order to make reasonable to answer the previously mentioned queries, we developed some methods and completed researches. Our goal is to establish a mature, playable simulation game that has a high degree of liberty. Our method is inspired by the asp codes and the WWF game of environmental protection. There are some good features of ASP code. First, with ASP, we are able to create more interactive and data-driven web applications. ASP consists a variety of control like text boxes, buttons, and labels for assembling, configuring, and manipulating the code to create certain pages. Second, since we only employed ASP in the limitations and arrangements of the map, which only require the basic lines of ASP, this kind of code is straight forward written and is therefore more easier to understand. Third, ASP allows the map to grow to a significant size in a short amount of time, if we only used the loop in C#, it may take too long to operate the code. The ASP code is flexible and swift when coding. Therefore, we believe that the usage of this method can help us avoid most of the coding issues. Moreover, our game is inspired initially by the WWF game, we learned from it's strong points to offset our weakness. By learning to add some introduction to the end of the game and adding more interesting to our game, we can effectively avert the misunderstanding and dissatisfaction towards our simulator.

In two application scenarios, we demonstrate how the above combination of techniques increases the play-ability of the simulator. First, we show the usefulness of our approach by a comprehensive trial of the game. We play the role of a user and start from the initial page of the game. Beginning with a site that shows the options of "play", "instructions", and "menu". As the player chooses "play", the scene will shift to a randomly organized map containing all the landscapes, like water, food sand and so on. After entering, we will test the movement and the functions of each scale, making sure that food number rises when the animal stays in the resource

area and hp drops when the growth of the poison area outruns the player's speed. After playing throughout the whole procedure, we have an initial plan of aspects that needs to vary and other functions that can stay the same. Second, we analyze the evolution of our java code and debug every error that shows up, both in C# and in Unity. Sometimes we ignore a bracket or a semicolon that is required in the language of coding Java scripts. Other times we may forget to add a sphere collider on to the player or the resource area. According to the reminder panel in Visual studio and the error window in Unity, we are able to locate the line number of the error and the kind of mistake we have made, after finding it, we have to check the line itself and lines around it abortively to seek the solve. After rigid identification of the exact word or symbol, the debugging process is quiet simple. After making sure no mistakes occur while running, we can ensure the reliability of the code.

The rest of the paper is organized as follows: Section 2 gives the details on the challenges that we met during the experiment and designing the sample, both in the coding process and in the functioning of Unity; Section 3 focuses on the details of our solutions corresponding to the challenges that we mentioned in Section 2, referring to the ASP code mapping technique and the additional supplement page in the end of game; Section 4 presents the relevant details about the trials we did, by putting the simulator on probation with various individuals playing a trial and giving feedback on their experiences and opinions, following by presenting the related work in Section 5. Finally, Section 6 gives the conclusion remarks, as well as pointing out the future work of this project, which is closely related to spreading the game and more importantly the idea of preservation for the wilderness.

2. CHALLENGES

While building the game, I encountered several challenges. But with effort to solve the issues and flexibility to find other ways, I managed to polish Apocalypse the simulator into a mature game successfully delivering my concept of wildlife protection.

2.1. Game Ending

The fundamental one is how the game ends each level. I would like to make one round of the game short to cater to the rapid lifestyle of modern people, while strongly delivering my idea of environmental protection in reality [10]. The ideas I came up with were just not perfect, I thought of letting the animal character defeating the human invaders, but that is just very unrealistic and would not arouse much empathy in the players, I though about calculating the times reaching resources areas in a limited time for highest score, but it is weak in telling the idea. One night watching a movie, I cried as the leading roles die, giving the movie a tragic ending, and something hit me. Tragedy and death is often the most powerful weapon for emotions, if I created my game with a never winning mode, letting the animal character never surviving, much controversy would be raised and attention will be set on to the problem of ecosystem reservation. In the end, Apocalypse turned into a game of which the player will eventually have no way to live, longest surviving time would be kept as high score and two warning sheets will pop up after the player loses. This can achieve both my goals of short game and direct communication of thoughts.

2.2. Map Building

Another major problem that has been encountered is the is the map building, if the original C# codes were used in the script, the map would take minutes to respond even if it is already built when we were developing the game. This means that the player would be stuck in the loading

interface for a long time, leading to a misunderstanding of bugs appearing in the build and therefore losing interest in Apocalypse the simulator, even spreading a bad reputation for it. I solved this problem by obtaining a different method of coding, ASP codes. ASP or Active Server Page, requires a client-use of Java script, it is especially using when a user is setting some limitations or conditions to his or her page, it can be easily managed and understood by the computer. ASP codes I used in my code appears in paragraph-like contents, which allows me to consider the restriction input separately. Also, ASP codes reduces times greatly, showing this advantage greatly in large projects like Apocalypse. With the help of this technology, I solved the issue and the maps now can be generated within seconds on the player's end.

2.3. Character Choosing

The third issue I met is selecting the representative animal that is used in the game. I wish to use an animal that is well known around the world and is not a pet creature, also adapting the grassland and forest junction environment I generated in the map. At first it was hard to decide because so many animals own a high reputation and is favored by most people, like the pandas or the penguins, but selecting an animal meeting all demands are hard. After discussion with my school teacher, we worked out using the deer as the symbol and character of our game because it is not only known and loved, but also living in a unique habitat between the boundaries of the grassland and forest.

Meeting problems on the way of processing a project is inevitable, but using different methods or researching for extra information can always let one solve an issue in no time.

3. SOLUTION

Some of Unity techniques and systems that have been proposed to model the creature figures and the map used in our simulation game, which allows the user to operate the character freely by the keyboard. Unity is a 2D/3D engine and framework that gives you a system for designing game or app scenes for 2D, 2.5D and 3D. Unity allows us to interact with them via not only code, but also visual components, and export them to every major mobile platform and a whole lot more for free. Also, Java scripts are written to allow the health bar and poison bar to elevate and drop, the camera to always follow the player and be able to rotate according to the movement of the mouse, the resource area to randomly generate and etc. Unity is a native C++-based game engine. We can write code in C#, JavaScript (UnityScript) or, less frequently, Boo. Much of the power of Unity is in its rich scripting language, C#. We can use it to handle user input, manipulate objects in the scene, detect collisions, spawn new GameObjects and cast directional rays around the scene to help with our game logic. However, these proposals assume deliberate methods of modeling in Unity, which is rarely the case in practice. Their implementations are also limited in functions, with samples given for the inability for C# code itself to assort a tatted map that meets the demands of containing all kind of landscapes and has precise height demands. Therefore, we have obtained ASP codes to generate maps fast and accurate according to the area and height demands.

This simulation game is developed mainly to raise the alertness for environmental protection of individuals, also leading to changes in lifestyle to benefit nature. People living in the wealthy areas of the world has never been to the developing areas where humans have to invade the territory of wild beasts, driving them to further lands or maybe blind alleys. However, this damages the ecological balance significantly because the distinction rate of endangered animal species is higher than ever. This topic has been widely discussed ever since humans entered the era of technology, people are struggling to find the balance of sustainable development in life to

live in peace with other species and maintain an appropriate biodiversity. This game has a significant benefit when compared with other products or slogans in society nowadays. This game can provide a first person perspective, an immersed experience for individuals that have little chance to actually feel the negative effects we have done to the nature kingdom. The commercials, slogans and products in the market currently cannot let the people directly soak themselves in the situation of wild lives. Consequently, while playing the game, the character that that player represent will slowly fall into the situation of lacking food, water and shelter, as the player helplessly watch the hp figure decreases to 0, they will have a similar emotion with animals that actually strives to live under this state. By receiving this sense of substitution, the individuals are more likely to make actual adjustments to life.

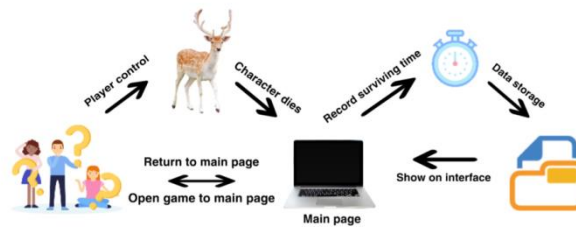


Fig.1 The overview of the solution

To achieve these goals, Apocalypse consists of three main components:

- The database containing all the scores and controls of the user in Unity.
- The controls of the player mode by C# and ASP in Visual studio.
- A interface for visual presentation of the game.

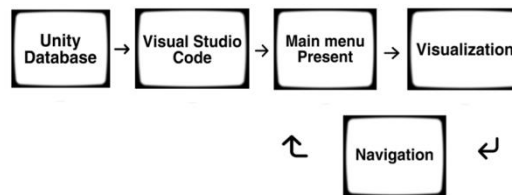


Fig.2 Overview of the components

Some of Unity techniques and systems that have been proposed to model the creature figures and the map used in our simulation game, which allows the user to operate the character freely by the keyboard. Unity is a 2D/3D engine and framework that gives you a system for designing game or app scenes for 2D, 2.5D and 3D. Unity allows us to interact with them via not only code, but also visual components, and export them to every major mobile platform and a whole lot more for free. Also, Java scripts are written to allow the health bar and poison bar to elevate and drop, the camera to always follow the player and be able to rotate according to the movement of the mouse, the resource area to randomly generate and etc.

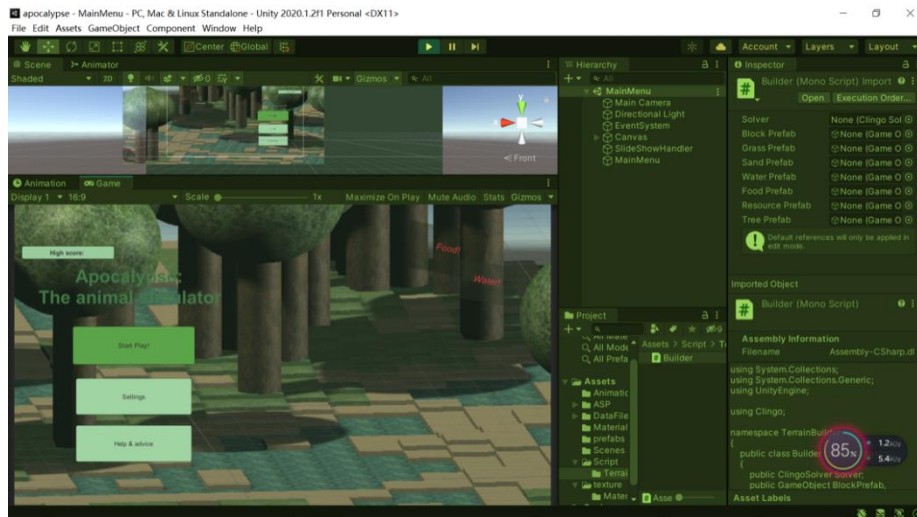


Fig.3 Unity UI

```

//Shuffle buildQueue to produce more random food placement
for (int i = 0; i < buildQueue.Count; i += 1)
{
    int index = Random.Range(0, buildQueue.Count);
    int temp = buildQueue[i];
    buildQueue[i] = buildQueue[index];
    buildQueue[index] = temp;
}

void BuildArea(int maxWidth, int minWidth, int maxHeight, int minHeight, int maxDepth, int minDepth)
{
    string boarderCode = GetBoarder(maxWidth, minWidth, maxDepth, minDepth);
    Debug.Log(GetFoodCode(new Vector3(maxWidth, 0, maxDepth), new Vector3(minWidth, 0, minDepth)));
    string foodCode = GetFoodCode(new Vector3(maxWidth, 0, maxDepth), new Vector3(minWidth, 0, minDepth));
    ClingoSolve(aspCode + boarderCode + foodCode, $" -c max_width={maxWidth} -c max_height={maxHeight} -c max
}

引用
string GetFoodCode(Vector3 max, Vector3 min)
{
    float minDistance = 30;
    List<GameObject> foodBlocks = FoodCount();
    Vector3 minMax = new Vector3(min.x, 0, max.z);
    Vector3 maxMin = new Vector3(max.x, 0, min.z);
    bool outOfRange = true;
    Debug.Log("FoodCount: " + foodBlocks.Count);

    foreach (GameObject blocks in foodBlocks)
    {

```

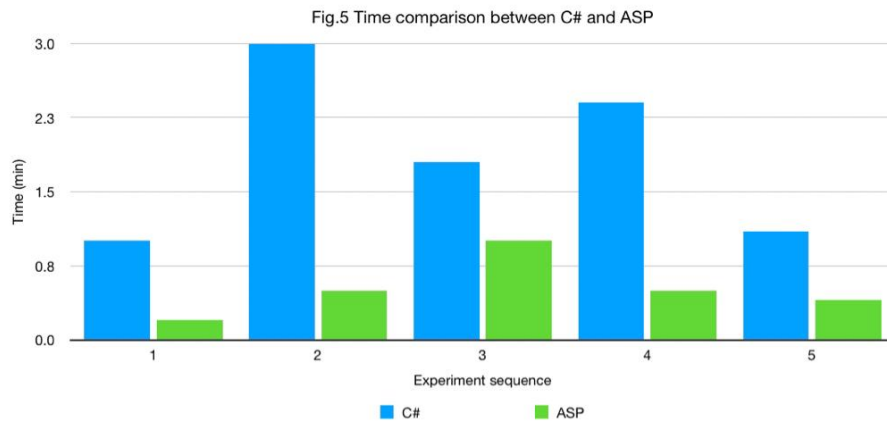
Fig. 4 Visual Studio coding

Unity is a native C# based game engine. We can write code in C#, JavaScript and Boo. Much of the benefit of Unity is in its abundant coding language, C#. We can use it to control player input, alter materials in the scene, spawn some imaginative GameObjects and obtain use of various shapes and color to build characters for the visual presence of the game. However, these proposals request complicated methods of modeling in Unity, which is rarely performed within the control of high school students like me. Their implementations are also restricted in abilities, with samples given for the unable of C# code itself to assort a tatted map that meets the demands of containing all kind of landscapes and has precise height demands. Therefore, we have obtained ASP codes to generate maps fast and accurate according to the area and height demands. Apocalypse obtains the technology of C#, ASP coding and Unity. Unity not only allows us to interact with scripts, but also sight components, and transfers them to every main mobile interface. Also, C#scripts are written to allow the health bar and poison bar to elevate and drop, the camera to always follow the player and be able to rotate according to the movement of the mouse, the resource area to randomly generate and etc.

ASP consists a variety of control like text boxes, buttons, and labels for assembling, configuring, and manipulating the code to create certain pages. Second, since we only employed ASP in the limitations and arrangements of the map, which only require the basic lines of ASP, this kind of code is straight forward written and is therefore more easier to understand. Third, ASP allows the map to grow to a significant size in a short amount of time, if we only used the loop in C#, it may take too long to operate the code. The ASP code is flexible and swift when coding.

4. EXPERIMENT

4.1. Experiment 1



Times	C#	ASP
1	1.0	0.2
2	3.0	0.5
3	1.8	1.0
4	2.4	0.5
5	1.1	0.4

Tab.1 Data of time used

Figure 5. Result of the experiment

Experiment one I designed to come up with a result solving the challenge of map building is that I measured the time used to generate the map using the original C# code and the time used to generate the map using the ASP code and made a comparison, repeated for 4 times. The results are shown above in Tab.1 and Fig.5, it can be concluded that building the map using C# code takes about 1.5-3.0 minutes, while using the ASP code only takes about 0.2-1.0 minutes, a significant decrease in time. Proving that obtaining ASP codes would be the best choice available.

4.2. Experiment 2

To reach the answer of the most suitable animal character used in the game, I have made a questionnaire and dropped it into my friend and family WeChat groups to collect some raw data. The question and conditions are shown below, in Fig.6 .

Choose an animal figure for Apocalypse

1. Widely known animal 2. Best to inhabit in forest or grassland areas 3. Arouse empathy from players 4. Wild life, not a pet



编辑

* 1. Animal choice

Fig.6 Questionnaire spread out

After gathering the data, the questionnaire program helped me to sort out the orders by most votes to least, and the best animal to use representing this game is the deer in most people's minds. It received 13 votes out of 58 votes. Deer votes take up approximately 22.4% out of total, tiger votes take up 17.2%, lion votes take up 13.8%, rabbit votes take up 12.1%, panda votes take up 10.3%, leopard votes take up 8.6%, horse votes take up 6.9%, wild buffalo votes take up 5.2%, and zebra votes take up 3.4%. The Results are present in Fig.7 below.

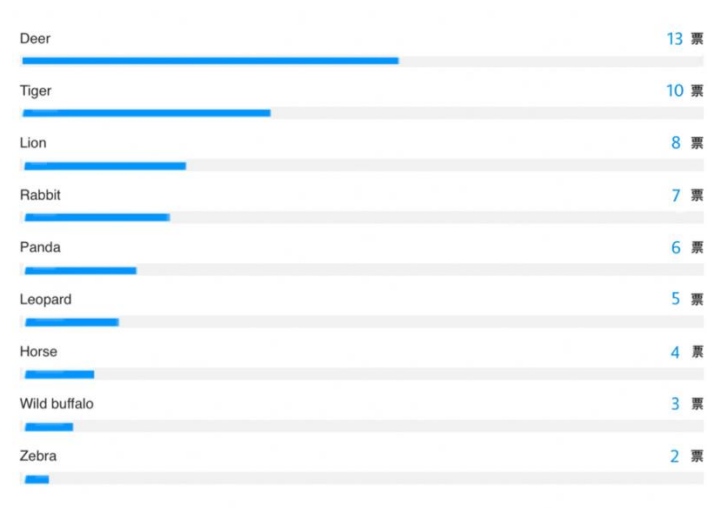


Fig.7 Experiment result

Both experiments have helped me obtain new ways or consider other people's opinions, which provided significant help since the game is based on environmental protection and hopes to arouse the empathy for wild lives, players will have to receive the perfected version of Apocalypse to become a persistent player and spread good reputation.

5. RELATED WORK

There are many animal simulators in the market currently and we chose three products to compare with Apocalypse. These games are: Roblox animal simulator, Wildcraft and Animalia [11]. First, Roblox is a game developed by @ragnar9878. This is a game where the player can choose from a variety of animals to play in the game. This is a bit similar to Apocalypse, however, after the player enters the map, things are very different from our game. The player in Roblox enters a big map with other players and attack with fireballs. Apocalypse surpasses this game by having a better idea of environmental protection because it is a thought of actual improvement for the world, also, Apocalypse has a better feeling of substitution since animals in real life do not fight each other with fireballs. Secondly, WildCraft is a simulator developed

published by Turbo Rocket Games. This game has single-player and multiplayer modes. Players can hunt in different worlds with real weather as different animals and create families. This way of playing is very similar to Apocalypse, even more deliberate. But Apocalypse has the uniqueness of never winning and cannot survive very long, which takes less energy and time meanwhile arousing more controversy and gives people a deeper impression. Lastly, the Animalia simulator developed by High Brazil Studio [12]. This game is released on steam and costs 70 rmb. The builds and fluency of the game is outstanding, but despite the great outlooks and precise plays, the game to me lacks an inner spirit, a theme. As I have mentioned, Apocalypse is unique because it devotes itself letting the players focus on the fact that wild-lives are in great danger and humans must set importance to the problem therefore making a change. The similar, competitive products in the market are as follows: Roblox animal simulator, Animalia survival and WildCraft simulation [13]. To begin with, Roblox simulator uses Roblox Lua coding for the game to run. Roblox Lua scripting is an embedded coding language to add features to your game. It is easy to learn and is a loose interpretation of modern day game programming. Although Roblox Lua is very similar to C#, the C# codes compile directly to a machine's native code, allowing it to be one of the fastest languages in the world, if optimized; Lua: Powerful, fast, lightweight, embeddable scripting language, but not as efficient as what it developed from. Moving on is the Animalia survival, the builds and code used in this game is varied and exquisite, the only technological or reality issue seems to be the degree of difficulty and the fee spent on it. Finally, comparing the WildCraft simulation to Apocalypse. The wildcraft simulator comes from Turbo Rocket games and uses c#. But the builds run more fluent than Apocalypse. The real life problem is similar to above, finance aid needs are larger to our game.

6. CONCLUSIONS

Environmental protection is the keystone of the project.

The Earth is under great pressure, millions of creatures has lived on this beautiful planet in harmony for centers, creating a saga of life forms [14]. However, as the invasion of human oppresses nature constantly, this balance is on the edge of tipping over. Many people today only have an idea of eco-protection, but makes no change to their life because they do not feel the actual crisis and survival pressure the wild beasts are facing. However, if we do not set importance starting now, we will be facing climate change, soil erosion, poor air quality, and undrinkable water in no time.

This simulation game is developed mainly to raise the alertness for environmental protection of individuals, also leading to changes in lifestyle to benefit nature. People living in the wealthy areas of the world has never been to the developing areas where humans have to invade the territory of wild beasts, driving them to further lands or maybe blind alleys. However, this damages the ecological balance significantly because the distinction rate of endangered animal species is higher than ever. This topic has been widely discussed ever since humans entered the era of technology, people are struggling to find the balance of sustainable development in life to live in peace with other species and maintain an appropriate biodiversity. This game has a significant benefit when compared with other products or slogans in society nowadays. This game can provide a first person perspective, an immersed experience for individuals that have little chance to actually feel the negative effects we have done to the nature kingdom. The commercials, slogans and products in the market currently cannot let the people directly soak themselves in the situation of wild lives. Consequently, while playing the game, the character that that player represent will slowly fall into the situation of lacking food, water and shelter, as the player helplessly watch the hp figure decreases to 0, they will have a similar emotion with animals that actually strives to live under this state. By receiving this sense of substitution, the individuals are more likely to make actual adjustments to life.

Some of Unity techniques and systems that have been proposed to model the creature figures and the map used in our simulation game, which allows the user to operate the character freely by the keyboard. Also, Java scripts are written to allow the health bar and poison bar to elevate and drop, the camera to always follow the player and be able to rotate according to the movement of the mouse, the resource area to randomly generate and etc. For future development of Apocalypse [15]. We have already released it on itch.io and is planning to make a mobile version for apple. After the game is launched on every planned page. We will publish a research paper concluding our goals and everything that has been done. When all is ready, we will enter some global computer science event like GameGala and Technovation to advertise Apocalypse and transmit our idea of environment protection. We will also broaden our horizons in these events and try to improve the simulator.

Currently, our team is meeting capital issues due to the lack of advertising and persuasive proposal to sponsors. But our game can arouse the responsibilities of people to protect the environment, very worth investing for a better tomorrow.

REFERENCES

- [1] Scheler, Max, and Graham McAleer. *The nature of sympathy*. Routledge, 2017.
- [2] Little, Margaret. "On basic unity." *International Journal of Psycho-Analysis* 41 (1960): 377-384.
- [3] Duarte, Nancy. *Resonate: Present visual stories that transform audiences*. John Wiley & Sons, 2013.
- [4] Southwood, Thomas Richard Edmund, and Peter A. Henderson. *Ecological methods*. John Wiley & Sons, 2009.
- [5] Rao, P. K. *Sustainable development*. Vol. 1. Blackwell Publishers, 1999.
- [6] Elliott, Jennifer. *An introduction to sustainable development*. Routledge, 2012.
- [7] Gaston, Kevin J., and John I. Spicer. *Biodiversity: an introduction*. John Wiley & Sons, 2013.
- [8] Kohler, Robert E. *Landscapes and labscapes*. University of Chicago Press, 2010.
- [9] Vartanian, Thomas P. *Secondary data analysis*. Oxford University Press, 2010.
- [10] Veal, Anthony J. "The concept of lifestyle: a review." *Leisure Studies* 12.4 (1993): 233-252.
- [11] Merika, Heidi. *Wildcraft: The science & spirit of wild plants as food & medicine*. Heidi Merika, 2019.
- [12] Base, Graeme. *Animalia*. Viking Kestrel, 1986.
- [13] Long, Roma U. *Roblox and Effect on Education*. Diss. Master's Thesis, Drury University, Springfield, MO, USA, 2019.
- [14] Schoenberg, Arnold. *Theory of harmony*. Univ of California Press, 1978.
- [15] Coppola, Francis Ford. *Apocalypse now*. University of Chicago Press, 2017.