AN INTELLIGENT AND INTERACTIVE GAMING SYSTEM TO PROMOTE ENVIRONMENT AWARENESS USING CONTEXT-BASED STORYING

Yilin Luo[1] and Yu Sun[2]

[1]Santa Margarita Catholic High School, Rancho Santa Margarita, CA 92688
[2]California State Polytechnic University, Pomona, CA, 91768

ABSTRACT

Since a child, I loved to play video games, especially platform games such as Metal Slug™, Mega Man™, etc.. Therefore, I was inspired to design my own platform game; with this paper, I have the opportunity to introduce our platform game, which is a "JFF Game" we developed using Unity and Visual Studio 2019.

KEYWORDS

Machine Learning, 3D design, Gaming System.

1. INTRODUCTION

I have loved to play platform games since a child and have learned a lot from doing so. For example, Metal Slug™ taught me that I should never give up when encountering difficulties, while Mega Man™ taught me that I should be honest all the time. I have always wanted to design my own platform game to express myself, and the development of our JFF Game has enabled me to do that. [1, 2, 3]

There are many software applications that can be used to make games, but each one will profoundly affect the way it is developed. There are two well-known tools in the field of game development. First, there is Unreal Engine. [4] The advantage of Unreal Engine is that the developer usage rate is high and it has many tools; however, some tools are difficult to learn and use. Second, there is CryEngine 3. [5, 6] The advantage of CryEngine 3 is that people can easily create complex and diversified special effects with it, but it also has disadvantages: for example, its developer community is not strong enough, and it is also difficult to learn.

We used Unity [7, 8, 9] and Visual Studio 2019 [10, 11, 12] to design our JFF Game. Compared with other software, the advantage of Unity is that it is easy to use and compatible with all systems (Windows, Mac, etc.). However, one issue with Unity and Visual Studio 2019 is that its available tools are limited, so it takes a long time to create complex and diversified effects. This was not so much of an issue for us, however, since our JFF Game does not require complex or diversified effects.

David C. Wyld: CSITY, NWCOM, SIGPRO, ASOFT, AIFZ, BDIoT, ITCCMA, CLSB, DTMN, MLNLP - 2021
pp. 157-162, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.111414

We employed several methods to test our JFF Game. For example, we asked friends to try it to see if they encountered any bugs. After some modifications, we believe there are no remaining bugs in the current iteration of our JFF Game.

The rest of this paper is organized as follows: in section 2, we introduce the challenges we encountered while creating our JFF Game. In section 3, we introduce how the game works in detail. In section 4, we illustrate two experiments we conducted to design the game. In section 5, we introduce three related works. In the final section, we summarize the process of creating our JFF Game.

2. CHALLENGES

In order to design our JFF Game, which we developed using Unity and Visual Studio 2019, a few challenges were identified as follows.

2.1. Challenge 1: Choosing an overall context and main character

Our first challenge was to choose an overall context for the game. We wanted to make a platform game, but weren’t sure which story frame might maximize the characteristics of this type of game and its perimeters. After watching a lot of videos and holding discussions with our team, we finally decided to use the medieval knight as our hero character. This figure appears in many classic works of Literature, so we were eager to use such a character to express our love and respect for classical works.

2.2. Challenge 2: Designing the game map

Our second challenge was to design the game map. Since I wasn’t sure how to design interesting maps, we ended up designing several test maps and asking friends for feedback. After summarizing our friends' opinions, we were finally able to build a solid game map.

2.3. Challenge 3: Finding a useable plot line

Our third challenge was to design plots. We initially wanted to make a "prince saves the princess" story, yet we found this type of story unpopular. My mother made the suggestion that "You can think about this story in another way,” and after listening to her, I suddenly had the idea that the importance of the prince's sword to the prince is similar to the importance of the princess to the prince. Therefore, we decided to write the story of a "prince looking for his sword.”

3. SOLUTION

Our JFF Game is a platform game made by Unity and Visual Studio 2019. Players can use “W,S,A,D” to control the direction of the game character and “Space” to make the character jump. There are three levels in our game: in level one, the goal is to find the sword; in levels two and three, the goal is to escape. There are two types of enemies. The first enemy attacks overtly by cannon fire, using bullets that are fired to attack the player. The second enemy attacks stealthily by ambushing the player. The total HP of the player is 100, and the damage caused by these enemies varies with the type of attack suffered.

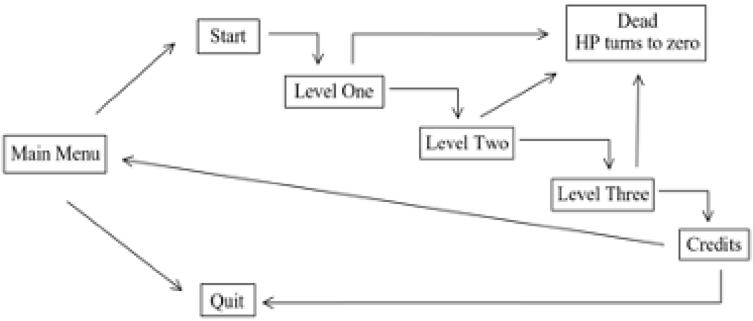
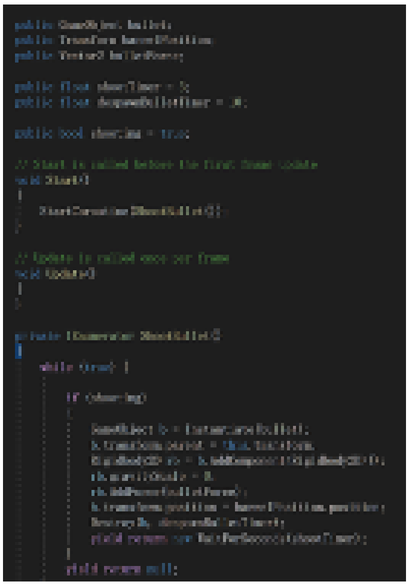


Figure 1. Game map

This segment of code (see Figure 2) describes how the overt type of enemy works. For example, from this segment of code, we can see that the shoot time is 5 and the despaen bullet time is 10.



```
public GameObject bullet;
public Transform barrelPosition;
public Vector2 bulletForce;

public float shootTimer = 5;
public float despawnBulletTimer = 10;

public bool shooting = true;

// Start is called before the first frame update
void Start()
{
    StartCoroutine(ShootBullet());
}

// Update is called once per frame
void Update()
{
}

private IEnumerator ShootBullet()
{
    while (true) {

        if (shooting)
        {
            GameObject b = Instantiate(bullet);
            b.transform.parent = this.transform;
            Rigidbody2D rb = b.AddComponent<Rigidbody2D>();
            rb.gravityScale = 0;
            rb.AddForce(bulletForce);
            b.transform.position = barrelPosition.position;
            Destroy(b, despawnBulletTimer);
            yield return new WaitForSeconds(shootTimer);
        }
        yield return null;
```

Figure 2. Code governing enemy behaviour

Below are screenshots from levels one and two (Figures 3, 4).

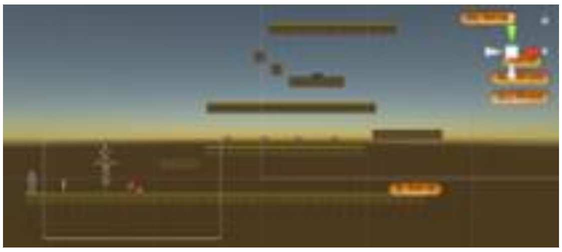


Figure 3. Screenshot from our JFF Game

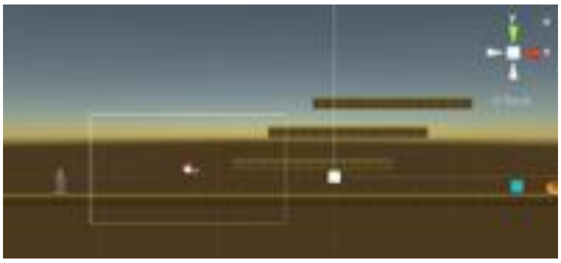


Figure 4 Screenshot from our JFF Game

When players touch the statue at the end of each level, they progress to next level automatically.

4. EXPERIMENT

We wanted to add a score system and time limit to our JFF Game, so we made a small game and invited ten friends to play it. Afterwards, we asked them to provide feedback. However, the results were surprising since many thought the scoring system and time limit both limited their ability to explore. Since players would likely only choose the pathway that would achieve the highest score possible instead of exploring others, we decided not to include a score system or time limit.

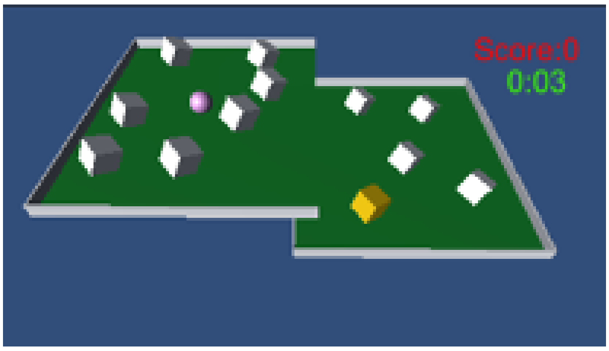
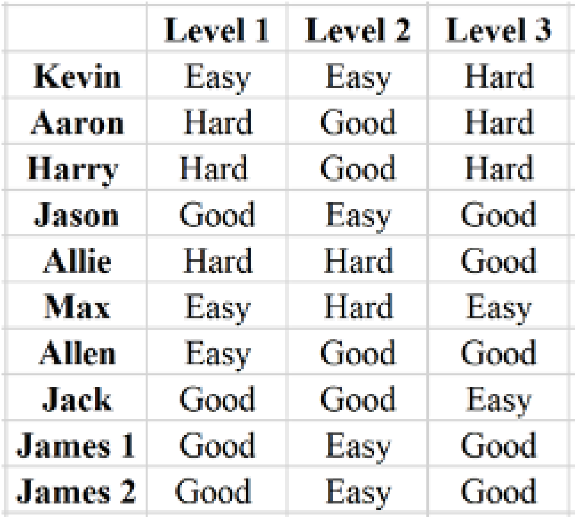


Figure 5. Screenshot of the test game

To test the difficulty of the game, we invited ten friends to play. Afterwards, we asked them to provide feedback. Our benchmark for editing was: more than three people saying it was "Good," more than one person saying it was "Easy," and more than one person saying it was "Hard." If these variable criteria were present, then the design of the level was considered good and reasonable. For level one, 3 people said it was "Easy," 3 people said it was "Hard," and 4 people said it was "Good." For level two, 4 people said it was "Easy", 2 people said it was "Hard," and 4 people said it was "Good." For level three, 2 people said it was "Easy," 3 people said it was "Hard," and 5 people said it was "Good." Therefore, the overall difficulty of our game was deemed appropriate (see figure 6).



	Level 1	Level 2	Level 3
Kevin	Easy	Easy	Hard
Aaron	Hard	Good	Hard
Harry	Hard	Good	Hard
Jason	Good	Easy	Good
Allie	Hard	Hard	Good
Max	Easy	Hard	Easy
Allen	Easy	Good	Good
Jack	Good	Good	Easy
James 1	Good	Easy	Good
James 2	Good	Easy	Good

Figure 6. Feedback for game difficulty

Our experiments proved useful since experiment one allowed us to reconsider using a score system or time limit, and experiment 2 revealed that the difficulty of our game was on par with our expectations.

5. RELATED WORK

Burke, Quinn, and Yasmin B. Kafai introduce the tools and communities of game making. This source helped us realize that we could use more tools to complete our game. For example, the RPG maker is useful, and can help someone complete a game with rich elements without the use of programming. [13]

James Paul Gee explores how video games can also be learning machines. This work influenced us to belive we can use games in many ways, for different purposes; they're not just for making people happy, but also possibly educating them about something. [14]

Susan Bermingham, et al. discuss how to collaborate to make games. This essay influenced our belief that if we wanted to make a better and more interesting game, a key component of the design phase would be to collaborate with others and discuss ideas freely.

6. CONCLUSION AND FUTURE WORK

We used Unity and Visual Studio 2019 to make our JFF Game. We chose Unity because, compared to other software, it is easy to use and compatible with all systems. In our game, players use "W,S,A,D" to control the direction of the character and "Space" to make the character jump. Our JFF game is a platform game with three levels, and the ultimate goal is to find the sword and escape the forest. There are two kinds of enemies to avoid, and players have 100 HP.

We did two experiments. Before starting, we ran experiment 1. As a result, we deleted the score system and time limit that were initially present. After finishing the draft of our game, we ran experiment 2. Experiment two confirmed that we didn't need to change the difficulty level of our game.

There are still a few issues remaining to resolve. Our JFF Game has only three levels, so it may not be complex enough to satisfy some players. Also, there are no complex and diversified special effects, and the plot can be dull. The character's movements are still slightly stiff, and there are only two kinds of enemies, which may be boring for some players. Also, when players become familiar with the enemies' attack routines, the game can become too simple to remain challenging.

To resolve these issues, we may make the game more complex with more diversified special effects. To do this, we plan to collaborate with other people so the game can become better through continued input and testing.

REFERENCES

[1] Schrier, Karen. "Designing and using games to teach ethics and ethical thinking." *Learning, education and games* 141 (2014).

[2] Shaffer, David Williamson, and James Paul Gee. *How computer games help children learn*. New York: Palgrave Macmillan, 2006.

[3] Shaffer, David Williamson, and James Paul Gee. *How computer games help children learn*. New York: Palgrave Macmillan, 2006.ß

[4] Sanders, Andrew. *An introduction to Unreal engine 4*. CRC Press, 2016.

[5] Tracy, Sean, and Paul Reindell. *CryENGINE 3 Game Development: Beginner's Guide*. Packt Publishing Ltd, 2012.

[6] Sousa, Tiago, Nickolay Kasyan, and Nicolas Schulz. "CryENGINE 3: Three Years of Work in Review." *GPU Pro* 3 (2012): 133-168.

[7] Halpern, Jared. "Introduction to unity." *Developing 2D Games with Unity*. Apress, Berkeley, CA, 2019. 13-30.

[8] Halpern, Jared, and Halpern. *Developing 2D Games with Unity*. New York City: Apress, 2019.

[9] Meijers, Alexander. "Unity." *Immersive Office 365*. Apress, Berkeley, CA, 2020. 87-119.

[10] Strauss, Dirk. "Getting to Know Visual Studio 2019." *Getting Started with Visual Studio 2019*. Apress, Berkeley, CA, 2020. 1-60.

[11] Strauss, Dirk. "Working with Visual Studio 2019." *Getting Started with Visual Studio 2019*. Apress, Berkeley, CA, 2020. 61-122.

[12] Strauss, Dirk. "Getting Started with Visual Studio 2019."

[13] Burke, Quinn, and Yasmin B. Kafai. "Decade of game making for learning: From tools to communities." *Handbook of digital games* (2014): 689-709.

[14] Gee, James Paul. "Learning by design: Good video games as learning machines." *E-learning and Digital Media* 2.1 (2005): 5-16.

[15] Bermingham, Susan, et al. "Approaches to collaborative game-making for fostering 21st century skills." *European Conference on Games Based Learning*. Academic Conferences International Limited, 2013.