

Truong Tan Thanh

**DIARY THESIS OF GAME DEVELOPMENT PROCESS IN ONESOFT GAME
STUDIO**

**DIARY THESIS OF GAME DEVELOPMENT PROCESS IN ONESOFT GAME
STUDIO**

Truong Tan Thanh
Thesis
Autumn 2020
Degree Programme in Business
Information Technology
Oulu University of Applied Sciences

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Business Information Technology

Author(s): Truong Tan Thanh

Title of Bachelor's thesis: Diary Thesis of Game Development Process in OneSoft Game Studio

Supervisor(s): Ilkka Mikkonen

Term and year of completion: Autumn 2020

Number of pages: 49

This Diary Thesis presents the process of Game Development in OneSoft game company in 10 weeks. The author is a last-year student from Oulu University of Applied Sciences in Finland. After moving back to Vietnam the author joins the company starting from March 2020.

The objective of this Diary Thesis is to record and analyze the work in Game Development process. To improve in skill and method in programming game, a Diary report is conducted in the work place environment in order to help the company improve and for the author to achieve it.

Before joining the OneSoft game company the author has prior knowledge of Game Development through courses from the University and the practical training in the Game Lab Oulu. Diary type of report was chosen due to the author already working in a company and finds it a suitable subject to discuss.

During the writing of this Thesis, multiple stages of Game Development is displayed and reflected with the material references available. The role, responsibilities and performance of many members in the development team of the game 1945 AirForce is shown, and the weekly theme analysis shows various finding about this process. The OneSoft game company and the author, although facing many challenges in this field of Game Development, has shown significant growth and improvement during this time.

After the weekly reports ended in 13.11.2020, the author continues to work for OneSoft game company expecting the maturity of both this company and individual skill. With the challenges for the growth of the company present, in the future the author will contribute to one of the leading teams in next-gen game development for the Studio and turning those also into opportunities.

Keywords: Diary Thesis, Game Development, Programming, Unity, C#

CONTENTS

1	INTRODUCTION	5
2	DESCRIPTION OF THE CURRENT STATE	6
3	PURPOSE AND OBJECTIVES	7
4	DIARY ENTRIES	8
4.1	Week 1	8
4.2	Week 2	13
4.3	Week 3	15
4.4	Week 4	18
4.5	Week 5	24
4.6	Week 6	27
4.7	Week 7	31
4.8	Week 8	36
4.9	Week 9	39
4.10	Week 10	43
5	REFLECTION	46
	REFERENCES	47

1 INTRODUCTION

Although OneSoft game company is one of the leading company in Vietnam when it comes to Game Development, multiple issues and challenges that are visible for this company to overcome still remains before the ambition of becoming a global success from this company can be realized.

In order to change the focus in the market from simple Shoot-Em-Up type of mobile games to bigger projects that requires time, effort and money of the company, the need to focus on expanding and training employees has become more significant than ever. It is for this reason that the author was recruited into the company and started working there for many months before the record of this Diary Thesis. The author aims to support the company in improvements related to working processes, and it is also very important that the author's skill needs improvement to catch up and match the team, so that the author can be one of the Programmers creating ground work for those new games.

This Diary Thesis was written in order to record the 10 weeks of working here of the author, the procedure and role of each members in the team, and the findings about Game Programming and Development and to reflect it with different researches of developers is underway. In order to help with the improvement of the company and of the author's programming skill, each of the theme for each week represents the different aspects of Game Development to consider.

2 DESCRIPTION OF THE CURRENT STATE

OneSoft game company currently has over 70 employees and 2 employers. The company hired an office of the 7th floor of Kim Hoan building, Duy Tan Street, Cau Giay Province, Ha Noi city in Vietnam. The author's place is in the middle of the company's office therefore it is very convenient for working and communicating with the team. The work place is equipped with PC with average build and 2 monitors. Most of the team's Programmers are very positive and great to work with.

The company has 3 stakeholders: The Director, Vice-Director and the Lead Programmer of team 1945. Their interest is in the development of the company as well as the growth progress of each member in a team they are in charge of.

OneSoft company use Unity platform as their main platform of choice to develop mobile games, and they are entirely focused on mobile. As such, the requirement for applying and working in the company including knowledge and skill of C# language, Unity platform, good programming base knowledge like Design patterns and algorithms; and more importantly, work attitude.

The author's main task upon working here is programming challenges for the game. Most of the time, those challenges include: Enemies, Bosses, common obstacles like Lightning, Rocket, etc. The author's programming job heavily leans toward the logical side of coding practice. Therefore, to successfully carried out the tasks given to the author in this position, the skill of logical thinking and quick problem solving is required.

The author through the process of developing individual skill has become used to the work and can consistently finish them before deadline. For new jobs to be assigned to the author, developing new skillset for it is really needed.

3 PURPOSE AND OBJECTIVES

The learning goal during this 10-week period is to improve in the skill of programming and also being able to understand certain aspects that contribute to the success of a game. The Thesis's weekly analysis each has its own theme, and they are:

The first week: The company background and growth of the business field as a whole. In here the author describe the founding of the company and the explosive growth in South East Asia gaming market. The second week is about the different roles in the game development process and how they interact with each other considering the importance of communication in Game Development as a whole.

The third week describes how the game development procedure is being handled in the 1945 team and the discussion about its effectiveness in the gaming business of OneSoft company as a whole. The fourth week consists the knowledge of different platforms for developing games and why the company chose Unity.

The following week, week 5 focused more on the platform work of the author. The sixth week's theme is about ways to improve skillset as a Programmer in the company, and week 7's theme is about the importance elements during game development process that can largely affect its success.

Week 8 revolved around different ways to monetize the game and increase revenue. Week 9 is about Game Optimization, a heavy topic. And finally, the last week is about the challenges that the company are facing in order to move forward and become a global success.

4 DIARY ENTRIES

This section will be the report of the daily works of the Programmer and the process of Game Development in the team for the game 1945 AirForce. The author's main task will be focused on programming and creating the challenges in the game, namely the enemies, Mini-Bosses and Bosses. At the end of the week will be the analysis and reflection for a discussion theme of that week.

4.1 Week 1

Monday 07/09/2020

The first week of the Diary report is concentrated in making new enemies and challenges for the game 1945 Air Force. In the morning of every first day of the week the code team holds a weekly meeting to discuss the work and assignments for each member for the week, and upcoming plans, deadlines as well as events of the game.

In the late morning the work for Monday begins. The new document for the upcoming 22 enemies, Mini-Bosses and challenges has been sent from the Game Designer to both the code team and the art team to work on. In the meeting the author was tasked with the making of these new enemies for the game, and after communicating through Skype about the design and mechanic of the first 3 enemies the Prefab and basic scripts for them are created.

In the afternoon the art, which is a 2D Spine file of this first enemy called Enemy_Bomber_shoot_2ways was sent to the author. The picture of this Spine can be seen below:

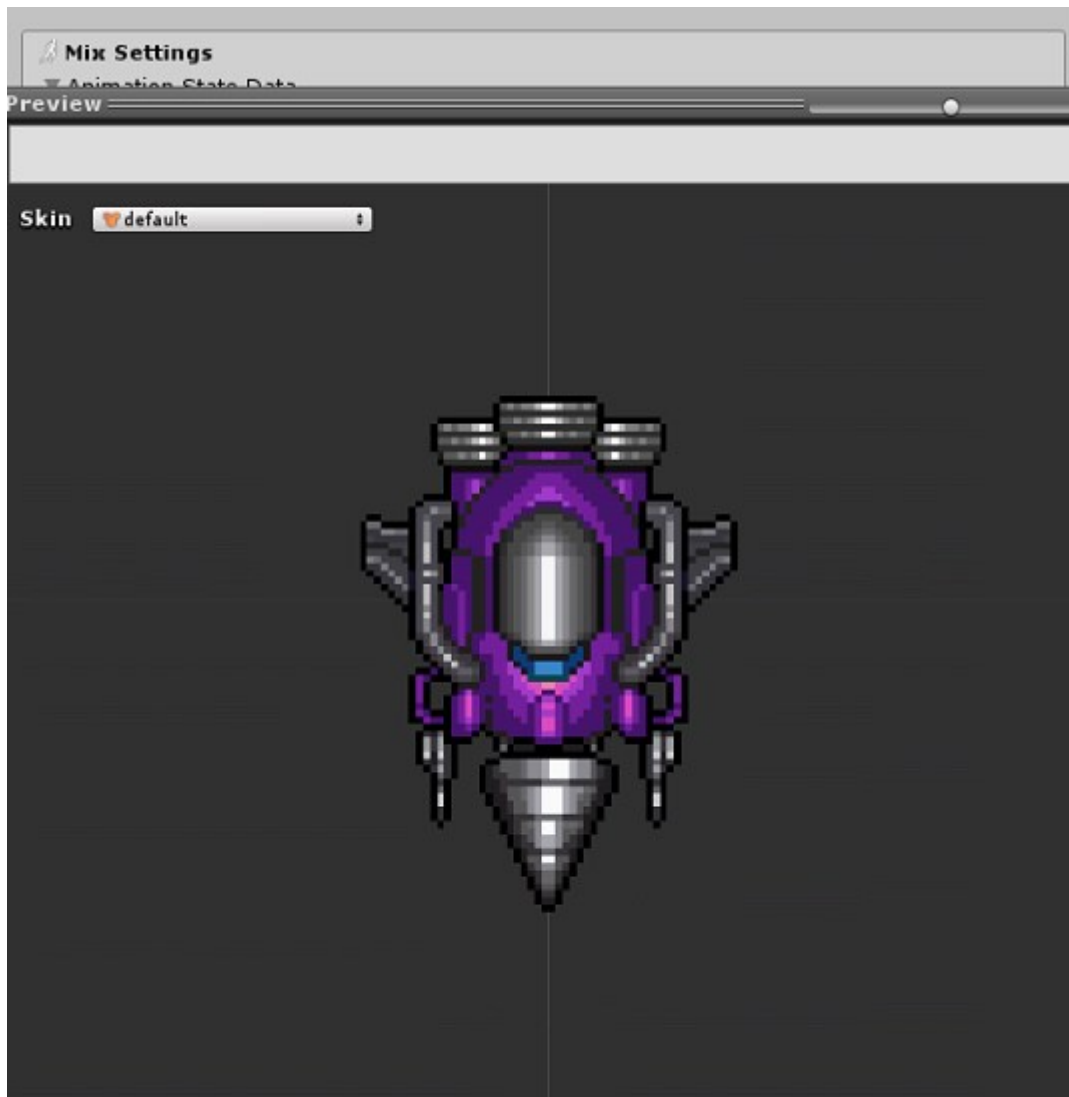


Figure 1: Spine of Enemy_Bomber_Shoot_2Ways

The mechanic for each enemy is always a challenge for the coder, and this time it is quite unique. The afternoon was spent animating the Spine and making the 1st enemy move as intended.

Tuesday 08/09/2020

Today the work continues with the Enemy_Bomber_shoot_2ways, and the real coding practice begins. Most enemies and Bosses in the game attack with a specific shooting system called Uni Bullet Hell (or Ubh). This is an asset on the Unity Asset Store developed by West Hill Apps to create different shot patterns for Unity with the use of scripting components (H. Nishioka, 2020). However, this enemy does not shoot Bullet Hell and will only explode and release out a barrage of bullets when it dies. The problem with this method of coding is that after an enemy is dead, the enemy is

disabled and added to a pool of all the enemy's Prefabs and most of the components related to this enemy, such as different functions in the script will no longer work. So, the Bullet Hell setup for it to shoot out when died will also be disabled.

After consulting with another coder in the afternoon, the author has reached a solution for the problem and will continue with creating the basic script for a new method, which will be implemented the next day.

Wednesday 09/09/2020

In the morning the new script for Enemy_Bomber_shoot_2ways is completed. It solves the previous problem by adding a small delay with Coroutine just before the method Die() is called and the Bullet Hell will be shoot out during that 0.005 second timeframe. Afterward this enemy is finished without further problem and new Prefab for second enemy is added. The new Enemy_Spiral is not very challenging by itself, however the Bullet Hell it shoots out is special and requires personal programming from the coder, since adding Bullet Hell to the enemies is usually the Game Designer's job, based on various existing bullet templates.

The afternoon is reserved to program the logic of this new Needle Bullet, and one of the problems is that Bullet Hell shot out from a normal Ubh template can no longer be controlled. The author will try to get a reference to the bullets through the parent Prefab, and stop the bullet as intended by design.

Thursday 10/09/2020

Today the script for the new Bullet Hell yesterday has been finalized and moved onto the testing process. After accessing the Needle bullet parent and referencing the bullet through it, the author can control this bullet type after it has been shot and further alter the movement of this Bullet Hell.

The new enemy to work on in the afternoon is a normal laser type one. It revolves around a circle before shooting, and this type of movement requires programming logic. All the movement in the game 1945 Air Force is managed by a very famous plugin called DoTween from Demigiant team (Giardini D, 2014). The programmer just needs to use the Tween library in code and most of the movement controls are covered.

Friday 11/09/2020

The Enemy_tower_flow is of a different type compared to other aircraft enemies. This one is a tower on the ground and it will move alongside with the background. Since the map background of the game is several big artwork canvases put on top of each other then use looping function as the canvas goes down, the tower also needs to move with the map. To achieve this there are several special turns in the Scene of Unity reserved for grounded enemies. Enemies that are put in those turn will move along with the artwork while other flying enemies will not.

Week 1 Analysis

The first week analysis will be focused on the company background and growth of the business field as a whole. OneSoft Joint Stock Company is a Vietnamese company that was founded in 2010 (OneSoft, 2020). After 10 years of operation, the company has grown to become one of the leading developer companies for mobile game in Vietnam (itviac.com, 2020). The company also has seen explosive growth in the recent years and now is among the top 10 major Game companies in South East Asia (ABI, 2020) and is riding on the market trend with several published games that are doing well and have potential.

The company currently has more than 250 employees, and is divided into many studios for developing games. The Falcon Game Studio in which the author is working for has the main headquarter in the center of IT focused region in Ha Noi city: Cau Giay. This studio has more than 100 employees and with 5 different development teams working on several games at the same time. The author is placed in the 1945 Air Force development team, the most successful game of the studio in term of revenue (sensor tower).

Due to the sudden success of OneSoft and many other Game companies in Vietnam as a whole these recent years, a research into the mobile market in South East Asia was performed. According to The ASEAN Post team, not only the gaming industry of this region has been blooming but also

the rise of the eSport scene. Below is a figure from The ASEAN Post (The ASEAN Post Team, 2020) on the online and gaming population in the region:



Figure 2: Online and gaming population in South East Asia

In the figure, Vietnam has the second highest gamer population and total online population in South East Asia only behind Indonesia. This is also due to the effect of the growing economies Vietnam currently has and population. As the income grow, it can be spent more on hobbies and activities such as video games. Mobile games are also on the rise due to the ease of localize compared to computer games.

4.2 Week 2

Monday 14/09/2020

Over the past days, creating Prefab template for the new enemies and mini bosses has become easier. One of the accurate examples for the claim is the time it takes to go through each of the preparation process. In the past, the author took approximately two days to prepare for a new opposing element for the game, the process since then has been dropped down to one and half of a day, marking the significant growth in game programming ability. On Monday after the weekly meeting, the work starts with a flame thrower type of enemy. Enemy_fire_loop pushes the use of Coroutine of the author to a higher level, since there are multiple logics for this enemies being delayed after one another. The author also has contacted the VFX Creator to make a new flame visual effect for this enemy.

Tuesday 15/09/2020

Today the making of enemy fire_loop continues. The flame that the VFX Creator worked on has been implemented, and the workload consists of changing the actual timer on the enemy and the hidden bullet to match that of the new effect. After that the process enters the testing phase for this enemy to make sure it goes up and down as intended by the designer. The changes of timing for this enemy has to be worked on from both the script and the editor sides.

Wednesday 16/09/2020

Today the main focus is to finish up testing the new fire_loop enemy and to start with a Mini-Boss. The morning is reserved for testing and meeting purposes, since there is a new event about to be rolled out for the game. In the afternoon the work for Miniboss_5Mini starts. The Mini-Boss requires two guns and a main body to be programmed individually then combined together. At the end of the day, the work for the main body has been completed.

Thursday 17/09/2020

The work continues with the Miniboss_5Mini started yesterday. After animating the main body avatar, the bones of the Spine need to be placed with Game Object point so that the guns of Mini-

Boss can spawn on it. Afterward, the programming for the boss consists of making the main body run and control the side guns through it.

All Bosses and Mini-Bosses use the special interfaces that another Programmer in the team designed. The two interfaces consist of basic methods that most bosses need such as the method to set health, method to call the small guns and method to set the different phases. One of the main different that makes the Bosses require more effort and time than normal enemies is the multiple phases and mechanics each one has. Miniboss_5Mini has two phases that switch once it has been brought down to half of the health pool, and the mechanics as well as the Bullet Hell of the Boss will change.

Friday 18/09/2020

In the morning the matter with Phase 2 of Miniboss_5Mini has been concluded, and the Boss is handed to the test team. The work for another two Mini-Bosses started right away with the new asset received from the artists. The Miniboss_Stalker has an unique mechanic of phasing in and out of the battlefield, and posed an exciting new task for the developer.

The afternoon is spent to create the ground work for the boss, and to discuss and address the desirable phasing animation with the Animator. The Spine of this boss then consists of several phasing stages so that the Programmer can run through them in the method and it would achieve the special visual effect of a boss turning invisible every few seconds.

Week 2 Analysis

With the event that happened for this second week, the discussion theme will be focused on the different roles in the game development process. There are five major roles in the studio's team of the author, they are: Programmer (Coder), Artist, Game Designer, Animator, and the Lead Manager. Each of the role has its own responsibilities and works, however all positions still need to combine with each other as the main focus of the work process is to improve the game and release new contents.

First, the role of the Game Designer, they are the decider of how the game would be played and would move forward as a whole. The Game Designer will prepare documentations about the

relevant aspects of the game that they would like to focus on, and hand it down to other positions to carry on with the task. Second, the Artist will draw the needed art assets for the game like new enemies, bosses, UI, icons, ... and send the Spine and pixel files to the code team. As for the arts of the enemies and bosses, those have to go through the Animator first so he can work on the Spine and create animation for the arts that were drawn.

The Programmers (Coders) are the ones that carry on with the material given and putting all the designs into the game so that the contents work as intended by the Game Designer. Lastly, the Lead Manager will try to pinpoint the general strategy and development plan for the game, and oversee the tasks that are being prepared by other members of the team.

For many Game companies in Vietnam, the development team's number of people varied greatly between projects and studios, from a few people team that handle casual games to larger group of hundreds of people working on quality games. However, the roles inside a development team for most of the companies in this industry remain the same (Bates 2004, p. 151). To handle different tasks of a game cycle, there will always be at least Manager, Designer, Artist and Programmer roles.

4.3 Week 3

Monday 21/09/2020

The usual meeting is being held on the morning to discuss the work left for the month and deadlines for each member in the Programming team. In the late morning the work continues with the Miniboss_Stalker from Friday. After animating the invisibility mechanic of the Boss, the programming work consists of matching the Collider of the Boss with the correct state, using the Bullet Hell only when the boss appears, and creating another phase for it.

Phase two of this Boss will have the Collider turning removed and the Boss will start to move on a loop path setup by the Programmer. The Boss will switch to the second phase once it reaches 50% of the health pool and this is achieved with the Event setup on the Unity Editor, since the game is using the SkyGameKit framework written by a veteran Programmer in the company.

Tuesday 22/09/2020

Yesterday the main programming tasks for the Miniboss_Stalker has been finished so today the author will pass the Boss to the Game Designer for the implementation of Bullet Hell. The author also has notified the VFX Creator of special effect work needed for this Boss and two other enemies.

In the afternoon after the Game Designer has completed the Boss and ran it through a level, the Boss then has been moved to the testing phase and will be handed to the Test team. At the same time the Artist team has finished the Spine of Enemy_MiniSaw and sent the file to the Programmer. The new assignment started with this enemy and will continue to the next day.

Wednesday 23/09/2020

The work to create Prefabs for Enemy_MiniSaw begins. The challenging work for this enemy lies in the special Bullet Hell it has. It requires individual programming and is also one of the most challenging task so far. This enemy shoots out a special small bullet saw that can bounce upon impact with the Camera edge. Since the edge of the Camera does not have special property, the saw's path is first calculated with right angle on the edge. However, this method proves to be quite inefficient since the algorithm is hard to execute. Furthermore, since the saw's path needs to be calculated in real time for every single frame, it will cause unnecessary garbage in the memory system. The task for today is to try out different methods to achieve this mechanic through coding.

Thursday 24/09/2020

During the morning the author has reached a solution for the problem posed yesterday and made a new script for the Saw bullet of the Enemy_MiniSaw and tried out the new method. Instead of checking the position of the saw every frame, the new script gets the position of all four edges of the Camera and only check when the saw has past that value. The new script then checks which edge among the four did the saw bullet came in contact with, and calculate the bounce's angle appropriately. In the afternoon the Enemy_MiniSaw is finished and handed to the Test team.

Friday 25/09/2020

The next enemy in the final day of this week is another Mini-Boss. The Miniboss_2Heads has two phases and two small enemies circling around it and shoots out Bullet Hell. After creating the Prefabs and animations in the morning, the scripts and the special animation are made in the afternoon. This Mini-Boss has a special circling animation to properly display the two enemies' movement flying around it. The special animation is a part in the main Boss' Spine and it has bone positions to put the small enemies.

Week 3 Analysis

The main theme of this week will be focused on how the game development procedure is being handled in the 1945 team and the discussion about its effectiveness in the gaming business of OneSoft company as a whole.

First the discussion will be revolved around the work progression for making of a new challenge. The first job is of the Designer: He will think up the main theme and function of the new enemy for a day, afterward he starts to create a guideline for it, a design document. The document will mention in details the enemy way of working, the strength, the weakness and special notes about it.

The design document then gets passed to the Programmer and the Artist. The Artist team will assign one person to draw the enemy or challenge based on the concept design, and the Animator will receive the final artwork for enemies and animate them. The Programming team will receive both the design document and the Spine art that has been animated, and start working on programming the logic and put everything together. The Programmer in charge will communicate regularly with the Game Designer to make sure the mechanics are made as intended, and to make changes and improvement according to the Designer if necessary. The final product will then be passed on to the Test team and usually will be focused on for one week, before finally being updated into the game so the users can play with new features.

This type of progression for a new challenge has defined the role of each member in the development team clearly and allow rooms for continuous producing of new contents. However, since formal software development methods are usually overlooked (Bethke 2003, p.4), this also

pose a problem for the team, as with no prototype of the new contents the game might encounter unexpected bugs, interactions or behaviors.

The state of the game 1945 Air Force due to this development procedure is also heavily relied on the plan of the Game Designer. It is essential that the Designers must have an open mind (Bates 2004, xxi), communicate clearly and make proper use of documentation for the game, so that their vision and plan are shared with other team members. The author also has found a small solution for this, that is to discuss regularly with the Designer about the mechanics of enemies and Bosses.

Nevertheless, this type of development procedure has proven to be effective in the speed of making and creating new contents for the game, and there is also several testing phases between to make sure of the contents' state before updating those to the main game in the market.

4.4 Week 4

In this and the following weeks, the focus of the coding work will be shifted from small enemies and Mini-Bosses to new normal Bosses. For this type of challenge, the programming work is much heavier and the duration of making a new Boss can cost around one and a half to two weeks, without the testing phase.

Monday 28/09/2020

The Programmers meeting in the morning is longer than usual with the discussion of the upcoming events for the end of the year 2020. With the season of Christmas and the Black Friday, the game needs a big update to attract customers and gain more revenue. The author's work is concluded to be heaviest as there need to be around 10 new Bosses ready for the week before the Christmas season start, which is at 15/12/2020, and the author will have to handle at least 6 or 7 new Bosses. A big update of the game always requires heavy testing to avoid possible bugs and unwanted interaction so all works must be completed one week before the actual update. In the afternoon, the work for the Miniboss_2Heads continues to the scripting part. Later during the afternoon, the Mini-Boss has been moved to the testing phase.

Tuesday 29/09/2020

Today marks the first day of making a new big Boss for the game. The first one that the author needs to create is called BossHugeFlyingBoat. This Boss is actually an improvisation version of a Mini-Boss, since after completion of that Mini-Boss the Designer decided that this challenge is more fitting to be upgraded to a normal Boss. Such improvisation and re-use of assets is very common in the Game Development process, since it can save much time and money for the company, while also makes sure each of the asset is thoroughly used to not waste the effort of creating it.

This Boss has 3 phases in total and it will transform to match each phase. Below are 2 pictures describing the Boss with 2 different transformations for different phases:

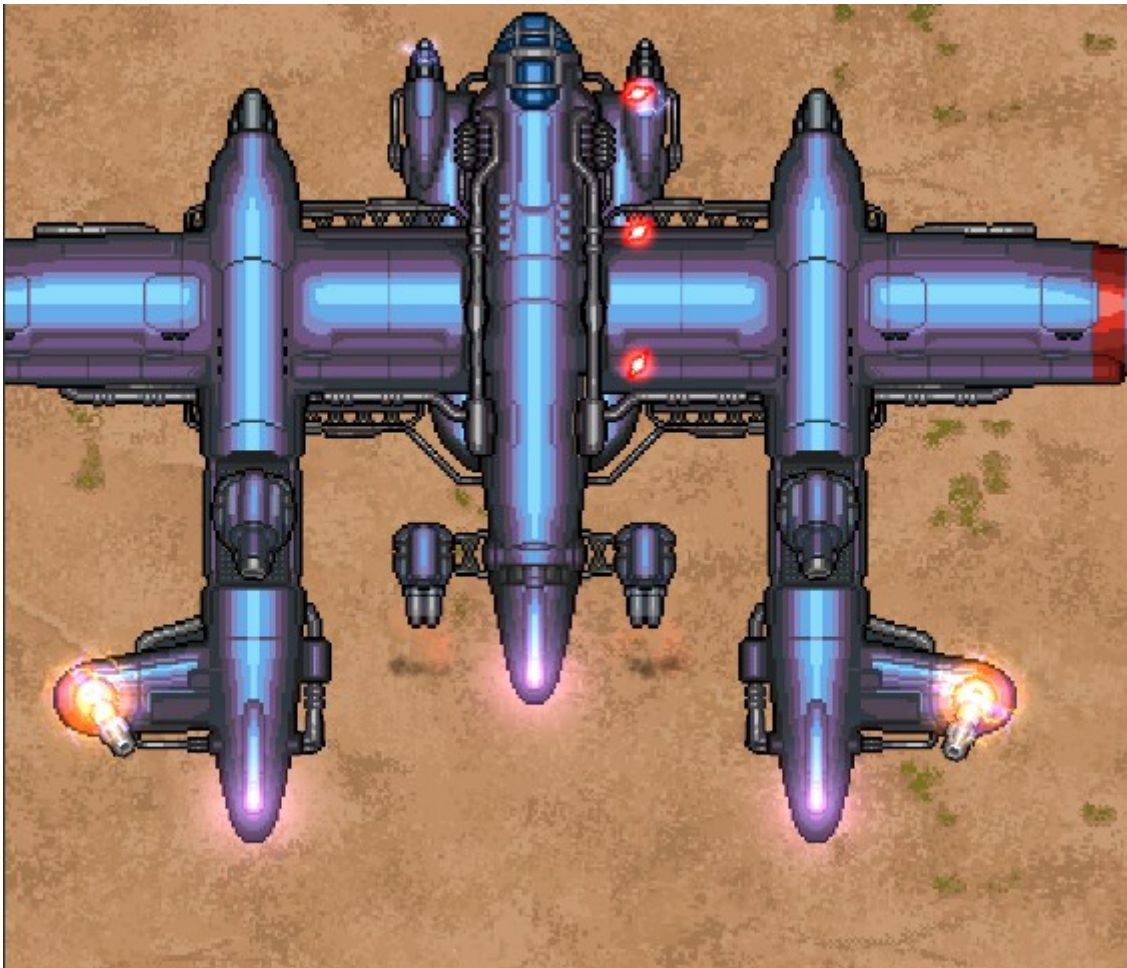


Figure 3: BossHugeFlyingBoat Phase 1

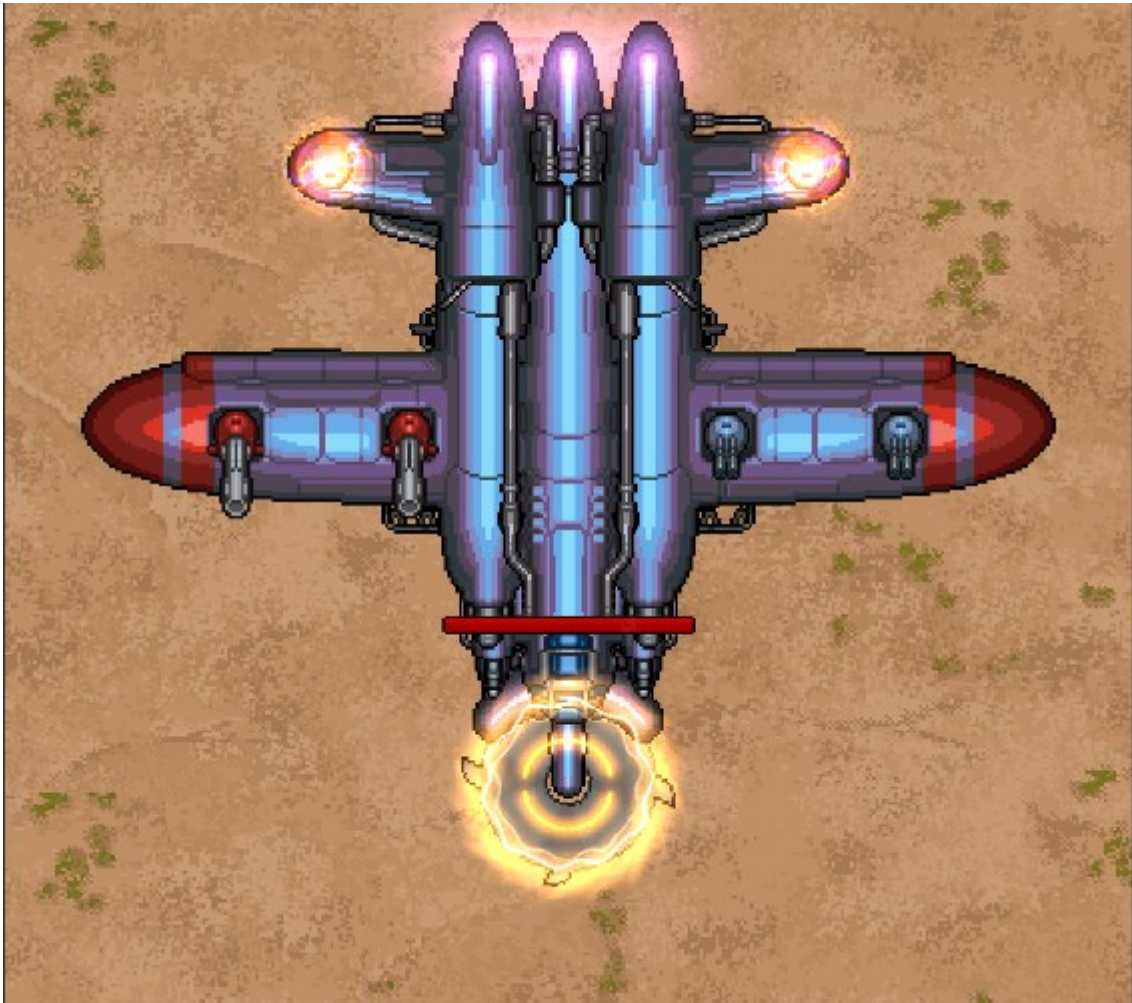


Figure 4: BossHugeFlyingBoat Phase 3

Wednesday 30/09/2020

After creating the basic Prefabs for main body and small Guns of the Boss yesterday, today the author moves to the scripting part. This Boss poses a new challenge for the Programmer's work: The small Guns from each phase will not die as normal but now they will revive after a period of time since they got blown up. The phase only ends if the main Guns are destroyed, it will also blow up any remaining small Guns for good. First the work needs to be done so that the small Guns will revive after some times, but then next job is to also make sure those Guns can still die properly after the main Guns are destroyed. Lastly there has to be a connection between the small and main Guns through reference in code, so that the small Guns will know when the main ones are gone.

The author has thought of implementing a spawn point for small Guns that will spawn a new Gun to replace the spot, however the new Guns cannot be controlled anymore after its creation, since

it is no longer in the initial list of enemies (because it is additionally created), and so connection could not be made between the Gun types.

Thursday 01/10/2020

On the second day of programming the Boss, the author has reached the conclusion for the problem earlier. The small Guns' Spine Object will only be disabled after dead and not removed, so those Guns maintain the connection they have with the main body of the Boss and can be controlled. The small Guns still have the method Die() to use in case they need to be blown up for real. The script in the Boss' main body has become too long due to the amount of code that the respawn loop of small Guns take; therefore, the author has moved them to a separate script and make a reference for it to the main body's script. The picture below will provide the method used to loop the respawn process of the small Guns:

```
//respawn gun1 left right
public IEnumerator LoopRespawnGun1Left(ChildTurnA childTurnA, bool isLv14 = false)
{
    if (boss.GetBossState() != BossHugeFlyingBoat.BossState.Phase1)
    {
        yield return null;
    }

    BossHugeFlyingBoatGun1 gun1;

    WaitForSeconds waitForRespawn = new WaitForSeconds(timeWaitRespawn);
    WaitForFixedUpdate waitForFixed = new WaitForFixedUpdate();

    for (int i = 0; i < childTurnA.EnemyList.Count; i++)
    {
        gun1 = (BossHugeFlyingBoatGun1)childTurnA.EnemyList[i];
        bool isWaitingToRespawnGun1 = false;
        while (boss.GetBossState() == BossHugeFlyingBoat.BossState.Phase1)
        {
            if (gun1.CurrentHP <= 0 && !isWaitingToRespawnGun1)
            {
                isWaitingToRespawnGun1 = true;
                animController.PlayAnimTailCapLeftClose();
                yield return waitForRespawn;
                gun1.Respawn();
                //check if 1 gun3 has died to switch
                if (CheckChangePhaseGunBossPhase1() && !isLv14)
                {
                    gun1.ChangeState(2);
                    gun1.LockOnTarget(1);
                    gun1.EnableColliderWithDelay(1, true);
                }
                else
                {
                    gun1.ChangeState(1);
                    animController.PlayAnimTailCapLeftOpen();
                    isWaitingToRespawnGun1 = false;
                }
            }
            yield return waitForFixed;
        }
    }
}
```

Figure 5: Coroutine used for loop respawn process of BossHugeFlyingBoat's Guns

Friday 02/10/2020

Today the author will finalize the first version of the Boss. The morning is spent to write the methods for changing phases and the condition to move to a new phase of the Boss. In the afternoon, the author has handed the Boss to the Game Designer for the implementation of Bullet Hell and further improvement.

Week 4 Analysis

This week the main theme will be about the 2 most famous platforms for developing games, as well as the advantages and disadvantages of them nowadays: Unity and Unreal Engine. The topic for debating these 2 platforms has been a hot one for a long time.

During the time in Finland, the author noticed that most of the company there use Unreal Engine as their main platform to conduct game development, and only some company, namely FingerSoft still utilizing both platforms (FingerSoft, 2020). However, this is not the case for game companies in Vietnam, where the author is currently working. Most of the game companies here chose to use Unity to develop their games and many developers don't have sufficient knowledge of Unreal Engine.

So what are the similarities between both platforms? Both are very capable of producing the biggest game hits on the market, and they have great bridges between most of the industry standard software. Both have a great variety of tools from their toolbox of technology, like: physic simulation, animation, terrain editor, advanced lighting and VR support. They are also very similar in the creation of next-gen graphic (PontyPants, 2020).

Let have a look at the chart of notable Steam games' market share (Gajsek D, 2020):

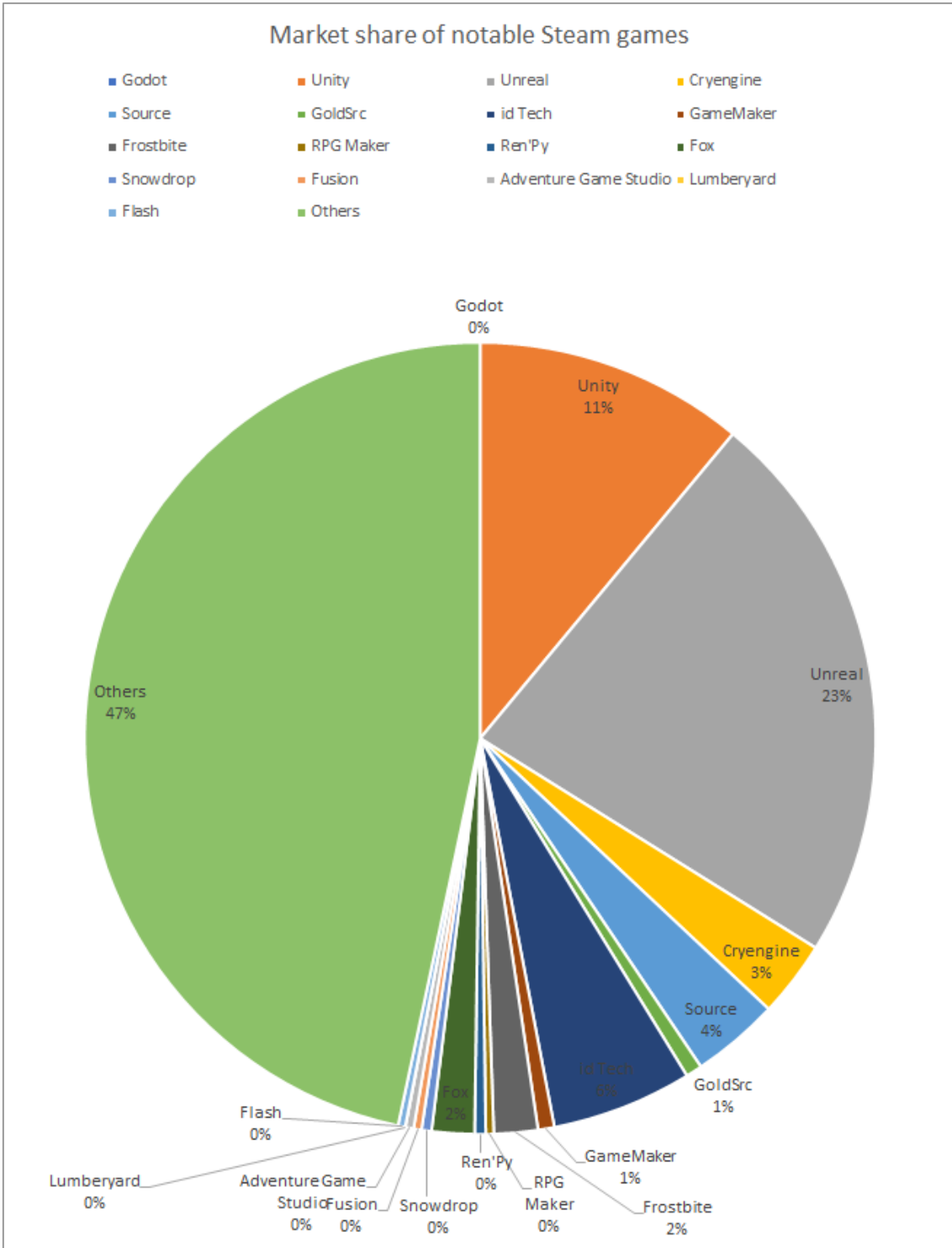


Figure 6: Market share of notable Steam games

As shown above, Unity and Unreal are the leading platforms for game development and together they take up a third of the chart. You can also notice that Unreal Engine is ahead of Unity in this field (VR development) and this will be one of the different between 2 platforms.

The biggest different of the 2 is that Unity is more suited to create mobile and indie games, while Unreal Engine is better at making cutting-edge level of graphic games. Unity using C# to code is also much easier to learn and use than Unreal Engine using C++. Unity is better at making 2D games, while Unreal Engine handle 3D games very well (PontyPants, 2020).

In the end, each engine has its own unique traits and both can be used for free with personal version, so it is the matter of how the company utilizes those traits when picking platform.

4.5 Week 5

Monday 05/10/2020

During the meeting today, the author reported the condition for BossHugeFlyingBoat, and is considered meeting the deadline as the Boss first version has been delivered to the Game Designer last Friday. The work switches momentarily to testing the new levels, while waiting for the feedback from the Designer.

Tuesday 06/10/2020

Today the Designer has sent his feedback and changes need to be made on the Boss. The communication through Skype for this type of exchange is not sufficient, therefore the author has come to the Designer's workspace and discusses the matter with him directly.

In the afternoon the author starts to implement the changes needed. Some of the changes require more time since those alter the function of that section of the Boss completely. Sometimes it is a waste to change the mechanic like this, since the programming effort and time went to that section will be discarded, however the Designer only decided to do so if the matter is truly significant.

Wednesday 07/10/2020

The author continues to work on the changes for the Boss and also comes in contact with the Animator for some of those changes like: The big saw part on top of the Boss' head needs to be

twice as long, so that it can pose a threat to the players. The speed of the animation for the wing part also needs to be slower, so that players can see it clearly. After receiving new Spine file from the Animator, the author will change the code to match the new animations.

Thursday 08/10/2020

The work of changing the mechanics has finished, and the Boss is given to the Designer once again so he can review those changes. With many Bosses, since those all have many unique mechanics and challenges, this process of changing and reviewing for more improvement is quite frequent between the members in the team.

The pre-testing work for a new Boss is also the Programmer and Designer's job. They have to make sure the Boss in its finished state is in good condition and can run properly before moving it to the test team for further examination and deeper analysis.

Friday 09/10/2020

The Designer has concluded the Boss to be finished, and the last job for it begins. In the 1945 AirForce game, there is 1 Boss for every 10 levels, and the Boss will appear in level 4, 7 and 10. To make the proper balance design of increasing difficulty for the game, the Boss in level 7 will have fewer functions than the one in level 10 (in other word, weaker), and the same with level 4. So the Programmer will need to create 3 waves (a wave in the game is a set of challenges for the players, it can be multiple types of enemies attacking or a Boss) of BossHugeFlyingBoat and modifies the code as well as the editor for each wave to have less mechanic than the last.

Week 5 Analysis

The discussion for this week will be more in depth about how the author are utilizing Unity as a workspace: test Scene, main Scene, Events and turns provided by this platform.

First let have a look at the author's workspace in Unity:

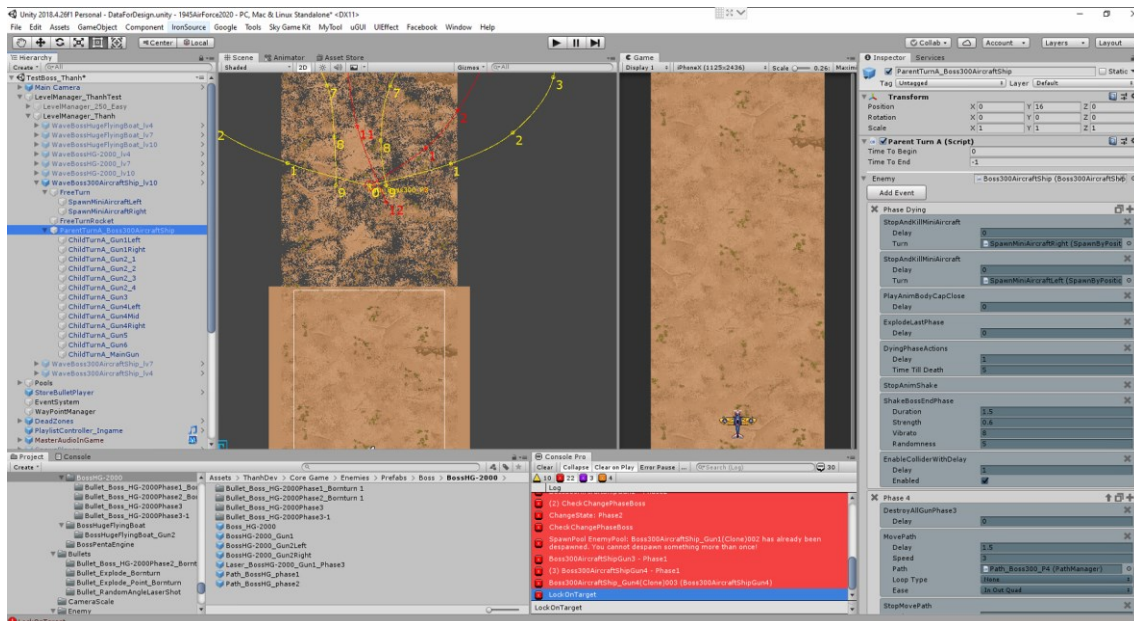


Figure 7: The author's workspace in Unity

Each of the Programmer in the team 1945 has his own folder in Unity and a unique branch in SourceTree version control tool (Atlassian, 2020). All the work for programming and creating contents will mostly be inside that Programmer's folder. For creating Bosses, enemies and challenges, the workspace of the author in Unity has been modified to serve this specific purpose.

The left side contains the Scenes that the author has setup mainly for testing and displaying Bosses, and below that is the folder of the author. There are 2 Scenes set up for Boss development purpose: the main Scene with all the normal functions and waves and player's Aircraft for testing and creating the wave of the Boss, and the experimenting Scene where the author creates the Bosses and Guns' Prefabs and building up new mechanics. There is also a third Scene needed so that other Scenes can run along with it called DataForDesign, which was initially made for the Designer to test the game in various Scenes.

In the middle, the author divided the main visual screen into two part: The Scene screen display part and the Game screen display part. Since while observing the Boss' behavior, the author needs to take note of both the in-game look and in Scene look with various changeable information. Right below the main screen, the red part is Console Pro, an asset from Unity Store made by FlyingWorm (FlyingWorm, 2020) which shows the information needed for debugging the Boss better.

On the right side is the part where the author places the Events and all of the methods written for the Boss that need to be called on Event will be used here. This time of workspace customization has proven to be the most effective to the author's work, and the reference as well as instruction for customizing can be found in the Unity Official Documentation (Unity Technology, 2020).

4.6 Week 6

This week the work with the early phase of the BossHG-2000 begins. During the time there is also a Coding competition in the company with money as winner's prize so the author also practices for it and competes with other Programmers in the middle of the week.

Monday 12/10/2020

The team meeting today is brief as the assignment's deadline for each member is still in effect. Afterward the morning is spent to examine the mechanic of the new Boss: BossHG-2000 and tries to come up with a programming plan.

In the afternoon, the author has come in contact with the VFX Creator and asked for the full effects of the old BossHugeFlyingBoat. The matter also has been confirmed by the Designer on the little details for the effects and how should they appear on the Boss.

Tuesday 13/10/2020

The morning is spent to write a script to control the animation of the Spine from the BossHG-2000. The VFX Creator has sent all Fx effects for the old Boss he worked on to the Programmer later in the morning, so the work has been shifted to applying those effects to make the Boss looks more visually pleasing.

In the afternoon the Programming competition for employees in the company (mainly Programmers) starts. It is held on CodeSignal platform, a website for Programmers to practice and develop their different skills in programming (CodeSignal, 2020). The author also participated however was not ranked highly in the leaderboard. Nevertheless, the amount of questions the

programmer completed was more than the previous competition, marking the improvement in problem solving skill.

Wednesday 14/10/2020

The BossHG-2000 has unique mechanic that changes the normal way of programming for a Boss. This is a satellite with Guns that cannot be destroyed and multiple hit points (HP) bars. Normal Bosses only have one HP bar, and the way to defeat them usually is to destroy all the Guns, then the Collider is active (turning Collider on allows players to interact with that Collider, in this case, the Boss' HP bar can be decreased) and the players can kill the Boss.

To make it so the Boss has multiple HP bar and players can deplete them without the Boss dying, the Programmer needs to write a special method for this. When the Boss' HP bar reaches 0, it will call the Die() method, so first Programmer needs to check which phase this Boss is in currently at that moment, to know whether the Boss should be exploded. If this search of phase returns another phase but not the last one, the Programmer will prevent the Boss' death by overriding the Die() method and replace it with another method that set the Boss' HP bar back to full. This code is demonstrated in this picture below:

```
//check HP
public override long CurrentHP
{
    get
    {
        return _currentHP;
    }
    set
    {
        _currentHP = value;
        if (State != EnemyState.InPool)
        {
            UpdateUIBarHp(_currentHP, startHP);
            if (_currentHP <= 0)
            {
                if (_currentState == BossState.Phase3)
                {
                    Die(EnemyKilledBy.Player);
                    StopLoopLightning();
                }
                else
                {
                    CheckChangePhase();
                }
            }
            EventCheckHPEnemy((float)_currentHP / startHP);
        }
    }
}
```

Figure 8: Script of BossHG-2000 with condition for the HP bar

Thursday 15/10/2020

The scripting for BossHG-2000 continues. In the morning the author comes in contact with the Animator to order new sprites for the HP bar, since this Boss has multiple HP bars, new ones need to be made and the Art team is in charge of this job.

The work for the Boss then moves onto programming the small Guns. For this Boss since players cannot destroy the small Guns and only the Boss itself, the coding for those enemies also needs to be changed. The Guns this time will have a special state called Closed state in the code, and will be changed to this state whenever the Boss changes phase, since in the design document the Boss will close all its Guns when the phase changing begins.

Friday 16/10/2020

The Art team has finished those new HP bars and sent them to the Programmer. These then get combined into a Sprite Atlas, used to contain many Texture files to call upon at will during runtime (Unity Technologies, 2020). The image displaying this Sprite Atlas combination can be seen below:

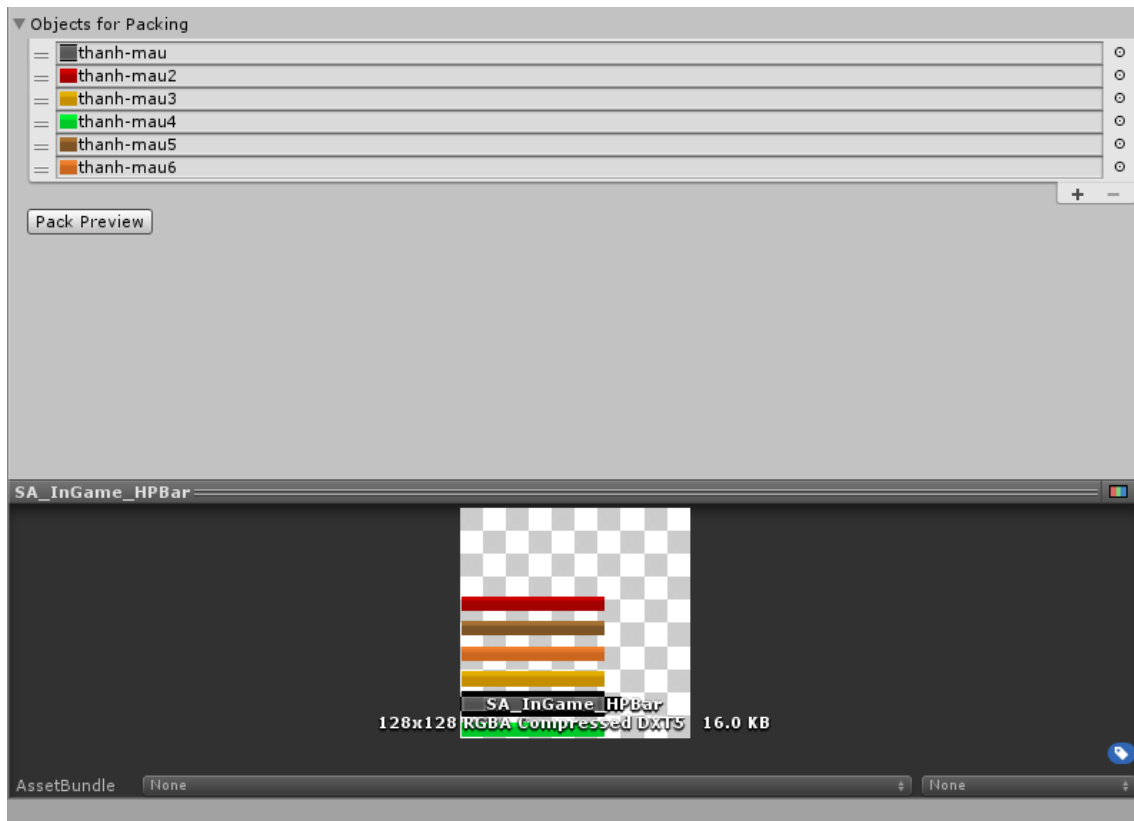


Figure 9: Sprite Atlas for HP bars

In the afternoon, the Boss' initial state is finished, and after working on a special laser for this Boss, the Programmer sends BossHG-2000 to Designer for first state's testing and review.

Week 6 Analysis

This week's theme will be about the ways to improve skillset as a Programmer in the OneSoft company. Since the beginning when the author joined the company in March, the skill and knowledge required to be a successful Programmer of the author was limited; however, the author has improved steadily in both programming skill and communication skill while working with OneSoft company in the 1945 team.

One of the fastest ways to improve as a Programmer that the author has found is to ask other veteran Programmers for guidance. Most people in the work place are willing to help and they usually have an open-minded and positive characteristic, therefore there is no faster way to learn than asking and discussing about the problems encountered during work.

The second way to improve in programming is to analyze and learn the code of one's own project. Especially since the 1945 AirForce game that the author is working for has been released for 2 years, there are a lot of coding practices and methods, algorithms and classes that can be learnt simply by digging up the old code. This type of learning also requires patience and hard work, since reading and understanding thousands of code lines is very challenging.

The third way that has helped the author improve is through the meeting every week of the Programming team. The Lead Programmer arranges those and the team always make full use of the time with planning, discussion and brainstorming problems. Through this the author has developed his communicating and social skill.

The final way that working in the company has helped the author improve is with many coding competitions and events that OneSoft company organized. Coding competitions are usually held on CodeSignal platform, and the company also held a big GameJam on June for 3 days to improve game developing skills of all employees. GameJam is a global theme of game creation event for a limited time and has gain recognition from game developers all over the world (GameJam, 2020) and so it is a great event to meet people, create games, and have fun.

4.7 Week 7

Monday 19/10/2020

The meeting this week has been moved to Wednesday due to the absent of the Lead Programmer in the team. And the focus in the morning of the author is on testing the next set of new levels while waiting for the Designer.

In the afternoon, the Designer has sent his review and the Boss needs to be heavily modified, due to the visual implementation of the programming does not match what he has envisioned. Luckily this Boss does not have many Guns because of its mechanic, so most of the changes only need to be on its main body. The author starts by removing the flame Bullet Hell that has been implemented previously on the middle antenna.

Tuesday 20/10/2020

Today the author needs to work on a new type of laser that the Designer required for this Boss, and the programming of this new laser has proved to be a major challenge, since the algorithm used to make the function is quite complicated. The ReflectiveLaser requires that the laser will reflect to a certain angle upon impact with the Camera edge, and this has to be done in real time since the Boss will shoot it for a while, so the laser's angle needs to be calculated continuously without making the game too heavy. The author spent Tuesday to try to come up with a solution for the reflection.

Wednesday 21/10/2020

The work for today revolves around experimenting with the solutions for the new laser. The author in the morning also contacts the VFX Creator for the effects on the satellite's wings, and in the same time contacts the Animator to ask him to redo the wings to match its different colors in the design document.

In the afternoon the effects ordered are finished and it really fleshes out the Boss' visual after implementation. The image below shows the wings of the Boss with all the effects on:



Figure 10: BossHG-2000 with full effects

Thursday 22/10/2020

Today the author needed to ask for help with the Laser from other Programmers, and worked on the matter with the assist of another. The method used to achieve the special continuously reflecting trait is complicated and will not be discussed in this report, however it does solve the problem and the work afterward is to utilize this mechanic and create the right angles so that it poses a challenge for the players.

Friday 23/10/2020

The Boss is delivered to the Designer in the morning and the testing process begins. However, in the afternoon after reviewing the Boss, Designer has discussed with the Programmer and concluded that the laser needs to be on 2 sides for better balancing and visual. The Prefab of this ReflectiveLaser then is duplicated into 2 separate controllable Prefabs by the Programmer so that the Designer can easily alter them. Below is the demonstration of the 2 new ReflectiveLaser:

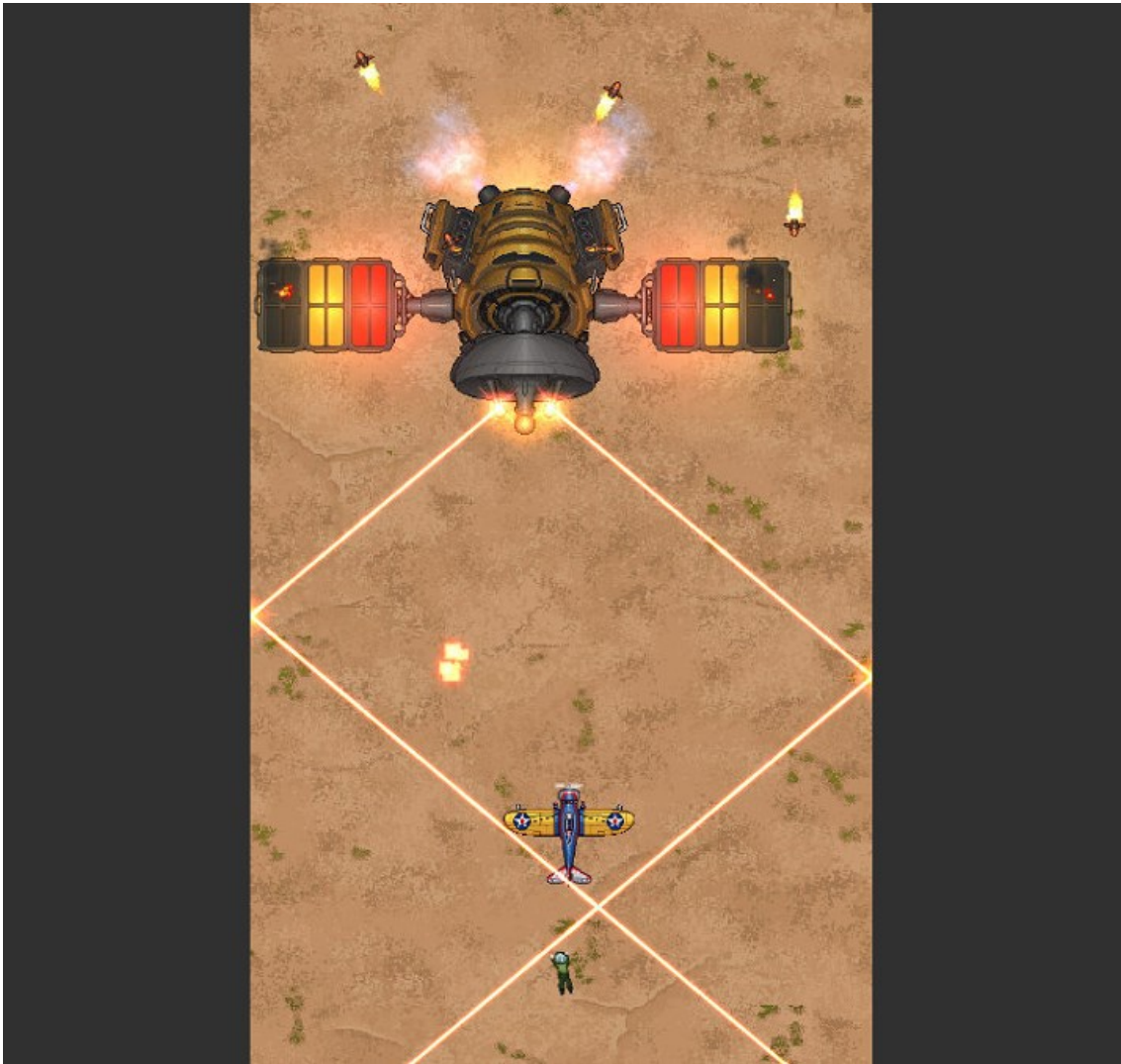


Figure 11: The ReflectiveLaser in real time

After the alteration of the laser is completed, the Programmer will divide this Boss' wave into level 4, 7, and 10 with decreasing difficulty like before. On wave 7 the Boss will run away after the second phase, and on wave 4 the Boss will not have additional phase for those small Guns on top of that.

Week 7 Analysis

This week analysis is about the importance elements during game development process that can largely affect its success.

Game development is a complex process, and different key choice and work in this process can really decide the quality and successfulness of a game (Aleem S., Capretz L.F. and Ahmed F, 925).

The key factors that can affect a game's success are, but not limited to: Team management, Game design document, Game arts and assets, Game programming practices and Game test management. Among those the Game programming practices and Game test management are the 2 factors that the author is contributing to.

For Team management factor, it is critical that the process of administration and cooperation between member in the team goes on smoothly during development timeline, since game development requires collaboration work a lot from different party members. The Lead Manager also needs to confirm the deadlines and makes sure of the job each member is working on in the team.

The Game design document is also one of the important factors in determining good game development plan. Basically, the design document includes the objective, the description and the goal of targets for development, whether it is Boss, enemy, Event or Aircraft.

Most things that contribute to the visual aspect of the game are considered arts and assets for that game. Llopis (Llopis, 2004) stated that "game assets include everything that is not code: models, texture, materials, sound, animations, cinematics, etc." Many games succeeded largely due to the assets those possess, and those assets enable developers to build responsive, interactive and attractive games.

Good programming practices is probably among the most important factors that contribute to the game's success, since bad coding can be very detrimental to the game. A programming team with the necessary skills is considered as the backbone of the game development process (Aleem S., Capretz L.F. and Ahmed F, 934).

Finally, the Game test management is highly valued as well, due to it being closely related to programming phase. Most of the bugs or unwanted interactions will be found during this phase and should be dealt with immediately since accidentally updating those to the game will leave consequences. The management part of the test team therefore needs to be handled properly.

With those key factors above, the conclusion is that most of them benefits the game's successfulness and should not be treated lightly, especially the Game design document, good programming practices and Game test management factors.

4.8 Week 8

These final 3 weeks will be focused on making the Boss300AircraftShip. This Boss is the hardest one for the Programmer to work on by far and so 3 weeks are dedicated to it. It needs to be difficult and fun at a large scale, due to the upcoming events for the game because these events will attract a lot of customers. It is also one of the big Bosses that the team only makes once every 50 levels.

Monday 26/10/2020

A new meeting on Monday is held to re-evaluate the works done so far and set new deadline for the Boss300. The Programmer then received the design document for this Boss: Boss300AircraftShip. It is a big battleship in the sky and the scale of this one is several times larger than other Bosses. Players will have to fight many sections of the Boss: the tail, each of its sides, and the head to finally defeat this challenge.

Since the Artists are still working on this Boss, the author can begin preparation stage first by creating new Prefabs and script beforehand, using any temporary Spine. The real Spines for the Boss can be implemented later when those are ready.

Tuesday 27/10/2020

Today the basic script for all the Guns of the Boss is made first. Since all child Guns have many similarities, the Programmer can create a script template for them beforehand. They will share most of the same basic methods and variables unless there is a special case. Methods like HP, checking method to change the phase of the Boss, basic Bullet Hell, and lock on function.

Wednesday 28/10/2020

In the morning the author has received all the Spines for Boss300 from the Animator, and begins to replace them in the Prefabs created. This Boss has 6 groups of Guns and can spawn small enemies to aid it in battle. The work for the script begins with the first phase, where players fight the Boss' tail. Gun 1 has been described to shoot out heart-shaped bullets which is quite tricky to do, however Gun 2 takes most of the time of the Programmer.

Gun 2 is a group of 4 small Guns moving according to specific patterns, and those patterns loop themselves continuously. The movement of those Guns are quite unique for each pattern so there is no solution to simplify the coding with a common method. The author needs to write enough methods of movement control for all Guns and for all patterns.

Thursday 29/10/2020

Today the author focuses on creating the pathing system for Boss300, since this Boss' movement is also tricky by itself. The Boss is too big however it needs to turn to the left, right or upside down every time it switches phases, so to imitate real Aircraft's movement, the Paths using Simple Waypoint System from Rebound Games (Rebound Games, 2020) need to demonstrate the action of turning while moving, and the Boss will travel outside of the Camera screen for a bit before moving back in. The image below will show one of the Path created for this and the Boss' movement while traveling on it:

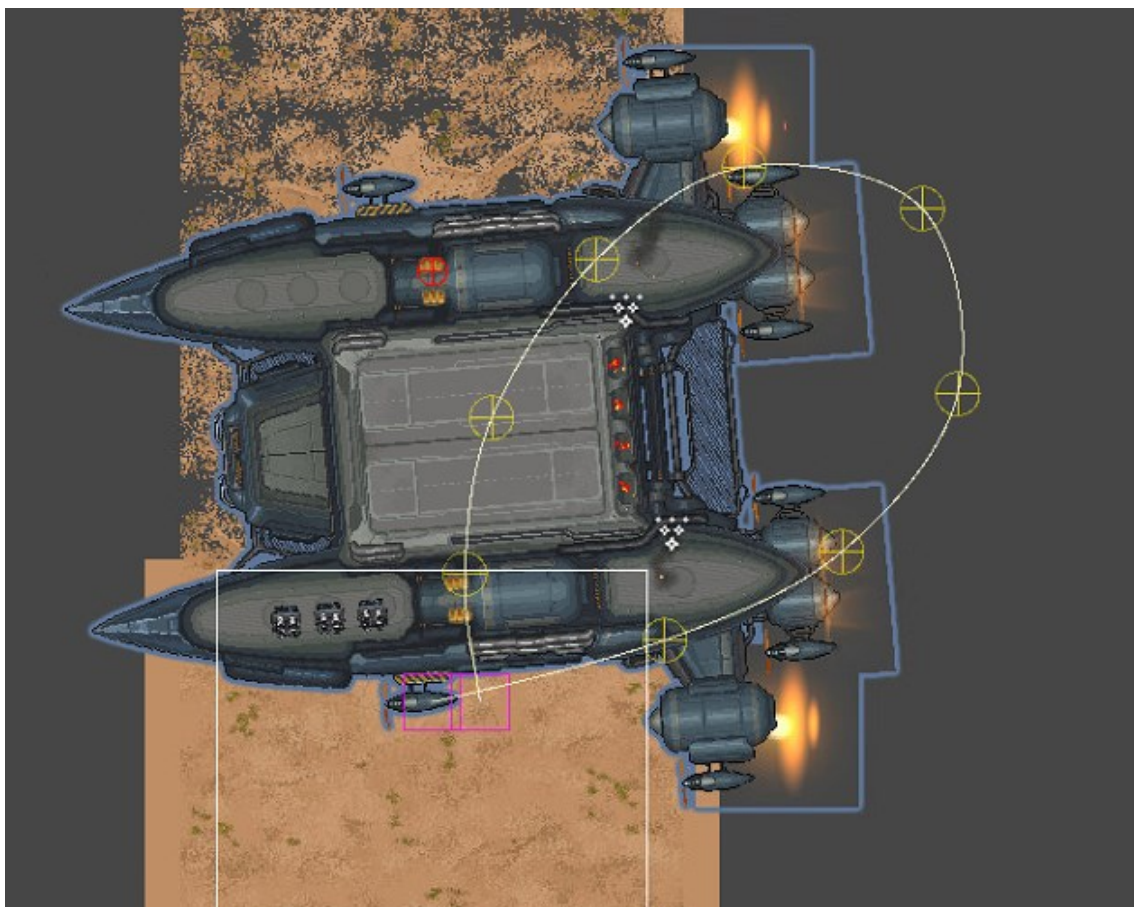


Figure 12: Boss300AircraftShip traveling through custom Path

The image also demonstrated the size of this Boss, as those white lines are representing the edges of Camera.

Friday 30/10/2020

The Paths for the Boss to move on has been completed yesterday and implemented, so today the scripting part moves on to phase 2 of the Boss. This phase has the same mechanic of small group of Guns reviving itself with the BossHugeFlyingBoat, so the author can re-use the code made to control the reviving process. The Bullet Hell for the bigger group of Guns from this phase is also a special type that needs individual programming job: a circling quad bullets system that travels together. After some initial brainstorming, the author decided to ask the Animator to help creating a Spine that has this type of movement, therefore greatly reduced the programming work needed for this Bullet Hell.

Week 8 Analysis

This week analysis will be about possible ways to monetize the game, and the choice of the game 1945 AirForce. The various ways to monetize the game including paid games, In-app purchases, In-game Ads and some other non-game focused monetization strategies.

Paid games method has been the obvious choice for so many companies in the past, and still remains somewhat viable as of today. It is still largely the case for AAA titles. For this type of monetization, to earn decent revenue selling the game at a fixed price, companies need to invest time and money into marketing, and only the best games will be successful with this type and make enough money back to break even or earn more (MDN contributors, 2018).

In-app purchases usually rely on a very big number of players downloads, since the proportion of the player base that spends for In-app purchases is very small. In this way, players can usually get and download the game for free and still have the option to spend and gain bigger advantage in the game. The very important factor that decide if this model can work or not is the strategy of pricing for all In-app purchases. There has to be a balance in what the developers are offering, mainly on the low end and high end of that pricing range.

In-game Ads is the biggest monetization method to use for the game as it has been proven to be very effective for the developers. Players who download the game can still enjoy the game for free, while the developers continue to earn revenue through the Ads. There are however some mentions for using it: Players don't really want to see intrusive ads or ads that are irrelevant, or if they are shown too many ads. All of the drawbacks from this method might really damages user experience (Knezovic A, 2020).

For the game 1945 AirForce, the team is using both In-app purchases with Subscription option and In-game Ads, and this combined method are very effective which leverage the game into the most successful game in OneSoft company in term of revenue.

4.9 Week 9

Monday 02/11/2020

The work of Boss300AircraftShip has to be postponed due to the urgency of another task given to the author during the meeting early morning. As levels of the game increase, the need to prepare for those levels to enter the real game version on market also rises. Our game is using a system called Object pooling in C# which acts as a container for objects that are ready to be re-used (Pal T, 2020). By keeping the objects that appear frequently in the game in a pool, those can be taken out and put back into the game without the need of creating new ones. In other word, Object pooling improves system performance by reusing objects. It is therefore important to run each level through a system called AutoPool before pushing it to the main game. It is a system developed by a Programmer in the company to help with the process of applying Object pooling to the level.

Tuesday 03/11/2020

Today the work for Boss300AircraftShip continues. The Animator has sent the unique Spine for the quad bullets ordered on Friday last week, and in the morning, it is implemented into the game. After showing this new bullet to the Game Designer in the afternoon, he expressed some concerns with the way this bullet moves and expands, and requested the Animator to change those movements

a bit. While waiting for the new Spine, the author has finished with the programming process for Phase 2.

Wednesday 04/11/2020

Phase 3 for this Boss is quite different in that it has 2 groups of small Guns, each group shared a common HP bar represent all Guns in the group, and all small Guns locks on to the player. The second group of Guns shoots out a special type of Bullet that can change shape continuously. This mechanic is done through the DoTween transform control as mentioned before: The author will write the code to utilize method DoScale to make the Bullet change scale property at first, then after some delay reverses the process so it can become smaller overtime.

Thursday 05/11/2020

Today the work moves on to the final phase of this Boss. Phase 4 has a big main Gun in the middle with the HP bar that represent the entire health of the Boss, and player will win when they destroy that main Gun. It has one unique Bullet Hell that requires coding and 5 normal Bullet Hell that can be implemented by the Game Designer. The author also needs to code in the functionalities for different Gun' s phases during this final phase. The main Gun will shoot out one random Bullet Hell among the 5 prepared, and change them every 15 seconds. If the health is above 50%, it will shoot out the unique Bullet Hell, called LaserBullet every 7.5 seconds, and every 5 seconds if the health of the Gun is below 50%. This is achieved with the use of Coroutine and Delay method in coding.

In the afternoon the author codes in the special LaserBullet with the help of another Programmer in the team. Since in Ubh template there is no pattern of Bullet fire matches this type of LaserBullet, the Programmer first needs to stop this Bullet from moving according to a template, and then controls its path manually with code. At the end of the path it will shoot out laser at specific angle. The final result of this unique LaserBullet is shown below:



Figure 13: LaserBullet of Boss300AircraftShip

Friday 06/11/2020

In Phase 4 of this Boss there is also a last mechanic: It has an airway for the small enemies called MiniAircraft to fly out of. These enemies are not children of the main Boss therefore those cannot be controlled after it has been instantiated. The author has placed them into a path so that they will move along that path after spawning. In the afternoon the first version of Boss300AircraftShip is transferred to the Game Designer for evaluation.

Week 9 Analysis

This week the topic of discussion is Game Programming Optimization in the 1945 AirForce game, which is very important in the Game Development process. Coding optimization will affect heavily to the smoothness of gameplay and the number of new features a Programmer can additionally implement (Krauss K, 2020).

One of the most common programming patterns that is heavily used in the process of programming the game 1945 has been mentioned in the Diary: Object pooling. The purpose of using Object pooling is to reduce the effect of creating new and complicated objects to the application performance. Especially in the game 1945, the use of duplicate assets is frequent and many of those GameObjects are expensive in performance cost to call. This leads to a big problem of game stuttering and crashing which affects user's experience badly. After storing those objects into a pool with AutoPool, and pre-load them during the loading screen before the new level, the performance of the game has been improved greatly and this change also got reflected in revenue.

The second most popular programming optimization method for the game 1945 is the inheritance of classes in coding. So in other word, the implementation of Subclass Sandbox (Nystrom R, 2020). By creating a base class with all of the overlapping methods as needed, the subclass that inherited it will also have all the methods available without the extra duplicate, or redundant coding. The author's enemy new classes are always subclasses of the original main class, and therefore inherited all of the necessary basic components. The author only needs to tailor the programming process for each individual enemy that has unique traits with extra methods or override existing methods from base class. This will not only save time and effort but also reduce the amount of methods from each object that the game needs to run at any given time, thus increase the performance.

The third is utilizing good coding practice especially concerning Update() method and looping in Coroutine. Most of the time the Programmers in the team do not use any method in the Update() at all, since the Update() method will be called every frame, the methods that are put into it will also be called every frame and this is a huge blow to application performance. To avoid this most of the methods are written separately and only get called when needed. For Coroutine, if a while loop is used inside it the Coroutine will also not end as long as the condition for that while loop is still fulfilled. Therefore, it is important to check carefully and end the loop when done.

Understanding the Game performance issues and what cause those is the crucial step toward good Game Optimization, and the game 1945 AirForce is doing so very well to keep the performance flow and avoid crashes. This is already achieved when the author joins in because the team has many veteran Programmers that are working on this game and similar game type for years, however like all of the optimization process, there are always rooms for improvement.

4.10 Week 10

Monday 09/11/2020

During the usual meeting in the morning, the deadline of Boss300AircraftShip has been decided to be this weekend. The author also has communicated with the Game Designer about this information to hasten the process.

In the afternoon, the author has received a list of changes needed for this Boss, and it is more comprehensive than normal one due to this one being the Boss of level 300. All phases of the Boss need to be changed and some mechanic will also be reworked.

Tuesday 10/11/2020

The author starts to implement the changes for first phase today. Gun1 of phase 1 of the Boss will have 3 different kinds of flame shooting and they need to be randomized between shooting normal Bullet Hell and shooting flame. To achieve this the author made 2 Coroutine types: One for the randomization of flame type and the other type is for randomization between shooting Bullet Hell or flame.

Wednesday 11/11/2020

In the morning the author fixes the angles for group of Gun2s at boss' phase 1 to be smaller than before, and it needs to be implemented for all shooting pattern of this gun. In the afternoon, the author noticed a bug about Gun4 not reviving when needed in the second phase of the Boss, and starts working to fix it. In the end of the day the cause for this bug has been found out, and dealt with.

Thursday 12/11/2020

The Game Designer would like to change Gun3 into a single HP bar kind of phase, so the players only need to shoot down one HP bar and the phase will end. This phase has 2 group of Guns: Gun5 and Gun6. To achieve this change, the author will not turn on Collider for Gun6, and put a Shield effect to it so players will know this group of Gun6s cannot be destroyed. Through

programming the author will find and destroy Gun6 group manually when players have depleted HP bar of Gun5, so this phase only have one HP bar left to shoot down.

Friday 13/11/2020

In the morning the author works on implementing changes for the last phase 4: The MiniAircraft will now be moved to be a child of main Boss first, in the launching part, to ensure that it stays on the airway after coming out. Afterward the author will disable this MiniAircraft object in Boss' body and instantiate new MiniAircraft from outer Camera screen then let it flies in. There will also be 2 types of MiniAircraft: the first one follows a path and shoots at player, while the second one cannot shoot but will charge at the player after coming down.

In the afternoon the final work for this Boss is done: Dividing the Boss Wave into Wave level 4, 7 and 10 like before. The Boss is then pre-tested a few times by the Programmer before being transferred to the Game Designer.

Week 10 Analysis

Last week will be about the challenges that OneSoft game company in general and Falcon Game Studio needs to face and how the company deals with them to improve and become an international enterprise with AAA game quality.

The first challenge is about the size of OneSoft company. Although the company has many successes since the founding time 10 years ago, to this day it is still a small to medium size company with about 60 employees. The area for OneSoft company is also only one big office in the 7th floor of Kim Hoan building at Cau Giay Province, Ha Noi. To reach higher the company needs multiple teams working on more games and contribute to the Studio as a whole. OneSoft company has realized this and expanding continuously the past few months, also due to the big success in revenue of the game 1945 AirForce. The company has recruited more employees up to nearly 80 people now and rising, OneSoft also has hired the whole floor instead of just an office and in the procedure of renovating the area into a big working space.

The next challenge is about the skillset of Programmers in the company. To be an international company and has multiple successful games, the company also needs really professional individual

as well as group of developers that can really bring Game production to the next level. As of now the skills of Programmers in the company are only quite average, due to multiple reasons. After recruiting most of the newcomers, including author are junior Programmers that need to be trained and get more experience. Another matter is that the company has never produce a big game title, so most Programmers even veteran ones will lack the skillset and experience for this matter. To face this issue, OneSoft company is holding multiple lessons in Game Programming, Game Optimization, Programming patterns and training. Aside from that, the company also held many coding competitions to improve the logical skill in programming of employees. Lastly, after some of the games in the Studio are stable, the company is having the intention of making bigger games with larger team of competent Programmers to push out onto the global market.

The third challenge that OneSoft company is facing is the attitude of many employees toward changes. Since the company is 10 years old, there are many employees that have been here for a long time working on those games. Some Programmers have become senior in the field however many are just average. The problem with this is that the employees that has worked the same job for one game multiple years have stopped developing their skills, since they are already capable of maintaining their game. However, to grow as a group and to create bigger hit, OneSoft company really needs to move forward onto more complicated and skill-intensive projects. If the employees still keep this mindset they will not have the necessary knowledge to move on, and will drag down the company as a whole. The CEO of OneSoft has addressed this issue during one of the big meeting of the company and asked for everyone's cooperation in moving forward.

So in conclusion, although facing many problems, OneSoft game company is dealing with them one by one in order to achieve the dream of success that many companies always have. When the people equipped with the necessary skills and mindset lead the new development team, the company will surely change for the better and grow.

5 REFLECTION

In conclusion, during the 10 weeks of this report the author has changed both in the knowledge and skill for Game Development. The finding of each week theme has also widen the knowledge of the author and stated the aspects that affect Game Development process, which directly contributes to the company's success.

Through the theme of each week the conclusion for finding can be found and reformulate: In this ever-growing South East Asia market, the company needs to adapt and follow the trend to further develop and makes a name in the game market. To achieve good quality game product, all the roles in a development team have to work smoothly together and good communication between member is the key. The development procedure like describe in week 3 has a lot of potential in making contents for the game fast and efficient, however it may lack the comprehensive testing and planning part and that should be improved. Since the OneSoft company only focus on mobile gaming and market, Unity platform is the correct choice when it comes to development tool. The platform is also very user-friendly to use and setup, so it has helped the author greatly in work after a proper interface is set. The company is also a great place for the author and all of the Programmers to hone their skill, both in programming and communication. There are multiple key factors that affect the game's success and all of them have their benefits and should not be treated lightly; the Game design document, good programming practices and Game test management factors are especially vital in this and should be prioritized. OneSoft company chose the most common ways to monetize the game: through in-app purchases and in-game ads, however to properly increase the revenue without trouble the Game Optimization also needs to be on point, so that crashing and lagging problems in game has to be reduced to a bare minimum in order to compete with other games in the market.

This 10-week period has been a growing experience both in author's individual professional skill in the field and in a group within the company. Although facing many challenges, the author expresses great interest in the future success of OneSoft game company in general, and Falcon Game Studio.

REFERENCES

ABI 2020. WE ARE ABI. Search date 27.09.2020. <https://abigames.com.vn>.

Aleem S., Capretz L.F. and Ahmed F. Critical Success Factors to Improve the Game Development Process from a Developer's Perspective, Journal of Computer Science and Technology, Volume 31, Numbers 5, pp. 925-948, DOI: 10.007/s11390-016-1673-z, Springer, September 2016.

Atlassian 2020. Built for professional teams. Search date 04.03.2020.

https://bitbucket.org/product?&aceid=&adposition=&adgroup=92266810077&campaign=9122400131&creative=414680965637&device=c&keyword=sourcetree&matchtype=e&network=g&placement=&ds_kids=p51241933763&ds_e=GOOGLE&ds_eid=700000001551985&ds_e1=GOOGLE&gclid=Cj0KCQiAhs79BRD0ARIsAC6XpaWPoGRG2oQByQsXjLbgA6z5KhEDxxGtAvxeD8BK3Y10CccW4gRplvwaAldXEALw_wcB&gclid=aw.ds.

Bates, Bob (2004). Game Design (2nd ed.). Thomson Course Technology.

Bethke, Erik (2003). Game development and production. Texas: Wordware Publishing, Inc.

CodeSignal 2020. The Most Advanced Assessment Platform

on the Market. Search date 06.05.2020. <https://codesignal.com>.

FingerSoft 2020. CAREERS. Search date 01.02.2020. <https://fingersoft.com>.

FlyingWorm 2020. Editor Console Pro. Search date 04.03.2020. <http://flyingworm.com/>.

Gajsek, Dejan 2020. Unity vs Unreal Engine for XR Development: Which One Is Better?. Search date 11.10.2020. <https://circuitstream.com/blog/unity-vs-unreal/>.

GameJam 2020. About. Search date 15.10.2020. <https://globalgamejam.org/about>.

Giardini, Daniele 2014. DOTWEEN (HOTWEEN V2). Search date 03.10.2020.
<http://demigiant.com/plugins/dotween/index.php>.

H. Nishioka 2020. Uni Bullet Hell. Search date 02.10.2020. <https://westhillapps.com>.

Knezovic, Andrea 2020. Mobile Game Monetization Trends: Best Strategies to Monetize Your Game in 2020. Search date 18.11.2020. <https://medium.com/udonis/mobile-game-monetization-trends-best-strategies-to-monetize-your-game-in-2020-fe2de42d4463>.

Krauss, Kirk J 2020. Performance Profiling with a Focus. Search date 21.11.2020.
<http://developforperformance.com/PerformanceProfilingWithAFocus.html#FittingTheSituation>

Llopis N. Optimizing the Content Pipeline. Game Developer Magazine, 2004.

MDN contributors 2018. Game monetization. Search date 18.11.2020.
https://developer.mozilla.org/en-US/docs/Games/Publishing_games/Game_monetization.

OneSoft 2020. ONESOFT GAME STUDIO. Search date 25.09.2020. <https://onesoft.com.vn>.

Pal, Tapas 2020. Object Pooling in C#. Search date 20.11.2020.
https://www.codeguru.com/csharp/csharp/cs_misc/object-pooling-in-c.html

PontyPants 2020. Unity VS Unreal – Which Engine Should You Choose?. Search date 11.10.2020. <https://sundaysundae.co/unity-vs-unreal/>.

Rebound Games 2020. Simple Waypoint System. Search date 16.11.2020. <https://rebound-games.com>.

The ASEAN Post Team 2020. The gaming explosion in Southeast Asia. Search date 27.09.2020.
<https://theaseanpost.com/article/gaming-explosion-southeast-asia>.

Unity Technologies 2020. Customizing Your Workspace. Search date 12.10.2020.
<https://docs.unity3d.com/560/Documentation/Manual/CustomizingYourWorkspace.html>.

Unity Technologies 2020. Sprite Atlas. Search date 10.10.2020.
<https://docs.unity3d.com/2019.3/Documentation/Manual/ciSpriteass-SpriteAtlas.html>.