Team\_H3S

Metal Adam

-Team Postmortem-

GAM150S17KR

|  |  |  |
| --- | --- | --- |
| **Name** | **Title** | **Coding Responsibility** |
| **Hun Yang** | **Producer** | **Game Logic** |
| **Soyeon Cho** | **Test Manager** | **Graphic, AI, Tutorial** |
| **Wonhyung Chae** | **Technical Director** | **Sound, Building** |
| **Sehyun Kim** | **Lead Designer** | **Option, Menu** |

Introduction

A 2D side-view tower defense game with mining contents to collect resources. Player should protect humans and help them develop technologies to stand up to robot armies by building attack towers and utility buildings.

Development Team

Team H3S was made at the end of last semester. Hun and Soyeon was looking for two members of their team. Therefore, Hun talked with Wonhyeong and Sehyun to work with them and they accepted his suggestion. At first time we met, we did not know each other well, so we spend much time to assign roles for every member.

Hun took a role as a Producer, because he has some experiences to manage some teams before.

Sehyun took a role as a Technical Director, because he is good at math and physics, and has much passion for coding tasks.

Wonhyeong took a role as a Lead Designer, since he wanted to be a game designer some day and has creative ideas.

Lastly, Soyeon took a role as a Test Manager, since her meticulous nature would help team to find any bugs through her testing game.

* Producer: Hun yang
* Technical Driector: Sehyun Kim
* Lead Designer: Wonhyeong Chae
* Test Manager: Soyeon Cho

Tools

* Warp Engine (Game Engine)
* FMOD (Sound)
* Visual Studio 2015 (Code Editing)
* Paint.NET (Art Assets)

What Went Wrong

1. Unrealistic Goal

* First time we met, we set a big goal, which is that we wanted to submit our game in Busan indie game festival.  
  In order to achieve our goal, we had several idea meetings and studied C++ language in advance to have enough abilities to develop awesome game in the next semester. We made our game in paper version and zero-engine version to test how fun our game is. The volume of our game became bigger and bigger, after additional idea meetings.   
  At that time, we did not worry about how many contents we will develop, because we thought we had enough abilities.  
  However, it was completely misjudgment, because unexpected problems occur and we lose our passion due to overwork.  
  Through team meeting for postmortem, we realized that how bad big goals affect game development.  
  We did not know where we were going and overestimate our abilities, which made us have an obsession about detail game features.  
  What we learned from this semester is that we must be humble and make some signposts that let us figure out the progress of development.

2. Lack of Professional Mind

* Since the middle of this semester, we had some conflicts, such as overworking, personalities, and missing deadlines. We tried several methods to resolve these issues, which is on the Team Management pdf file. At the end, all attempts to resolve conflicts turn out a failure. At that time, we did not have enough room and time to consider resolving issues, because all team members build up stress. Therefore, we chose the easiest way, which is to avoid resolving problems. We believe that if we ignore problems, time will help us to resolve them naturally. However, as time goes on, these problems became bigger and bigger and affected game development seriously.
* After all, all of us acted like unprofessional game developers. We just tried to give up resolving problems and ignored them. We learned we must not avoid problems, when we face them. In case of problems with team members, resolving them must take the precedence of development, which is more helpful in entire development.

3. Different Levels of Coding Skill

* This is the hardest part at the first time we met as a team. Some of the team members had good programming skills, while others did not have. It caused serious problems because some of us used advanced skills compared to other members’ skills. The members that have relatively poor skills could not know what methods did the other use. Therefore, the speed of development was slow at the first time because the members should learn advanced skills first to understand the other's’ codes. Moreover, those who have advanced skills should consider the level of the other and fix at the end to optimize. Both of groups had problems because of different levels of coding skill.
* At the end, most team members improve their programming skills a lot through game project to catch up with other team members. However, there were some serious problems that broke members’ confidences at the first step of development. Therefore, during vacation, we will try to help team members to improve their programming skill and avoid using advanced and complex programming skill in next semester.

4. Fail to Prioritize important tasks

* Since all of our team members are programmers, we did not know how much time we will spend for overall game balancing and overlooked it. We thought many game contents will make players fun. Therefore, our top priorities were always developing many game features as much as possible.
* At that time, we completely forgot fundamental fun of game comes from well-designed game balancing, such as map design and tower’s attack points and so on.
* Therefore, even during final release, all of our team members did not have extra time for game balancing and spent time to develop extra features, such as particle system and game result board. At the end, when we send our final submission, overall game balancing is poor, which make our game boring.
* Now, we realized that we are not here as programmers. We are here as game developers. We must not focus on only programming jobs. We must consider entire game design.

5. Poor Usage of SVN

* At the first time, we tried to use SVN to maintain source files and resource assets. The SVN was confusing at the first time, because we never used it before. Therefore, we depended on our technical director who now how to use SVN. After we assign a role of managing SVN to him, other team members did not try to learn SVN themselves. The technical director had a complaint about his job, and suddenly changed team SVN without notice. He did not try to teach us how to use changed team SVN. Therefore, during final release, our team did not use any team SVN and used physical memory. At final release, it was okay to use physical memory, because we did not change their own codes and just modified some features. However, we also realized that it is time consuming.
* Now, all team members realized that how important SVN is for team project. Therefore, during vacation, we will do mini project to get used to SVN.

What Went Right

1. Constant Task Management and Schedule Management

* Efficient time management and schedule management worked nice during our project. Although we had some trouble and should work hard, thanks to this management, the work was less hard then wort situation.
* We manage our whole task which cut by weekly task at start of week, and evaluate result at end of week. With stand meeting to check progress of work, this method improves our efficiency and lets us work punctually. Despite of lots of tasks we must managing, we can control every part of work and work done, because of this schedule management

2. Design by Everyone

* At the begin of this project, we made lots of decisions by conversation. Especially, designing is main advantage of this thing. While we decided our concept, story and theme, every team member could suggest one’s opinion and we had time to consider about it. The Metal Adam was created by all these steps. It did not only occur when we decide at first. During the project, when we should make new contents or revise old ones, we also made meeting to talk about it. It is constantly work until our work is done.
* For example, the main story and theme of Metal Adam is created by Soyeon cho. Every team member suggest one’s own story, but after we share all opinions, we just chose this one. Detail of Adam that should has effect on in-game was also discussed by all team members. Skill of Adam, which Hun said, was accepted. Other balance parts and system details by Wonhyung were selected.

3 Asking Helps from TA

* During developing our game, we need lots of help to make one’s code work. Unlike other games, we have lots of contents that should consider and control. Therefore, we must manage these things to correctly work. However, we are just fresh man, so lack of our knowledge sometimes become serious problems. At that time, we usually did not waste time. We asked proper help to TA, and professor to get answers what we needed. Accepting one’s weak point and asking helps to improve one’s skill is very important and influential ways in studying. Through this experience, we learn lots of thing.
* For example, FMOD is most important part of this. Wonhyung was not familiar with this kind of work. Therefore, he was very confused at the first time. However, he asked help to TA, and understood how to use and manage the code for this.
* Also, Particle System is other parts of result. Soyeon had hard time to understand how particle work and what is its algorithm, but she finally accepted the steps of use particle by TA’s explanation and make her own system. Not only these things, there were many other parts that were completed.

4 Passion about Game Project

* One of the most important things for game developers is ‘passion’. Without passion towards to their game, nothings get better. In that kind of view, all of team members in ‘TeamH3S’ have great passion to their game and work. From the beginning, we discussed hard about with design of our game, and did many meetings when we brought up with some ideas to make the game funnier. Moreover, we tried to learn more effective skills from TAs, professors and other classmates to make the game be more stabilized and fast.

5 Applying Skills from Class

* There are many things that we could learn from class. When my team learned something new from class, we tried to apply new methods as soon as possible, such as class, linked list, and so on. That was wonderful experience to practice skills, review some materials from classes, and implement in practical way. With this kind of practicing, each of team members could learn more effectively. It is fact that this process took much time than we thought, but all of team members did challenging work to get familiar with that learned materials. Eventually, the my team members’ skills are improved a lot.