IEFP Internship at Miniclip

Pedro Geraldes

Abstract—In this report I describe my experiences while working on my first real job as a software developer at Miniclip. From the new tools I have learned, to dealing with different people and different projects within the company, I was able to gain very important soft skills that will help a lot while trying to grow as a video game developer.

Index Terms—Miniclip, Internship, Meetings, First Job, Mobile, Dude Perfect 2, Scrum

1 Introduction

My dream is to work in the video game industry. I love games and creating games. I decided to submit an application to enter Miniclip because its one of the biggest mobile gaming companies that have offices in Portugal. I though that this was a good way to start my career. My main learning goals for this activity were:

- Have my first professional job experience
- Learn more about how to build games (which is my passion) in the gaming industry
- Deal with very different types of people: Producers, Artists, Server developers, Stakeholders
- Gain more responsibility by having to respect a schedule, keep up with the tasks for the sprints, report to others, making presentations
- Be included in the development of my first "professional" game

I was able to accomplish all of those goals in the internship so far.

• *Pedro Geraldes, nr. 63915, E-mail: pedro.geraldes@tecnico.ulisboa.pt, Instituto Superior Técnico, Universidade de Lisboa.*

Manuscript received July 6, 2015.

2 Recruitment Process

Before Miniclip I had some interviews with other companies, but only with Miniclip I had a group activity. Since we were divided into groups of 3 and needed to talk in English through the activity, I realized the important thing was not building the tower with 10 papers but to see how we would talk with each other. Because of that I tried to be as helpful as possible considering the goal we had in front of us. It was very clear to me the importance of preparing for an interview. Since Miniclip is a game development company, I should have prepared myself for some specific gaming questions.

3 First Job Experience

Since this was my first job, I had a lot to learn. Usually in college the schedule changes and its more flexible, but now I had to be more organized and responsible. In the first few weeks after beginning on the Cocos Builder project, I realized how dependent the whole company was of what I was doing, since this was something all developers had to use to build games. This was a huge difference in responsibility compared to the college projects where we almost always had small groups of students. I felt a sense of opportunity to real make the difference in the way we build our games, and that helped me through the process.

(1.0) Excellent	LEARNINGS						DOCUMENT						
(0.8) Very Good	Context×2	Skills×1	Reflect×4	Summ×.5	Concl×.5	SCORE	Struct ×.25	Ortog×.25	Exec×4	Form ×.25	Titles ×.5	File ×.5	SCORE
(0.6) Good	0.9	0.9	0.9	1.0	0.6		1.0	1.0	1.0	1.0	1.0	1.0	
(0.4) Fair													
(0.2) Weak													

4 Meetings

I learn a lot from the meetings we had in the company so far. In the developers weekly meeting for example, I had to talk to everyone about my work, and how I solve problems. I had to talk in English to an audience of more than 15 people and perform several presentations. This allowed me to improve my speech skills.

I also learn the importance of being aware of what everyone says on meetings, since it is the only way to know how other projects are doing in a more technical way. Sometimes, the problem I'm trying to solve, was already solved by others. These meetings allowed me to understand how the development pipeline works at Miniclip.

I have learned to speak and ask questions when needed and not being afraid to say mistakes in front of colleges. At the beginning I was a little shy to do this and felt a little difficulty to understand some of the things discussed in those meetings. But in time, I have gained confidence and improved a lot at this area.

5 Dealing with People

One of the biggest experiences that allowed me to grow as a person and as an engineer was definitely dealing with people with different types of roles inside the company. From producers, artists, game designers, to other developers, I had to adapt in order to understand what to do in each situation. For example, during my work at Dude Perfect game, I had to follow the instruction of the producer and game designer of the game. Although I could give my opinion on the subjects, I had to learn to understand that my job was to program, and the decisions of how the game should be were to be made by them. This is something really important in a job (knowing my space). Of course I asked questions when I didn't understand some decisions, but it is important to maintain respect and to understand that each and every person has their job in the process of making the game. This is a huge difference in what usually we do in college projects, where all the students in the group have the same role in it (at least most of the times).

From the experience of working with different kinds of professionals, I can say that a video game must consist of these elements:

Producer - Have the final word on everything. Deals with stakeholders. Prioritize tasks. Schedule deadlines. Artists - Build the art for the game (obviously). This includes images to use in the game, but also for marketing, promotions, trailers, etc. Game Designer - Design the game and asks the artists to create art for it. Server Programmers - Program the server side of the application. Client Programmers (where I'm included) - Program the client side of the application. QA testers - Test the game and report bugs and crashes.

This organization implies that I must not do the work of artists by trying to change something in an image. Instead, I should ask them to do it if I need to. This allows me to concentrate only on my tasks and be more productive.

6 Projects and tools

Of course I had already used scrum in college projects but never within a project with so many people. Now I understand better the importance of scrum (or other agile methodologies) in computer science projects. Without it, it would be very hard to organize what to do and to see what others are doing, as well as understanding if the project is on track according to the deadlines. The estimations are really important to understand the teams speed to better realize how much time the game will take. This is actually used to see if the team needed more people or not to finish the project on time.

7 Dude Perfect 2 Development

It was only after starting working on Dude Perfect game, that I have learned the most about building games in a more professional environment. I had the opportunity to be included in the project and also to experience its release process, which was very important for me.

One of the things that I have never understand very well in the scrum process is how

we can create a testable product easily for the QA team to test. At Miniclip we use Jenkins, which is a software that runs on the server and organizes builds. Over the course of the Dude Perfect project I was able to understand how the process of creating builds for QA works. Jenkins runs a script that compiles the project that is in the version control and puts the binary accessible for download. The QA team simply have to download the game from Jenkins and run it directly on their devices.

Another important thing I have learned is configuration. Since releasing a game in the app store is a process that takes around 10 days, what we do is to put all resources (images, animations, levels, etc) in the game, but also on our servers. When the game starts, if it has a connection to internet, it downloads this information from the server and replaces with its local resources. This allows us to make some changes to the game without having to submit a new update. We can change the assets from the game in our servers, and when the users play, the game will automatically updates if it has an internet connection.

8 Dude Perfect 2 Release

Besides all those important technical aspects that I have learned through the development of Dude Perfect, I have also experienced the whole process of dealing with the submission of the game. Our soft launch was on May 21, but was rejected by apple a few days after. They said the game crashed on launch on the iPad Air 2. Our producer had to talk to them in order to better understand what was happening. It turned out it was a mistake by them and not by us. The game was accepted for submission only one week after it was first submitted because of this. This is why its important to have a soft launch before a global release. If the problem was real, we still had time to fix it. Also, once its released, being on soft launch, allows for us to better analyze the behavior of the players (through analytics software) and to have a first notion of how the game will perform in terms of profit. In addition we can see the comments of users and the crashes/bugs reported by them.

9 Conclusion

With this activity I consider to have learn a lot of important soft skills, such as:

- Sense of responsability
- Improved speech quality in presentations
- Understanding of the whole recruitment process for a job
- Dealing better with people with different roles inside the company
- Better understand the way a software product is planned and produced

I am sure that all these learnings and experiences were crucial for me to grow as a person and as an engineer.