1. Introduction

Hi my name is Joeri and I'm a grad student at the Utrecht School of the Arts faculty of Art Media and Technology. I study Game design & Development, specialize in game design yet at moment of writing am doing a Quality Assurance(QA) internship at Vanguard Games. Needless to say things didn't quite work out as I expected.

Through this report I hope to give some insight to those interested in a QA internship at a game developer and offer some useful advice to designers like me having a tough time breaking in.

2. How I ended up at Vanguard Games:

As said in the introduction I specialize in game design. To be more specific I fulfilled the role of project lead and game designer during most of the projects during my studies. Unfortunately neither of these is a frequent intern position at a video game company.

In advance I'd like to say that I don't at all recommend the process I went through to find my internship. It was really messy, full of doubts, and chaotic. A lot of times I was completely clueless what I was going to do next. Having said that I would like to share the real story so the following is how I found my internship.

Like any other student I knew that I was eventually expected to go look for an internship. And like any other student I aimed my sights high at first then lowered them as the internship approached. At first I hardly paid attention to the Dutch industry and was mostly concerned with studio's abroad. My image of the Dutch industry was that it was small, specifically excelled at casual and applied games and knew only 3 triple AAA studios. Finding an interesting intern position at one of these studios was going to be hard and to be honest neither of those really connected with what truly interested me in terms of game design.

While the internships one can take are varied I think the following three types encompass most of them:

1. An internship in a team of developers. You'll be taking care of all the work that's left over.
2. Working at a non-gaming company, as someone that knows about games and create a project for them. This company usually knows very little about game development but is interested in doing something with games.
3. Working at a small developer, you're given a bit more responsibility and influence into the games development. Usually at small studio's (mainly startups)

So I first looked at an internship abroad. I however wasn't really sure what I wanted and what I wanted out of an internship. I know I wanted to influence the game concepts and I have an interest in exploring new, experimental, concepts that challenge existing conventions of the medium. Looking at that I found two recent developments interesting:

1. The experimental indie development (which was rising quickly at that time but wasn't as established as it is now)
2. Developers like Quantic Dream and for example the smaller ThatGameCompany. These focused on providing a less conventional game experience but with triple AAA means.

Apart from what developments had my interest I also had some practical intentions.

1. I wanted to find a company that was an established studio where I could learn the ropes from experienced professionals.
2. I wanted to see how a traditional studio operated, observe if it matched my initial assumptions and learn from it.

With my sight still aimed at abroad I also oriented myself on the options available to me in my own country during the 1st year of my study. I attended the Global Game Jam, the Summer Game Dev hosted by the Dutch Game Garden and Festival of Games to name a few. I quickly discovered that the traditional Dutch game industry is small and the areas that the Dutch industry was quickly growing in weren't interesting me as much. In that sense my initial impressions of the industry were correct. Finding a game design internship at a studio that suited me was going to be highly sought for and scarce.

The high standards asked in the AAA openings were very intimidating and to be frank paralyzing.

Portfolio:
Probably the hardest obstacle I had while looking for an internship was the lack of a good portfolio. I only had one finished near the start of the initial internship period. The lack of a good portfolio ruined some pretty good chances for me at a good internship. Each developer I met eventually asked for it and I didn't have anything to show yet or didn't have anything to link them to.

This was the most apparent for me when I managed to talk with Peter Molyneux directly at a game convention in Amersfoort. He replied that I was to send a letter proving him of my passion for games. I would then receive a test.
I started to work on the letter but I couldn't put to word what my true passion was. It was all very confusing for me, in the end it paralyzed me and I never send anything out in the end.

Creating a good design portfolio was quite a puzzle for me and to be frank I still don't think the current one really shows who I am. Since I strive to communicate as efficiently and quickly as possible my work is mostly communicated through quick sketches on paper or very quick text documents. Not at all impressive when you upload them on your site.

Eventually I did get a good representational portfolio online and I have always gotten good feedback on it from developers. It's possible that you might take the portfolio more serious then your employer though and while I do recommend a good representational portfolio you may find that developers pay more attention to other traits that you initially would think are less important then the portfolio it self.

By the time my portfolio was finished the 2nd intern period was about to start. I send out emails but I got little promising responses. Most companies had already chosen their interns. I did get one interview at Paladin studios but didn't get the position in the end. I was then left with the following options:

1. Find an internship at a company that I didn't initially think of.
2. Spend this period on retakes and do the internship after my grad year.

Since I still had retakes to take care of and didn't feel like applying at developers that I wasn't interested in I opted for the 2nd.

When summer came I had learned more about Unity3D. Positions were opening up at Ronimo Games and Vanguard Games. I applied at both and was invited for an interview at both. Ronimo opted for someone more level design focused and Vanguard in the end didn't have a design position to grant me. They did inform me that they were looking for QA but I wasn't looking for that at the time.

So after all this time the main feedback I got was: "You seem to be good at what you

do but were just not looking for that at this time". This kind of feedback can be quite depressing even though it isn't really negative.

One year later I had done my graduation project and one other school project. I had added them to my portfolio and time was running out for an internship. I send out mails to the usual and got some more responses but still hardly any invites. The reply's I got were standard but nothing promising.
I was getting stressed and was looking into less conventional options. I then received an invite from Vanguard for an interview. I packed all the portfolio content I could. I was all prepared to show my design growth over time but the interview was for a QA position. There was little interest in my design work since it wasn't what the position required. But it was "A" position. So I took it this time.

To qualify for the position I think the following convinced them:

1. Experience with game development
2. Experience with mods (I did QA for a mod team of Super Smash Bros Brawl)
3. Designer, so creative at solving problem. Must be able to creatively find bugs.
4. Could possibly assist design but not as full position.

All of these can be summed up in the most essential desired trait I found:

"Must be able to contribute in a meaningful way to the company in a way that will justify the €300,- a month"

So all I can advice to fellow students is to take every opportunity you can prior to your internship to learn about game development, participate in mods and find as many ways as possible that your can contribute. The position you'll find is still going to be somewhat of a gamble though but if you can convince them that you'll be useful you've won half the battle.

About a week later I received the confirmation that I got the position and when I could start. "As soon as possible" I of course said.

3. Profile of Intern Company

While it might have been a bumpy road and I applied for a game design internship I eventually found a very satisfying QA internship at Vanguard Games.

Vanguard Games is the result of a merger between W-Games(known for My Horse and Me and Greedcorp) and Karami games.

It is a rising AAA studio located on het Damrak in Amsterdam about 5 min walk from Amsterdam Central Station. It counts about 50 employees.
Vanguard produces triple AAA quality downloadable titles for platforms like XBLA, PSN and mobile devices. Vanguard Games most recent game released is Gattling Gears. Unfortunately I signed a NDA (Non Disclosure Agreement) when I started the internship and am thus unable to disclose any information about current projects, this includes target market etc.

To be fair it also isn't very relevant for the internship so the rest of the report will focus on the QA position itself.

4. About the internship

So how did my internship actually go? As said above I can't go into specifics about the projects I worked on. Therefore I will only go into the process and procedures of the projects I worked on. It shouldn't matter much since most of my work wasn't game specific.

I started my internship late July/ early August. Since I was asked to start on short notice I was gracefully granted one day off till the end of august to work on game projects I was still involved in. After that I worked full time till the end of my internship with the exception of about 2,5 months of visiting the physiotherapist one morning in the week (due to a early state of Carpal syndrome).

I was hired to do QA for Vanguards Internal Toolset. Vanguard has a division made up out of 5 people that form the engine team. They have developed their own engine and toolset for the rest of the company to use.
The toolset had to be developed quickly in the past two years. As a result it was quickly filling with bugs, stability issues and crashes. Initially one of the senior designers did QA for the toolset. But this was proofing too much for his workload. Hence the intern position.

For those unfamiliar with the term, QA stands for Quality Assurance and is mostly seen as the division that verifies if the product is functioning as expected. QA positions can differ. Linguistics testing is often also seen as QA but is most often outsourced to specialized companies.

In this case, QA for the internal toolset encompassed the following things:

1. Crash the toolset as much as possible
2. Test new features before they are integrated in the production work branch.
3. Report reproduction cases that point to the crash or bug in mantis
4. Run a large test protocol over the toolset before a new version of the toolset is released to the rest of the company
5. Maintain the bug list on mantis, close resolved bugs and reproduce new reported bugs

I didn't start doing everything at once of course but it wasn't long before the above was my full responsibility. In addition I got some left over design tasks including building the gameplay for one of the leftover levels for one of their games.

Since the internship required me learning a lot of technical stuff at a rapid pace I found while writing this report that it's important to explain some of the tools and procedures we used to prevent any confusion during this report.

The engine toolset:
Vanguard has its own engine toolset. This is made up of the following editors:

1. A Game Object Editor
2. A Level Editor
3. Particle Editor
4. Lightmapping editor
5. Sound Object editor

Like Unity3D designers can create game objects with enabled or disabled components and specify specific parameters. Unlike Unity these components aren't added to a game object but instead activated or disabled. Based on the project needs components are either added or removed to the editor by some one of the game or tech team.

The most fancy component is the states and rules component. In this component designers can create state machines without coding. They create states and rules, the rules than run actions based on conditions or events (for example when ''state1' is

entered' 'display a message'). "'State1' entered" would be an event.

Events are triggered when something special occurs (a timer expires for example). Conditions are checked every frame ('distance to player is x' for example).

This way designers can build most of the easier game functionality reducing the game programmers workload so they can focus on the harder to implement features.

Filesharing through a subversion system
Vanguard uses Perforce 'a subversion' system that allows them to share folders and files over a network. They use build servers to build stable builds (green builds). Using perforce, different employers can add or check in changes to files. Files they want to change are checked out. Two people can change these files at the same time but these changes must contain no conflicting changes before they can be submitted and are accepted by the build server. If a build fails (conflicts are found) the build shows up red. This way conflicting changes show up early preventing a lot of bug fixing in a later phase of development.

The engine toolset uses 3 branches (directories).

1. The Production branch (used by everyone working on the same project)
2. The PreRelease branch (used by the tech team to integrate features to the production branch prior to a release)
3. The main branch (a outdated branch in terms of game code but fairly up to date with the latest engine code, this is a test environment to test new features before they cause any big problems in the other branches)

The tech team only makes high priority changes to the Production branch. Other changes are marked as 'fixed in next release'.

Release
Changes fixed in the next release are integrated to the PreRelease branch. This is hardly used by the game, art, sound or design team (apart from a few that desire the latest features). These changes are tested by me before production is updated to the latest version of PreRelease with the newest changes.

Every couple of weeks the team integrates PreRelease into Production. This is called a Release. When they Release they follow the following steps:

1. The latest version of production is integrated into the PreRelease branch.
2. The new features tested in main are integrated into the PreRelease branch.
3. The PreRelease version of the editor toolset is tested by me using a small and large test protocol, in addition it's tested by leads of the games divisions.
4. Once the toolset passes these tests, it's integrated into Production.
5. This new version is then released to the game team.

Reporting bugs through mantis
While emailing bugs to the tech team or walking up to a developer can be useful with such a large team it can hamper workflow. To reduce this bugs are reported in a database called Mantis, accessed in a browser.
Mantis allows a developer to create a new entry for each bug and provide detailed information to reproduce the bug. In addition bugs can be given relations to other bugs and priorities for fixing.
Bugs are either open, resolved (fixed but require validation), fixed in next release or closed. This was soon going to be my second home and the primary reason why I was hired in the first place.

Bugs are reported in different categories, which can be filtered. This way the engine toolset bugs don't show up in the game QA list.

The first thing I did every day was check if there were any new bugs in the following categories:

1. Game Object Editor
2. Level Editor
3. General (bugs not specifically game object / level editor related)
4. Game team (bugs in-game that are editor related)
5. Some more

I would then check the following filters per category

1. Bugs
2. Resolved (bugs supposedly fixed, require validation before closing)
3. Fixed in next release (not fixed in current version but in next version of editor)

Eventually this list grew with some more categories like the sound editor and some more game specific lists. The most important thing though was that I was aware and reproduced the highest priority bugs.

The list spans about 3 pages (about 200/300 entries in total I think, possibly more). When I was asked to reproduce the highest priority bugs and categorize them in:

1. Can reproduce
2. Can't reproduce but with extra info provided
3. Can't reproduce and no extra info

I went through all of the lists that I was asked to check. I somehow missed the word 'highest priority'. Instead I checked and categorized all the bugs (all the way up to bugs reported in 2010). In the end this proved to be quite useful for me since I was now up to date with most of the bugs that they had encountered and where still open.

This also made it easier for me to check the list for new bugs based on the color of their title (since links that are already clicked are colored purple in a browser). This way I could note down the number of a bug each morning, test all of those and be done with the mantis list maintenance in as quick as 15 min or take as long as 2 hours.

Other tools used:

In addition I used the following software and tools.

1. Microsoft Word (for own bugs lists to keep track of new bugs)
2. Photoshop cs2 (screenshots and quick highlighting of visual bugs)
3. Autodesk Maya (importing and exporting models testing, build quick prototype models, shader testing)
4. Notepad++ (to open and examine xml files, quickly note new bugs before reporting etc)
5. Microsoft Outlook (I hardly used this prior to this internship, used extensively during internship to manage emails)
6. A paper notebook (My most trustworthy companion during the internship)

Bug Reporting Procedure
As mentioned earlier, bugs are reported in mantis. This list is quite useful but it's a bit unclear for me to track personally. I quickly discovered that I needed some structure for the way I was going to deal with bugs that I hadn't reported yet or I was quickly going to get confused during testing. This would increase the chance of bugs not even making it to mantis at all or worse, slipping through entirely.
I started with quick notes of bugs in my notepad with checkboxes next to them. I would go over these and check off the bugs that I had reported. The checkboxes worked well but I didn't like backtracking in the notebook and space wasn't flexible enough to increase or decrease when needed.

So I created a table in Word with the following layout:

Date	Type	Extra info
01/01/2013	Crash	Editor crashes when delete is pressed
		Delete crashes editor
	Added to mantis ID 0004567	1. Open editor 2. Press delete 3. Crash?
""	Bug	GO disappears after window is minimized
		When minimizing window GO disappears
		1. Open level 2. Add Game Object 3. Minimize window > maximize 4. Observe: GO disappears
	etc	etc

This proved very useful. For each specific release test I was requested to do, I would create a new version of this document and I would then be able to add new bugs quickly as I encountered them. I could use color-coding to quickly glance over the document and see what bugs weren't reported yet (red), were reported (blue), or were fixed (green) before they were reported.

I could then email this list to the rest of the team so they could pick out the most essential bugs them selves and ask for more reproduction cases. This allowed me to focus on other tasks when testing was done but quickly return to testing when necessary.

Game QA
Obviously I can't disclose much of this but when I didn't have anything to do in terms of bug testing I was asked to do game QA at times. This mainly encompassed bugs discovery and validation for the games. I didn't do too much of this since Vanguard has a dedicated QA employer already taking care of most of this.

My First Day at the Internship:
I made quite detailed notes each day of the tasks I had to do each day throughout the internship, in the first place to stay up to date with my tasks. Since these tasks are mostly project related I can't disclose these though so daily reporting isn't possible. Instead I want to report the biggest QA tasks and hurdles that I encountered during the internship.

While there is no need to go in-depth in each day of the internship, the first day is always a special one so this will be the exception to the rule.

I got in at 9:00 as was instructed to me by mail. I was given a quick tour of the company, shook hands with every employer there, signed a NDA and was then handed to the tech team. There I met my lead 'Jelle'. He and one of the senior engine programmers 'Ritesh' gave me a quick overview of the tools I was going to be working with and got me setup.

After that I was kinda thrown in the deep. I was giving a lot of technical information at a rapid pace, was introduced to the primary tools and database that I was going to use and asked to experiment in the editor for the day and quickly went through a 50 page test protocol as a tutorial (it wasn't as accessible as a tutorial though). I was kinda confused at first but I was able to maneuver through the editor's basic features fairly quickly.

Between 12:00 and 13:00 one is allowed to have lunch in the lunchroom. Lunch is fully provided for. You basically grab a plate, get in line then grab some sandwiches, some sandwich filling and then you have about 30 min to eat your lunch. There is a sandwich toaster that I used daily.

I was given instructions to report anything I found to the tech team. At first I mostly bumped into existing bugs but by the end of the day I found my first real bug. I informed the senior designer and he reported it to the game team. I will never forget it.

At the end of the day both Jelle and the CCO checked up on me. This was not going to be the case often since they are very busy. Then I went home.

First week:
This week was mostly spent getting to grips with the editor. I got more familiar with the bug-reporting database 'Mantis' and was giving some specific test assignments. One of the wishes of the designers was the ability to expose every parameter used in the Game Object Editor to the Level Editor. This posed my first real creative testing challenge. I was asked to try as many complex scenarios as I could where a parameter would be exposed and report any bugs encountered. Eventually I was even required to compare xml files to see if any irregularities occurred. For me this was my first test to see if I could grasp technical instructions quickly.

Daily schedule and obligations
The rest of the months I was going to work according to the following daily schedule.

09:05	Scrum standup (we meet up in a circle and take turns sharing what we did yesterday and what we are going to do next)
9:00 to 12:00	Mantis Bug check, any other tasks to do
12:20 to 13:00	Half hour lunch break (lunch fully provided for)
*** 12:30 to 13:00**	2 games of Quake3 Arena deathmatch with the tech team
13:00 to 17:30	Tasks to do during day, bug testing requests etc
17:30	End of work day > go home

So we start early everyday, have a half hour lunch break and work till about 17:30 then go home. A normal workday.

The tasks I had to do varied per day. One day could be filled with bug testing request, other periods would be dry where I was mostly doing game QA or was free to experiment with my own tests in the editor. The last allowed me to learn new stuff with the editor functionality and learn how to construct game objects for different purposes.

Prior to each Release there was a 'tools taskforce'. This is a roundtable with all the leads of the various departments using the editor. During this roundtable we share the newest implemented features, fixes and go over some of the more important bugs. After the roundtable notes of the meeting are stored in an internal wiki. I was responsible for the meeting notes for two of the meetings during my internship. This was the only meeting I was scheduled to attend.

Release test protocols
The first weeks weren't to heavy on workload. The only essential tests required from me were the release tests using the large test protocol. The only protocol they had was the large 50 page document that I had to work through when I begun. Not that useful when you want to do some quick stability tests. So I was asked to create a simpler version for quick tests. I made a table in word with steps to test the basic functionality (Game Object creation, opening and closing levels etc.)
This would later be done prior to any large test protocol to catch bugs early.

Design Tests
During the first month I had a lot of free time available so I used this to learn to use the Game Object Editor. At first I was experimenting freely but once I had asked for some possible design tests (like building game objects that design didn't have time to do) I was given some basic tasks like building an exploding barrel or health pickups. This allowed me to get familiar with building basic objects to be used in a game and it allowed design to consider possible features for the game that they would lack time to develop otherwise.

Level Design
After about 1,5 month or so I was given my full list of responsibilities for the rest of the intern period. While I was unaware, the period prior to this was used to test if I was the right guy for this intern position.

In addition to al the bug testing I was obviously hoping for a chance to help design with the game. This is after all what I contacted the company for in the first place. All the way below the list of responsibilities I was given that opportunity. One of the levels for one of their games was proving trouble some. They weren't going to pick it up till January the earliest. So it proved useful for them if I designed and placed the gameplay for this level in the months prior to January and worked out some of the kinks that they didn't have time to tackle.

I was given a briefing of the level and then asked to come up with some kind of plan for the level. I made a quick document of about 2/3 pages to explain what I wanted to do in sequence. I was quickly given the remark that I should avoid too many scripted events (A trap you easily fall into when designing on paper) and then asked to start constructing the gameplay using the editor.

I first created test scenes that I could use to test parts of the gameplay. Eventually I added the gameplay to the level itself.

I was asked to make use as much as possible of prebuild game objects. So I didn't build much new functionality. Instead I just placed game objects in the level to construct interesting gameplay scenarios. I was given a really quick steam course of level design (about 1 hour or so of 2 designers sitting next to me showing me stuff in the editor toolset)

During the intern period I had several review moments with the CCO of the company and 1 or 2 of the design team. They helped me decide on a final plan for the level. Eventually I had placed all the gameplay with the required timers and triggers. But these functioned in a more then desired complexity. When the level became part of one of the scrum sprints and it was picked up by Art, one the designers (Nikki) helped me to place the gameplay objects from scratch in a more clean / less complex fashion.

Around the end of the internship Art had finished creation and placement of most of the artwork for the level and I had finished most of the gameplay placement with help of one of the designers.

One really nice experience was when the level was shown during the last sprint review and I was asked to join the review and see Nikki present the level. Reactions

were mostly positive and it was mentioned that the fact that I was able to use my spare time helped the level a great deal. That sort of feedback is always nice to hear. These kinds of meetings are very insightful for me and left a desire for more.

I was only able to work on this in the time not taken up by testing requests. I spent about 2/3 hours a day if possible on this (whenever possible!). While only a small contribution in the grand scope of things I hope the player enjoys the level and it contributes something to the overall game experience.

Sound Features testing
About 1/3rd into my internship I think I was asked to test new sound features. It required a lot of stability testing, creation and overwrites of sound objects, adding wav files to the editor etc.

One of the sound features build was positional sound. This morphed sound from 3D to 2D when in a certain distance. This required a lot of positional testing.

At first this feature only offered the ability for the sound designer to change the parameters. As the feature grew a curve editor was added so the sound designer can create his own DSP curve that would only allow a certain height of frequency's to pass through. This for example allows for a more realistic waterfall loop when you approach it using only one sound file.

The game object system provides the ability to inherit from different game objects. Inconsistency with this system caused a lot of crashes. One of the first things I would test would be overflow errors (values going over the max) and finally edits using inheritance.

I estimate about 2/3rd of my internship was spend testing bugs related to these features.

Engine Test Automation
This was one of the more creative requests I got. My lead had heard of AutoIT from a friend of his. This is a simple coding language that allows one to code UI automation. Simply put, it can automate the opening of notepad, add some text and finally close the file all with the execution of one simple script.

Since the test protocol I used was largely a list of set instruction with little required creative input, using such a language could help us build smoke tests. It could decrease my test workload and it could help the tech team to test stability after implementing new features before asking QA to test more precisely.

Although I learn about programming in my free time I am not a programmer. So I downloaded tutorials for AutoIT and learned how to use variables and functions in it. The syntax is very similar to BASIC.

The tutorial tests were running well but when I tried to automate simple usage of the editor I bumped into a major roadblock. I was not able to access .NET objects properly and this was going to pose a problem.

If this was going to prove any useful I wanted to reduce the time it takes to build test cases to the bare minimum. It wouldn't be very productive if you spent a majority of the time bug testing the bug testing software while bug testing.

AutoIT proved to be a dead end for now. Instead I looked for alternative solutions and found Ranorex studio. Unlike AutoIT this was a commercial package with a lot more bells and whistles. It's a full UI automation framework that allows the user to build recording modules that execute actions on objects in a UI (for example it finds a button in the UI and clicks it with the mouse).

The software license is quite expensive and it's unsure if Vanguard is going to make further use of it. But I had 30 days (the time of the official trial period) to build a test case. Due to the other tasks that I had I wasn't able to spend a lot of time on it but I managed to automate about 1/3rd of the large test protocol. To achieve this I created recording modules for each of the actions that one could perform in the editor.

While the Test Case isn't useful enough yet, Ranorex can be changed using C#. If a more experienced programmer looks at it we might be able to create a more useful Test Case that really helps to spot early bugs in the months to come and perhaps allow for other useful automation.

I only made a small step in that direction but it was a nice challenge and proved an interesting creative challenge alongside the repetitive testing tasks.

5. Personal Evaluation

Since I have already gone in depth into the different major tasks and hurdles I will keep this brief.

Goals
When I started the internship I had given myself 3 major goals.

1. Make sure you keep yourself busy as much as possible.
2. Contribute as much as possible, hinder as little as possible
3. Observe and Learn as much as much as possible while your there

Needless to say I think I met most of them.

1. Make sure you keep yourself busy as much as possible.
When I had off time I tried to keep myself busy as much as possible. At first I was dependent on instructions from the team. I was hesitant to ask for tasks but later on learned to ask for tasks and approached members more directly. I think I have done a lot during the times that I didn't have any clear task.

2. *Contribute as much as possible, hinder as little as possible*
Fulfilling QA duties took priority so I made sure to always do those first. But I streamlined this process well enough that I freed up time to work on left over design work and research possible QA enhancements (like test automation). I think I contributed a fair deal this period. The biggest sign of this has been the increase in editor stability over the past few months.

I tried to interfere as less as possible. Everyone is obviously busy and I'm just a small cog in the machine. So I made sure to only offer my view on possible design questions when asked but never to just jump in with my own opinion myself prior to that. Eventually those questions come your way anyway if you show good behavior.

3. *Observe and learn as much as possible while your there*
At first I was positioned close to a game team. I caught of a lot of design talk with my ears and I was able to observe the game development process a great deal. Halfway through my internship I had to move due to the arrival of a new programmer and spend the rest of the months close to the engine team. Thus the last few months I worked more closely with the engine team, was able to approach them faster and that sped up the bug reporting I think. I also tried to catch glimpses of tech problems they tried to solve but I definitely learned that true game tech code is a great deal more complex than I can now grasp.

Professional attitude
I think I was quite professional. I was on time with most of my tasks, I worked hard during the day and I asked question but avoided to ask too much.
During a workday I didn't encounter many issues.

Travelling to work posed some issues though. I tend to wake up a little too close to the time that I need to leave. I left home at the latest possible time to catch my train to arrive in time. This usually works for me but rush hour traffic in public transportation posed quite a problem. I arrived late due to delayed trains a lot of times and had to call in late more often then I would have liked. While it improved in the last month I think I could have been more punctual and less hindered by traffic.

6. Tutorship during internship

From the school side I hardly knew it was there. My position is obviously different (did not take place during normal internship period in curriculum). But regardless of that I have only met once with my intern tutor and that was during a review with members of the company. Apart from that there was hardly any guidance apart from a few instructional e-mails directed at large groups of students.

From the company's side it was adequate. I had a couple of short reviews on my performance (nothing that weighted to big) and due to the scrum checkups I think most were aware of what I was doing. I was largely on my own during testing though. I got a lot of guidance while building the levels and whenever I encountered issues someone was always quick to help. There wasn't a complete intern guidance program in place but I didn't feel alone, felt part of the group and it wasn't too hard to find advice when needed.
When I encountered early signs of carpal they were quick to provide me with an ergonomically better mouse and mouse pad. All in all it they were very supportive.

7. Intern evaluation for other students

Depending on what your looking for I would recommend this position for the experience it gives you. It's hard to find entrance in this company if your less production focused like I am. If you're an artist for example I would advice applying for the Art Intern positions of course.

QA isn't game design and you shouldn't forget that. It will feel repetitive to a designer and the creative process is often stopped when the bugs occurs.
But in this case it got me acquainted with all the tools designers use. I probably now know a few uses of the toolset that they might not even know since I bug tested it so much. That experience is invaluable and you won't be able to find it hobbying on your own. The benefit of this is that when they are looking for designers I have a step ahead because they don't have to familiarize me with their tools.

Make no mistake, as a designer you will be mostly using their toolset to build the games over there. You will not be getting a position as lead designer that charts the course for the next project. Instead you will at best be asked to design and build systems for the concepts already made in their toolset, best to learn that as soon a possible.

In addition, learning the ropes of a video game developer is a big plus and again something you can't simulate when working on your own.

You need some prior tech knowledge though so make sure to familiarize yourself with different engines workings and usage.

8. Conclusion

I don't have much to add or conclude to be honest. I look back at this as a fun internship, I learned a lot about the process and tools used at a professional game development company and I'm glad I was able to contribute to development in a meaningful way.

While I still don't desire a QA position I now at least know I can fulfill it. Hope remains though that one day I can find a proper design position cause even though I only got a small taste off it at this company it left me wanting for more.

Thanks for reading.

Joeri van Ees

9. With special thanks to:

Utrecht School of the Arts
Emiel Kampen
Wendy Draaijer
Annelie Prins

Vanguard Technologies
Jelle van der Beek
Ritesh Oedayrajsingh Varma
Arjan Jansen
Nicolas Mercier
Johan Donderwinkel

Harm van Dinter
Mike van der Voort
Benjamin Koning

Vanguard Games
Guy Melis
Martin de Ronde
Nikki Kuppens
Kent Kune

And anyone else at Vanguard Entertainment that I forgot to mention.

If you are a student interested in a QA internship at Vanguard Games and would like more information about the internship, feel free to email me your questions at:

lifeisagamenotaproduct@gmail.com