

Laiho Antti

Collaborative game development environment for educational purposes

Helsinki Metropolia University of Applied Sciences

Master of Engineering (M. Eng)

Degree Programme in Information Technology, Media

Tekijä Otsikko Sivumäärä Aika	Laiho Antti Yhteisöllinen pelinkehitysympäristö opetustarkoituksiin 97 sivua 13.9.2013
Tutkinto	Master of Engineering
Koulutusohjelma	Information Technology
Suuntautumisvaihtoehto	Media Engineering
Ohjaajat	Harri Airaksinen, Principal Lecturer Markku Karhu, Principal Lecturer
<p>Tämänhetkinen ongelma valtakunnallisesti on kouluttaa pelialan ammattilaisia teollisuuden tarpeisiin. Korkeakoulujen on pyrittävä täyttämään työtehtävien monimuotoisuus ja alan yritysten tarpeet kouluttamalla pelialan ammattilaisia. Haasteita koulutukseen tuovat korkeakoulujen rakennemuutokset ja rahoituksen leikkaukset.</p> <p>Opinnäytetyönä tehdyssä tutkimuksessa perehdyttiin Suomen nykyistä peliteollisuutta ja pelikoulutuksen tilannetta ja tarkasteltiin yritysten työtehtäviä sekä pyrittiin luomaan kokonaiskuva siitä, minkälaista koulutusta opiskelijoille tulisi tarjota. Opetuksen kehittämistä tutkittiin eri oppimisteorioiden ja -ympäristöjen avulla, ja yksi kokeileva pelikehityskurssi toteutettiin käytännössä.</p> <p>Pelikehityskurssia varten kehitettiin käytännön toteutuksena tehtäväpohjaiseen opiskeluun pohjautuva virtuaalinen oppimisympäristö, joka mahdollisti läpinäkyvän opiskelun seurannan opiskelijoiden välillä. Tavoitteena oli luoda ympäristö, jonka avulla opettaja voi viedä kurssin läpi vaikka opiskelijamäärä kasvaisi tavanomaista suuremmaksi kasvattamatta kuitenkaan opettajan työkuormaa. Ympäristö antaa mahdollisuuden seurata kurssilaisten etenemistä, kommunikoida heidän kanssaan ja ratkoa ongelmia ajasta ja paikasta riippumatta. Oppimisympäristön seurannalla pyrittiin luomaan virtuaalista ryhmäpainetta, jotta opiskelijat aktiivisemmin ryhtyisivät tekemään kurssin tehtäviä. Ryhmäpaine oli yksi kantava voima virtuaaliympäristön käytössä kurssilla ja joitakin opiskelijoita se motivoi tekemään enemmän töitä.</p> <p>Projektissa myös tutkittiin ja toteutettiin massiivinen monen pelaajan verkkopeli. Pelin sisällöstä ja kehityksestä vastasivat kurssin opiskelijat. Opiskelijat loivat peliin uusia pelialueita, hahmoja ja peligenreen kuuluvia ominaisuuksia. Verkkopelin jatkuva olemassaolo mahdollistaa tulevilla kursseilla pelin kehittämisen yhä monipuolisemmaksi.</p> <p>Lopputuloksena saatiin aikaan toimiva verkkopeli ja virtuaalinen oppimisympäristö tehtävien jakamiseen ja opiskelijoiden ohjeistamista ja ryhmäpaineen luontia varten. Pedagogisesti kurssi toimi, joskin tekniset ongelmat olivat välillä hidasteita. Palaute oli kurssilla positiivista ja kannusti kurssin läpivientiin jatkossa samalla tavalla.</p>	
Avainsanat	Game industry, Education, Game development, Massive Multiplayer Online, MMO, Learning Environment

Author Title Number of Pages Date	Laiho Antti Collaborative game development environment for educational purposes 97 sivua 13.9.2013
Degree	Master of Engineering
Degree Programme	Information Technology
Specialisation option	Media Engineering
Instructors	Harri Airaksinen, Principal Lecturer Markku Karhu, Principal Lecturer
<p>Currently nationwide problem is to educate enough professionals to meet the demand of the industry. Diverse work assignments and increased demand of workforce should be fulfilled by educating game specialists with various qualifications. Challenges in education are structural changes at Universities as well changes in funding.</p> <p>This study undergoes the status of Finnish game industry and game industry education. Also, a big picture of different job descriptions is made to examine what kind of education schools should offer in the future. Improvements in education are done by studying different learning theories and one experimental game development course is done in practice.</p> <p>As practical work in addition to the course, a virtual learning environment was created based on task based learning. This environment allows transparent tracking of studies between students. The goal was to create an environment that does not require much more work from the teacher even if the student group gets exceptionally big. Virtual environment offers teacher tools to follow student progression, communicate with them and solve problems regardless of time and space. Strive for virtual group pressure was done using monitoring activity on the learning environment to make students work actively. Group pressure was one key feature of virtual environment and some students were affected more making them to work harder.</p> <p>In the project also Massive Multiplayer Online game was designed and implemented. During the course students were responsible for developing content and features to the game. Students created new areas, characters and features applicable to the game's genre. Ongoing availability of the game makes development possible in future courses making game development more versatile.</p> <p>As a results a working multiplayer online game and virtual learning environment were developed for sharing tasks and guiding students and to create group pressure experience. Pedagogically the course was successful although some technical problems slowed development for a while. Feedback from the course was positive and encouraged to deliver a similar course in the future.</p>	
Keywords	Game industry, education, game development, Massive Multiplayer Online, MMO, learning environment

Sisällys

Lyhenteet

1	Introduction	1
2	Research problem background: Game development education	2
2.1	Game Industry in Finland	4
2.2	Game development education in Finland	10
2.3	Game Cluster Project	13
2.4	Focus on Game Development in Metropolia in the future	16
3	Pedagogical methods for game development education	18
3.1	Board range of skills in Game development	18
3.2	General learning theories	24
3.2.1	Behaviorism	24
3.2.2	Constructivism	26
3.2.3	Humanism	26
3.2.4	Cognitivism	27
3.3	Issues in teaching methods	27
3.4	Degrees to supplement game industry	28
3.5	Learning environments for game development	31
	Moodle	31
	Digital Tutors	32
	Viope	33
	Game Institute	33
4	Virtual Learning Environment and Massive Multiplayer Online Gaming	34
4.1	Definition	34
4.2	Tasks on Demand Online (TODO) environment	37
4.3	Metropolia Online, Massive Multiplayer Online (MMO) Game	40
4.4	Course goals	41
4.5	Course scheduling	41
4.6	Course Challenges	44
5	TODO learning environment	45
5.1	Original design and specification	45
5.2	Development process of TODO	52

5.3	Group pressure	54
5.4	Creating a course using backend	57
5.4.1	Course content	61
5.4.2	Task content	63
5.4.3	Social activity	64
5.5	User experience	64
5.6	Student progression and Teacher's role	67
6	Massive Multiplayer Online (MMO) platform	68
6.1	Metropolia Online Background and Design	69
6.1.1	Server - Client approach	71
6.1.2	Unity	72
6.1.3	Photon	73
6.1.4	uLink, noSQL and Riak	73
6.2	Final Technical structure	75
6.3	Sub-version Control	77
6.4	Main features	78
6.4.1	Game world design	81
6.4.2	Village creation	84
6.4.3	Character Creation	85
7	Projects results and future development	87
7.1	Current state of the Metropolia Online and Future Development	87
7.2	TODO environment improvements	89
7.3	Feedback from the course	90
8	Conclusions	93
	Sources	95

Lyhenteet

MMO	Massive Multiplayer Online
MMORPG	Massive Multiplayer Online Role Playing Game
WOW	World of Warcraft
LOTRO	Lord of the Rings Online
ADHD	Attention Deficit Hyperactivity Disorder
MOOC	Massive Open Online Courses
IGDA	International Game Developers Association
ESF	European Social Fund
R&D	Research and Development
TODO	Tasks on Demand Online
A. I.	Artificial Intelligence
UDK	Unreal Development's Kit
NPC	Non-Player Characters

1 Introduction

The Study problem of this thesis is to search solutions to educate professionals for the game industry. Funding of universities will be cut in the future while demands of companies and students are growing. This puts schools in a situation where same results are required to be made with lesser resources. It is possible that traditional class teaching is going to receive various and hopefully creative methods that are going to help with future challenges.

The Background for this study is the increased need of specialist in the game companies. Many Universities have realized this and are constantly enhancing their game education and Helsinki Metropolia University of Applied Sciences is responsible for educating professionals in southern Finland. This study tries to find solutions how flexible but efficient teaching could be made to meet challenges such as decreased funding and increased demand.

This study focuses on different methods of teaching game development within University environment and analyzes their positive and negative aspects. The Study also covers current situation of the Finnish game industry as well game development education. In the theory part different learning theories and different skills required by game development are introduced. Also some different virtual learning environments are examined to give a scope to current possibilities in learning game development without contact lessons with the teacher.

Second part of this study provides an overview to a virtual course. For the course a learning environment where many different aspects of game development could be taught to a numerous students was developed. The aim was to teach the basics of Unity to students with a method where the amount of students on a course does not generate too much extra work for the teacher. These kinds of massive open online courses (MOOC) has begun to come more and more popular within universities where students from other countries can take part in the education.

The second part explains the technology behind 'Metropolia Online' - Massive Multiplayer Online (MMO) game that can be utilized and enhanced every year by students on Game Development courses. While students do not actually develop game from start to finish, they can take part in one part of the game. With online game environment, students can also prototype their ideas and learn to develop online game development features that are common in modern games. Students who are more artistic oriented can develop models and other art to the game while students more interested in programming can develop new features like Artificial Intelligence and logic of the game. Using a version control system all students can participate in the same project at the same time regardless what others are doing. Updating content on server with incremental version number keeps the game updated for all the students. Second part also sums up the courses results, issues, future development possibilities and feedback from the students.

The Background of game industry in Finland and game education is examined in chapter 2. Various pedagogical methods and required skills in game developed as well as learning theories are studied in chapter 3. Chapter 4 discussed introduction to virtual learning environment and Massive Multiplayer Online game and how the course was arranged. Virtual learning environment with technical description and practical usability is described in chapter 5. Massive Multiplayer game with architecture structure and features are found in chapter 6. Future development ideas of virtual learning environment and massive multiplayer game are written in chapter 7. Also feedback from students is covered in chapter 7.

2 Research problem background: Game development education

Gaming has strong roots in Finnish culture and while Finland is having a great reputation within game industry the education must be enhanced to keep up the hype. That is why a Game cluster project is started and several Universities are working together to improve the education and sharing the results.

To understand the importance of game development, it is required to examine a few important issues about gaming. Worldwide we spend 3 billion hours in online games every week. It is claimed that playing games can treat symptoms of autism, ADHD and even cancer. If this really is true then it seems gaming has a lot more to give than just an entertaining factor. Tests have proven that people who play social games are willing to help other people more easily. Autistic people have come more active in social situations after playing games. [1]

A big question is what makes us to spend so much time in virtual worlds. What happens there that makes us want to escape the real world and to start to play games? Another question is whether we play too much or should we even play more and what the consequences are. Things that are not possible in real world are probably much easier in games. There are tasks in games that are not available in real life and even a fantasy of saving the planet can become true in online gaming. Usually games are designed so that nothing is impossible within a given scope. All tasks given to us will test our skills but we know that the tasks are doable and thus keep trying. Playing games is actually mostly constant problem solving. According to studies 80% of gaming is failing. An Interesting thing is that even the when most of the gaming is failing we still keep doing it. [1]

The emotions we get from playing games are mostly positive. Curiosity, enthusiasm, excitement, intensive concentration and focusing to the relevant things are the major aspects of online gaming. These feelings are what we are looking for and what we aim for is so called "Epic Win". This means we have achieved something we thought would never be possible. This special moment keeps us trying and looking for new solutions. [1,2]

In a study an Electroencephalogram (EEG) test was made to gamers to measure the brain's electrical activity. In some areas of brain there was lots of activity while playing games. The results showed that the most active moment was not when player found a new item, leveled up or completed a task. Instead the most active moment in brain was when there was a difficult mission and the player tried some new way to solve it after failing before. This moment develops our own skills and way of thinking. [1]

An average young person in a country of a strong gaming culture like in Finland has played 10 000 hours by the age of 21. [2] This 10 000 hours is quite a magical number for a couple of reasons. Malcolm Gladwell's idea is that it takes approximately 10 000 hours to master a skill. Let it be true or not but this means that the average gamer has played games so much that he/she is an expert in playing games. Unfortunately this gaming skill does not help much in real world unless it is deployed in to a specific use. The gamers are really good in something but nobody actually knows what that is. [2]

10 000 hours is also an interesting number in school context. With a decent attendance a regular student will spend around 10 000 hours in elementary school between grades 1-9. While students attend school every day for almost a decade they also play games around the same amount. This means we have a potential parallel education system working with the school environment. If we just put that time into right use we can even find creative solutions to the problems of the real world. [3]

Since 2004 the start of World of Warcraft (WOW) gamers have spent 5,93 million years adventuring in online world. Currently there are 500 million online gamers looking for new missions and solving problems in virtual worlds. The intelligence and mental power combined in these gamers is great. It might even be extensive enough to find solutions to real world problems such as poverty, unemployment, homelessness, corruption and other major issues of our society. Gamification is a method of using game-like features to solve problems and reward gamers after accomplishing a certain task. [1,2]

2.1 Game Industry in Finland

Gaming as well game development in Finland started in the beginning of 1980 when the first game systems appeared on market. Systems like Commodore and Amiga started gaming but the actual commercial development can be said to start in mid 1980's when first games Yleisurheilu and Raharuhtinas became available.[4] In 1986 the first Finnish game called Sanxion was released and it received publicity also abroad. The game was made by a Finnish developer Stavros Fasoulas and it was published by an English company called Thalamus. [5]

Remedy is one flagship company in the Finnish game industry. Remedy's first game Death Rally was released in 1996. Death Rally can be considered to be an old game but it is still sold in Steam and actually Remedy re-launched their first game on iOS and Android and Windows phones in 2011 and 2012. Death Rally has been downloaded over 1,8 million times since launch covering development costs in three days. One of the first globally noted game was Remedy's Max Payne released 2001. Max Payne has been famous of 'bullet-time' effect where the player can slow down time to operate more accurately.[16]

Finnish game industry is not only about Remedy. Bugbear has developed many rally games like Rally Trophy and FlatOut games. Frozenbyte developed Shadowgrounds to be released in 2005, Housmarque did Super Stardust HD to PS3 in 2007. These are just a couple of examples until Rovio unleashed their Angry Birds for mobile devices. Rovio, Supercell, Remedy and Fingersoft are great success stories that show how the dedication in games in past has now become an important part of Finnish image around the world.

Finnish game industry attracts international publishers and Finland can be considered an very interesting place within worldwide game scene. Finnish Game Industry has been growing rapidly during the past five years. There has been notable increase in employment of the Gaming industry. In 2004 there were 600 persons working on games while now it has increased up to 1800 as seen in Figure 1.

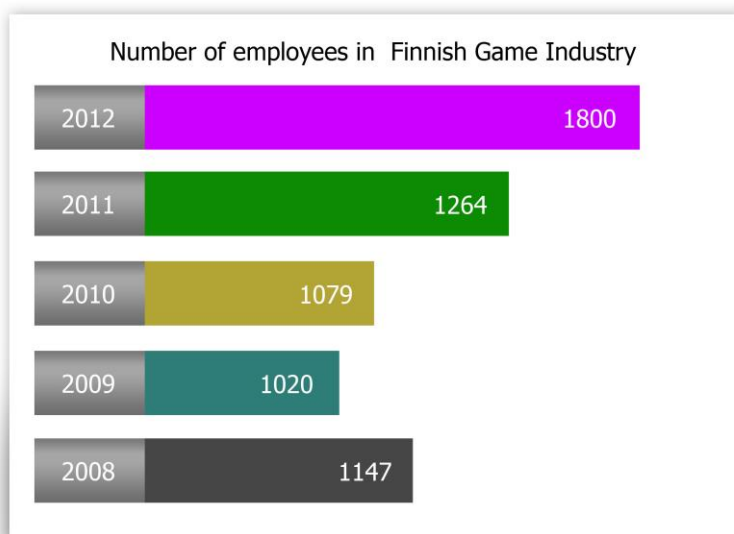


Figure 1: Employment has been growing recent years

Finnish markets are still relatively small, so most of the companies are aiming to foreign markets with their games. Games are also important cultural export for Finland. Currently in Finland there are at least 150 companies in game industry. Almost half of the companies are less than two years old so there are lot of games to be expected in near future. Between 2011 and 2012 there have been over 80 million USD investments on Finnish game scene and industry value has gone up as seen in Figure 2. Finnish start-up companies attract international investors and currently Finnish game industry is expected to grow more than global game market. If growth keeps constant, the Finnish game industry will have a turnover of 1,49 billion by the year 2020. [6]

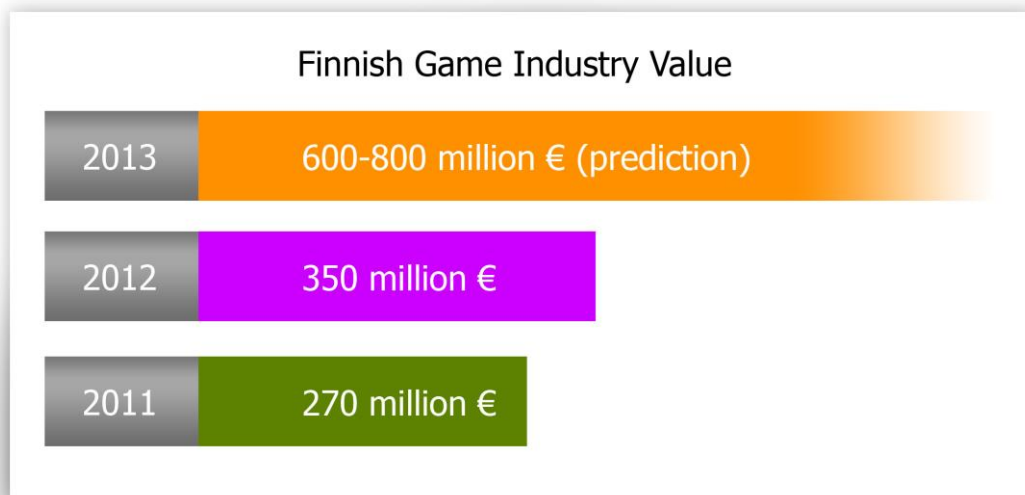


Figure 2: Value of Finnish Game Industry has increased notably

Finland has numerous strengths for being one of the best countries for game industry. Companies and developers have a creative mind-set and are looking for new innovations. When this is combined with technology and excellent know-how, the quality of the industry is high. Game companies do require money for research and development. TEKES is one of the most important funding organization in Finland. The purpose of TEKES is to fund and support companies and organizations that aim for growth and innovations. Companies that are building a product or service can have funding from TEKES to keep their business alive during development time. This kind of funding system is effective and quite many of Finnish IT companies have received TEKES funding. [6, 7, 8, 9]

Finnish gaming industry has world class reliability combined with agility and flexibility. When this combination is added to high technology and good financial support system we have good environment for success. There is also excellent gaming education in Finland that probably takes its popularity from the history where Finland has had a strong gaming culture. Finnish winter tends to be quite long so games can be considered to be a relief for the dark and cold times. [6, 9, 15]

While TEKES is one important factor in Finland, it is not the only organization working for game companies behind the scenes. 'Invest In Finland' (www.investinfinland.fi) is government agency whose purpose is to attract new international investors and give them guidance and help to come to Finnish markets. Invest In Finland is not only for gaming but other industry fields too, but currently Finland being the hotspot of gaming Invest In Finland is a good companion to keep financial flows open. This can be seen in figure 3 as the turnover has been increasing constantly. [10]



Figure 3: Turnover of Finnish Game Industry

Neogames Finland ry is an association that keep track of Finnish game scene. Working as a non-profit organization it support companies, educators and public sector to work

together and find contacts. It is the hub of Finnish game industry that regularly makes studies of the direction where industry is going. [11]

The International Game Developers Association (IGDA) is a society that supports game developers to succeed in their work placements and careers and connecting professionals with each other. IGDA also brings forth issues and matters concerning game developers minds and tries to solve problems. IGDA has several local chapters where IGDA Finland is one of the most active and fastest growing. IGDA Finland has monthly gatherings where professionals and individual developers greet and meet to share best practices and build social networks. In gatherings professionals share their knowledge while at the same time new talents are being attracted to the community. IGDA does not represent Finnish game companies like Suomen Pelinkehittäjät ry does. This way we provide support for individuals and companies from two different sides. [12, 14]

Finland is a small but vibrant country in game development. There are several Regional hot spots that have attracted companies, education and talented individuals to develop games and the industry. The main area currently is Helsinki region that is the home of Rovio, Supercell, Remedy and over 50 other game companies. The area is also noted by educators where Aalto University and Helsinki Metropolia University of Applied Sciences are both educating new professionals for the industry. Distribution of game companies can be seen in Figure 4.

Tampere works as another hub for gaming in Finland. There are around 40 companies working on games employing over 200 persons on local game industry. Tampere game industry has been increased with the help on Manse Project. The outcome of the project has been positive. The industry is growing very fast and turnover has gone up from 1,8 million euros to 6,4 million euros between 2011-2012. Number of employees has gone up from 107 to over 200 within same time scope. [13]

Other regions in Finland are Kotka, where nearly 20 companies with over 100 employees are creating games. This South-East region in Finland is the home of Playa game cluster that includes all different aspects of game development and business from individual developers to various stakeholders. Playa's goal is to grow and expand to be one of the most important cluster in the world in game Industry. [6]

All gaming industry is not located in southern Finland. Oulu and North-Karelia have over 40 companies with two universities and other high quality education and couple of success stories like Fingersoft with Hill Climb Racing and SingOn.fi, an online karaoke singing game. [6]

Kouvola and Turku regions are also lively places on gaming scene. Kouvola region has recently seen rapid growth in game companies and there is concentration to motion capture, post processing and audio work whose are also needed in game development. Turku region had LOAD club to support game development and the area has also been growing rapidly within couple of years now holding at least 15 companies in the industry.

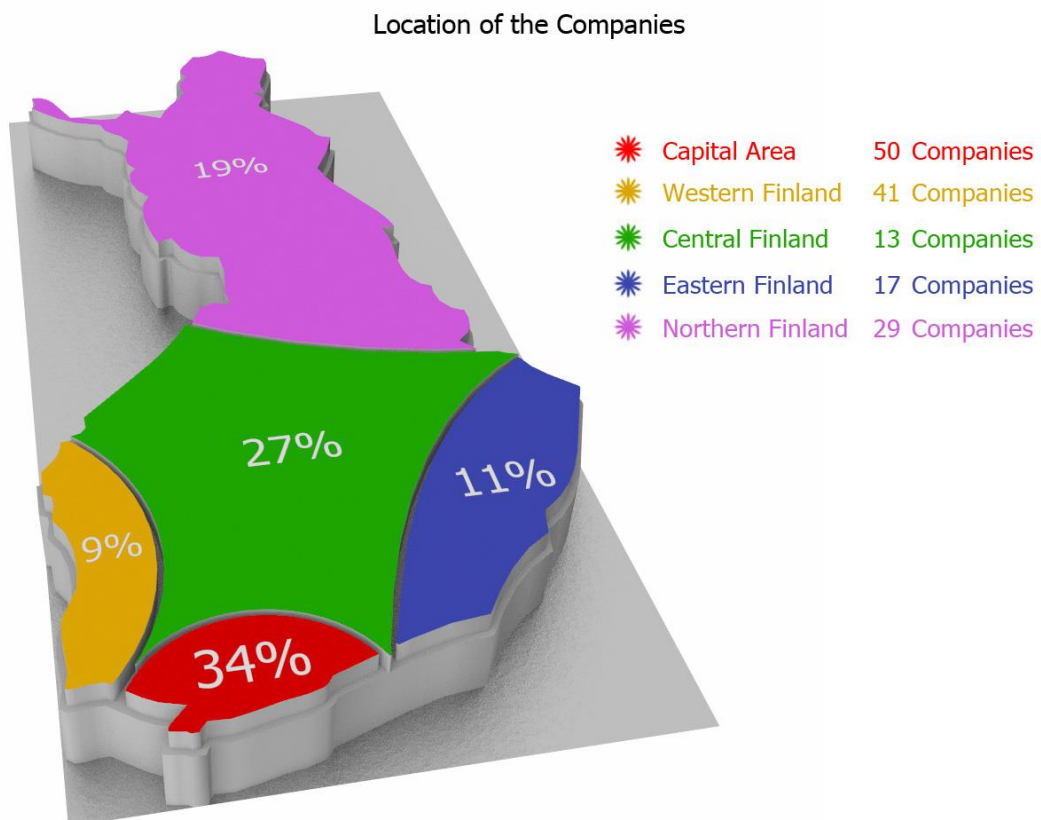


Figure 4: Location of game companies in Finland

Overall Finnish game industry is an active and vibrant community. Finnish game companies are more like one big family than rivals who try to grab the biggest share of the industry. A current problem is the lack of professional programmers and graphic de-

signers that puts companies to support each other in various ways. Growth of the game industry is due the digitalization of the industry. Online market places such as Apple's App Store have given possibility to distribute small games more easily to bigger audience. No more a concrete game box on the shelves of a general store is needed. Instead games are sold in virtual market places. [72]

2.2 Game development education in Finland

Finnish game education is famous for its quality. Currently there are several educational institutes from high school to universities, where game development is one option to study. The education is also spread across the country just like the industry itself. Some schools concentrate more on graphics and art while other are more focused on programming, Artificial Intelligence and designing game usability and mechanics. Some schools have courses on music and sound effects. The amount of game development education is increasing in near future. In many organization studies are focusing heavily on projects and uses 'learning by doing' mentality. At least following organizations give education in game development.

- Aalto University
- Kajaani University of Applied Sciences
- Kouvola Vocational College
- Kymenlaakso University of Applied Sciences
- Helsinki Metropolia University of Applied Sciences
- North Karelia Municipal Education and Training Consortium / Pelitalo
- North Karelia University of Applied Sciences
- Oulu Vocational College
- Sataedu
- Tampere University of Applied Sciences
- Turku University of Applied Sciences

Aalto University offers a specialization in Master's Degree Programme in New Media. Game Design and Production is 120 credits (ECTS), where game design and building is studied in theory and in practice. Education is mostly project based and students

improve their skills in game developing and production. Additional studies can be taken from areas of 3D animation, storytelling and interaction design. Studies take 2 years and are full time. [17]

Kajaani University of Applied Sciences provides information technology training program. After the first year students have chance to choose a specialization. Either game technology or vehicle information systems can be chosen. Final decision of specialization will be done based on first year's success. Game technology studies focus on real-time 3D and programming of game engines and embedded systems. In Kajaani studies are also mostly project based where students are given possibility to produce their own game ideas and designs. Bachelor studies take approximately 4 years and language of instruction is Finnish. [18]

Kouvola Vocation College (KSAO) offers and supplement course of game design. In the course necessary skills and knowledge is provided to work in the industry. Studies are fulfilled by local lectures but also virtual studies and tasks are used on the course. Duration of the course is 8 months where first half will be spent to improve and deepen development skills, while on the latter part the skills will be adapt in practice. 16 students are accepted to the program. Studies include things like concept design, learning games and game platforms. Also teaching of programming, mobile games and 2D and 3D graphics are given along with business and entrepreneurship. [19, 20]

Kyminlaakso University of Applied Sciences also has information technology programme where Game Programming is one of the two specialization options. Studies take 4 years including 240 credits. Game Programming is held within GameLab environment that is a learning environment for game enthusiasts. Focus is on game programming but students acquire skills of general programming also. Students participate on projects with local companies and make their own projects during the studies.[26]

Helsinki Metropolia University of Applied Sciences has started to improve game education. Currently there are various courses of programming, 2D and 3D graphics, design and projects in different study programs. Game education is going to increase dramatically within couple of years offering specialization both in bachelor of engineering and medianom. First courses give basic knowledge in game graphics, programming and designing during the first year of specialization. After receiving initial knowledge in

game development, students participate in innovation projects. In these projects students create games from start to finish. Innovation project also offers customers from the industry to participate in the process. After studying game development courses and completing innovation project students have ability to work in a game company or form a start-up company. This thesis and Game Cluster project are part of the ground work for improving game development education.

North Karelia Municipal Education and Training Consortium offer Media-Assistant studies where game development can be taken as a specialization. Studies are held in **North Karelia College Outokumpu** and the focus is constructing game scenes and worlds using 2D and 3D software and tools that are used in the industry by various companies. Studies include lots of practice and projects that are published. The scope is unique and studies are full-time and take about 3 years with 120 credits. [21]

North Karelia University of Applied Sciences have modules of Game Programming inside Information technology program. These modules use C++ as programming language and studies include topics like object-oriented programming, design patterns and concurrent programming. Bachelor studies take around 4 years.[22]

Sataedu also has a Media-assistant studies where focus is on computer animations and games. Education brings versatile experts to the industry who have problem solving skills with ability to work in groups. Studies consist of 120 credits that include animation, concept art, scriptwriting and content production. After graduating from Sataedu studies can be continued at University or University of Applied Sciences. Adult education is also available. [23]

Tampere University of Applied Sciences have a degree of bachelor of business administration that takes around 3,5 years and includes 210 credits. Program in information technology has various specialization options where game production is one option. Studies give good knowledge in business thus giving ability to work specialist in a game company or an entrepreneur. Studies lean heavily on projects and work placement and final thesis are naturally part of the studies. Specialization in game production includes designing, programming and graphics. Business aspect is not forgotten and cooperation is recommended with local companies and societies when working on projects. [24]

(106)

Turku University of Applied Sciences offers a bachelor's degree in information technology. Four years and 240 credits includes basic engineer studies and after first year one of the specialization options is required to choose. Game technology being one option concentrates on programming and various fields of digital media. Graduates can work in companies as product developers of within digital media. Studies include project work and students can affect their studies according to their interest.[25]

2.3 Game Cluster Project

Game Cluster is a project where several Universities of Applied Sciences with Game Industry companies are working together to improve industry and game education. The aim is to enhance entrepreneurship and guarantee high quality education of Southern Finland. Project is leaded by Cursor, Kotka-Hamina Regional Development Company. Project is funded mostly by European Social Fund (ESF) in Finland accompanied by government and private companies and partners are various education institutes in Finland as well as game industry companies seen in Figure 5. Budget for the whole project is around 2,5 million euros and duration for the project is 2012 - 2014.

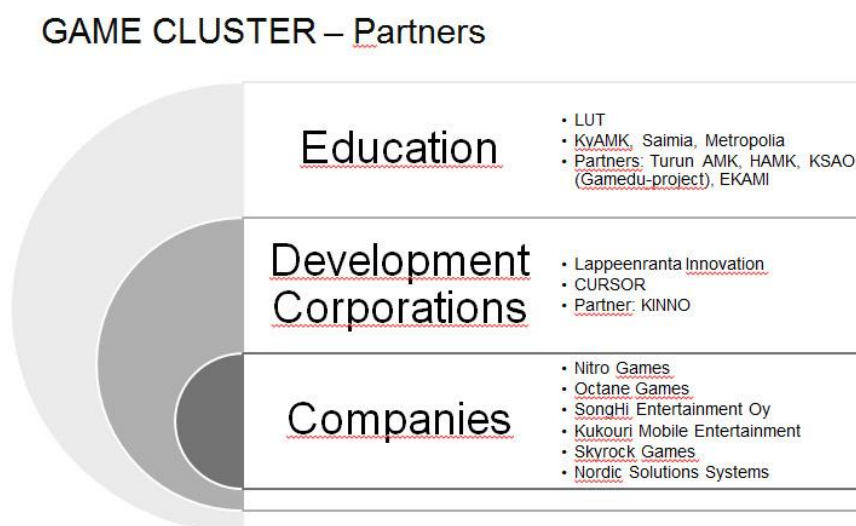


Figure 5: Game Cluster Partners [28]

(106)

Worldwide Game industry is one of the biggest field in the entertainment industry where turnover was around 50 billion euros in 2011 when Game cluster project was about to start. Game industry in Finland is becoming more popular thus timing for the project has been perfect. Working in the game industry is complex, where high technology and diverse art is combined in a working product. Advanced skills and knowledge is required to success in game industry. These skills in turn require proper education and attention to hobbyists of the industry. Game industry also competes with ICT industry thus sharing manpower. This means that education for game industry is supporting other field too and game industry can be expanded. Timing of the Game Cluster project is good but Finland is having challenging financial years ahead. Game Cluster project and Finnish game industry have their strong parts but there are also numerous weaknesses and threats as seen in Table 1 of Game Cluster SWOT.

Currently game industry in Finland needs 200 - 300 new professionals every year. This means education has to be greatly expanded to respond to the demand. Education should be raised to the level where other similar fields are, like for example TV and movie industry. Proper special education programs and courses should be created within Game Cluster Project. [27, 28, 29]

(106)

Game Cluster SWOT

Strengths	Weaknesses
<ul style="list-style-type: none"> • Some Flagship companies nearby • Education • Good regional support system • TEKES and R&D support system • Russian market and subcontracting • Strong gaming culture • Good price vs. quality ratio 	<ul style="list-style-type: none"> • Lack of risk-funding • Business know-how • Marketing know-how • Selling know-how • Too few micro companies
Possibilities	Threats
<ul style="list-style-type: none"> • Game industry keep going despite depression • Finnish good reputation abroad • Education makes groundwork for Game Cluster • Value chain pieces in many companies • Building up funding instruments 	<ul style="list-style-type: none"> • Workforce availability • Possibilities of Game Industry is not understood by decision-makers. • Young people do not have courage for entrepreneurship

Table 1: Game Cluster SWOT [28]

(106)

At the moment there are not enough modern and efficient study programs in Finland where industry demands could be fulfilled. Support is needed for start-up companies to learn best practices of entrepreneurship. Competition on domestic markets is hard and that is why marketing and business education is greatly needed. These are needed and examples what Game Cluster project tries to solve.

The goal of the project is to guarantee education that supports the field as good as possible. Project also creates study and developing environments between companies and universities. Best practices are shared among teachers and common course material is created for sharing. Project also improves business knowledge and education for students and goal is to form international co-operation with companies. One important role is to educate professionals and bring them to the industry to work on games. After all nowadays everything is up to money. That usually is the one that matters. So finally the most important thing is that the revenue from games should be increased and money flows should be targeted to Finland.

Measurable goals for Game Cluster Project are 15 new formed companies and 60 new job positions. Qualitative goals are that co-operation with companies is improved and education is improved to match industry's demand as mentioned before. Business and entrepreneurship is improved and relationship with domestic companies is enhanced by all sides. Overall the goal is to create strong brand to southern Finland that attracts people and companies from all over the world.

This study and all parts of it are included in the Game Cluster Project. TODO system and Massive Multiplayer Online environment were developed under the project. [27, 28, 29]

2.4 Focus on Game Development in Metropolia in the future

Current plans are to create a specialization option towards game development. Specialization would be part of information technology and plans are to start it as soon as possible. Hopefully in the beginning of 2014 there will be the first round of courses starting. This means several game design courses needs to be formed. Detailed plans

(106)

for the specialization are under work but course list is most likely look something like following:

- Game physics programming
- Game A.I. design and programming
- Game graphics
- Algorithms in games
- 3D graphics
- Game designing
- Producing games
- Game development with Unity
- MMO game development

Compared to some other specializations game development is looking intriguing. Game development educations currently has a good ongoing drive so there may rise a problem if there is large crowd of students willing to take game development specialization. There has not been much discussion what will happen if suddenly everybody wants same specialization. How is school going handle a situation where teacher capacity is exceeded because of the popularity? Will there be a selection process done for students and what happens to students who cannot take the specialization they want but are forced to some other studies that are not so interesting for them? This may lead to a loss of motivation and in worst case to quitting the studies. Of course everybody cannot take game development specialization because school has responsibility to educate people to different industries.

But the basic idea is clear. Courses should be designed the way they support a massive amount of students if needed.

3 Pedagogical methods for game development education

3.1 Board range of skills in Game development

From many job titles game developer is the one that can mean many things at the same time. National Careers Service working mainly in USA gives brief description what game developer job is all about. It is mentally heavy to work under a pressure and meet deadlines but still keep up creativity and imagination flying. Working in a group and having good social skills are important qualifications for a game developer. These qualifications are just some prerequisites before programming, art or other skills will be needed. [57, 58, 59]

There are various elements in games that need to be tackled during Game Development phase. This phase in a complex game requires either a lot of people to complete or lot of time from a small team. First of all the first element of a game is a gamer, who is looking for excitement, adventures, logical problems or just pure action. This is a matter of the designer to determine what the game is going to be like.

When the game is developed, there usually is a set of rules that the player needs to obey. These rules need to be clear and well designed to player to enjoy the game. Again, the game designer is in important role but the actual implementation needs code. Programmer or usually programming team creates the features needed for boundaries of the game. This programming task is one of the most time consuming during the whole game development time.

Gamers want to achieve something. Usually they are looking for the 'epic win', the ultimate experience the game has to offer. It can be based on a feeling, successful ability or it can be a visual representation of character to save the world. Overall whatever the goal is, there needs to be a designer, coders and lot of ideas how the 'epic win' is accomplished. Without a goal the game will be meaningless.

To achieve the goals usually player is going to meet some opponents. Creating these opponents is a combination of various professionals. Designer tells the team what the opponents are. Graphic designer will sketch and plan the visual look of the opponents.

(106)

After that, the 3D modeler creates the actual game object. It is possible that in high technology games there is 3D sculpting and various other steps needed until the game object is usable in the game engine. Rigger and animator are responsible of the movement of the opponent and after the opponent is UV-mapped a graphic designer needs to create different kinds of textures game object to look believable. When the opponent is completely done, it still needs to have logic and artificial intelligence (A.I.) to give challenge to the player. This A.I. needs to be coded by programmers and everything needs to be given to testers to make sure everything works as expected. [30]

After different elements are shaping up there is still a need for a concrete environment for the player to play the game. This representation of the game is up to level designer and 2D and 3D artists to complete. Graphic designer is also needed for sketching and concept art for pipeline to work. When the game levels are done with the player and opponents, there is usually a need for some kind of story to keep up excitement. A good tale keeps the gamer come back for the game.

After the game is done, it needs a marketing campaign or otherwise nobody knows there has been invested lot of time, money and effort to complete a game. This marketing campaign and other business aspects are important because after all they bring the money from the sales. The style of the campaign can be various. It can be anything from a small viral campaign to huge commercials in TV. Again, an individual or a team of professionals are needed to make the game look appealing.

Game development needs lot of professionals from various fields. From university's point of view this is difficult, because educating students to be professionals in everything is almost impossible. This leads to that some specialization is needed because everyone cannot be good at programming, 3D modeling, level designing, testing and marketing, not to mention all other tasks needed game to be good. If a school is going to teach game development, there is a need for many teachers who are professionals in their own field. This way school has resources to educate students for the different parts of the game development pipeline. This also means there is a possibility to create great games in school environment, because all support is available during study time.

(106)

Students usually don't know a thing about game development when they start studying. This means that they cannot know what they want to specialize. Do they want to be programmers, artists, designers or do they want to concentrate to the marketing and business. Within this study there was an idea to develop a system where most of these skills can be explored with friendly environment thus gaining experience what could be student's favorite field in game development. This way students know what they are interested in and they can continue their studies to that direction. Later in study program when there are project works in groups where game development is done they can take the role they are most confident with. [31]

Usually employees in a game company can be divided roughly into three different categories; Business, Programmers and Artist. But in one category every employee does not do the same job. There are programmers who have their own speciality and there are also different kinds of artists. The problem from the schools point of view is that there are no resources to educate programmers to specialize to a very narrow scope of a game development. It could be said that school can provide general skills for programming or creating art and graphics but after graduation the person itself need to specialize to the area it is needed. Of course working full days with a special task makes one to learn really fast.

Below is a list of job titles and short descriptions of Remedy's employees [61]. From the list we can see that almost every single person has his/her own special area in the pipeline although it is probable that many job overlap another a bit.

Business

- CTO: The Boss
- CFO: "The only dude" who understands the current financial statement.
- Executive Producer: Leader of the whole project. Takes care of the project on very high level. Keeps in touch with the publisher. .
- Producer: Takes care of timetables
- Online/Marketing manager: Takes care of Facebook, Twitter, Youtube, Forums etc. that only positive is spoken about Remedy

(106)

- Outsource coordinator: Looks for partners and keeps touch with them.
- Head of franchise development
- Executive Vice President
- iOS platform director
- Business development manager

Programmers

- Graphics programmer: Programs graphic engine and Shaders
- Gameplay programmer: Programs game features
- Backend programmer: Programs server side features, if needed
- Generalist programmer: Programs whatever is needed
- Audio programmer: Does everything where audio is involved
- Animation programmer: Programs the animation system to the game
- Pipeline programmer: Programs stuff to the pipeline, so data can be easily put to the game.
- AI programmer: Takes care of the Artificial Intelligence
- Tools programmer: Programs various tools
- Engine programmer: General programmer for game engine

Artists

- Technical Artist: Works between Programmers and Artista. Writes tools and plugins to art programs (2D and 3D programs etc.) Takes care the content is in right format for the game.
- Environment Artist: Models the game environment
- Concept Artist: Sketches and models concept pictures
- Animator: Takes care of animations.
- Artist: Draws and models
- Game Designer: Game designer. Design gameplay and scripts functionality
- Scripter: Creates scripts for game encounters.

(106)

- Outsourcing Artist: Takes care of assets that are outsourced. Fixes the assets and put them to right format for the game.
- Drama Director: More or less same as Director in movies, but a smaller role in game developing. Takes care that camera and events work together and the story is appealing.
- Character Artist: Character modeler
- Creative Director: Dude with the Vision
- Level Designer: Designs game levels and environment
- Video Artist: Creates videos
- User Experience Designer: Designs and creates Users Interface for the game. Responsible of the usability of the game.
- Audio Designer: Creates audio effects and put the to the game
- Art Director: Directs the style of the graphic look. There can be many, like Art Director and Franchise Art Director
- Writer: Writes the story of the game. Creates dialog and everything that involves writing.
- Visual Effect Artist: Particles, effects, color correction

Support

- IT: Helps when there is issues with computer
- System Engineer: Builds office's IT system
- HR: Reply to job enquiries, takes care of job interviews, take care of problems of people coming from foreign countries.

There are people who purely concentrate on programming and some are only doing graphic and art. Between these is a person called technical artist who more or less blurs the gap between technology and art people. These all around specialists work with art programs but are able to program new tools and features as well. In a case of a small company one person needs to cover wider area of skills to get everything done. This lack of manpower and resources means that smaller teams usually make smaller games. Individual developers or small teams without the funding of game publisher are

(106)

called indie developers. Usually these indie games are little bit shorter and not graphically so intensive but are more creative and bring new ideas to the industry.

Every game development is a unique process with its own features, challenges and timetables. But generally game development can be described with graph seen in Figure 6. This is a simple graph that shows how game designers, programmers and artist work for the game features and outlook while other experts like Technical directors and Subject Matter Experts tell how the content of the game should be done to make educational game work well. These specialists are of course important part of designing and testing during the game development process. [60]

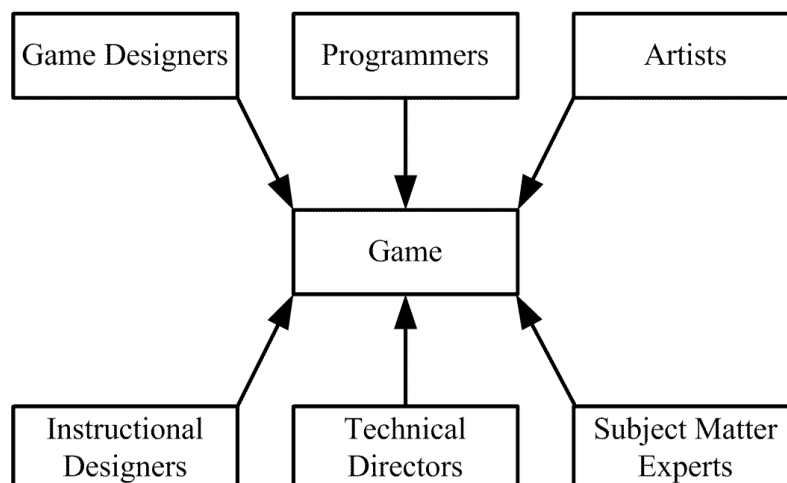


Figure 6: Game development team [60]

Many audio visual projects as well as games can be cut into three parts: Pre-production, production and post-production seen in Figure 7. This is just a rough representation of the project and it is probable that these three phases will overlap a lot. The design may need to be changed afterwards if implementation phase has a problem with it. And testing may reveal problems in implementation and even design that is going to need adjustment during the whole process. This blurs the borders between different productions phases. Eventually everyone needs to work for the common goal and communication is a key to success. [60]

(106)

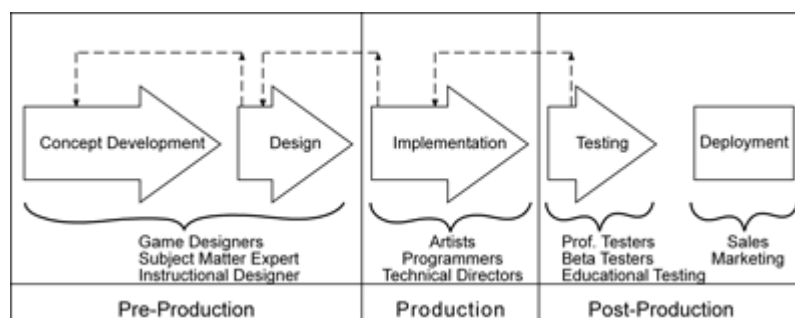


Figure 7: Game Development Process [60]

Teaching game development in school environment is not only covering specific skills but also teaching overall project management. At Remedy different teams work together for common goal. These goals are called milestones and are mostly fundamental features of a game, like “Player needs to be able to protect itself”. These milestones then are completed by using agile development process. This means that two week sprints are made to complete a desired feature. Every day there is a recap of what was done yesterday and what everybody will concentrate on today. Most of the milestones are planned during the design phase and teams may not have heard of those until the production phase so they may come out of nowhere to develop. Milestones are important, because the publisher’s payment comes for completing them. [61]

With smaller companies things are of course different because the whole company can have a common meeting where all aspects of a game are examined and everybody knows what others are doing.

3.2 General learning theories

One purpose of this study is to find out ways to teach game development to masses where teachers workload is not increasing according to amount of students. It is important to examine a couple of most common learning theories to find different possibilities and tools how students could be taught and how they could learn game development. Many of the learning theories have been available for a long time and all have their advantages and disadvantages. While examining different learning theories it is possible to find the best and usable features in all of them and put those features to practice when teaching different things in game development. [32]

3.2.1 Behaviorism

Behaviorism as a concept was introduced in the early 20th century. Learning process is seen as a stimulus based events where some kind of stimulus is given to student and as a result something happens as a physical action or in students mind. This reaction to the stimulus is improving the student thus causing learning. Successful work is generally awarded while penalty is given for unwanted action. In this theory the student is seen as a passive person whereas teacher has done lot of background work. This means that student does not have to be self-imposed or an active learner but rather waits for the stimulus and then makes the correct response. [33]

This kind of learning theory works for simple repeatable tasks where the environment is actively giving a stimulus. The behavior gets reinforcement by giving positive and negative feedback. This way the learner starts to behave in the desired way. An important thing in behaviorism is to split the learning material to small pieces. This way learning is done in small steps where response to more difficult tasks is wanted later. One good example of conditioning in elementary school is when the teacher comes to class room all and students stand up and say "Good morning". It does not matter if the teacher is their own or a substitute, students will stand up anyway. The stimulus there is just that an adult steps inside the room while the response is to stand up and say simultaneously "Good morning". This kind of response can be split into smaller steps and eventually after several repetitions it works smoothly. The same works for people who have done military service. When they hear a whistling voice in the air, they immediately start looking for a pit to dive in even when the audio source is not dangerous. [33]

Ivan Pavlov made a famous experiment with classical conditioning where a dog was the learner. When ringing a bell and giving food to the dog at same time the food makes dog to salivate. After repetition when the dog understands that ringing bell means that food is coming, just ringing the bell without food made the dog salivate too. This classical conditioning is an example where dog's behavior is dramatically changed. [33]

For Teacher Behaviorism means that teaching has to be done with certain way. There is a need for clear goals for responses that can be measured. The learning material needs to be in small chunks and correct stimuli needs to be created to award or punish

student after a response. Until the student has not demonstrate the correct response, the current task is repeated and when correct reaction is done, then teacher moves to the next phase. In this kind of learning the teacher is sharing information and the student is acting as a passive receiver. Feedback is really important and usually learning results should be measured. [33]

3.2.2 Constructivism

In constructivism the learner is seen as an active part of the learning process. The learner is forming new information based on prior knowledge and acquired information. This way learning is in many ways different as in behaviorism where the student had a passive role. It is up to student's prior skills, knowledge and experience how the student interprets new information. Important is that the student actively analyzes information and makes own experiments and questions. This is called problem based learning where the learning is a result of student's own actions. Of course it is necessary that the student understands the problem and the required actions to make learning possible. Social collaboration is also seen as an important role in this learning method. [33]

The teacher's role is not that much in sharing knowledge but creating an environment that supports student's learning as much as possible. This means that the student starts to ask questions and solve those problems that are felt most important. Teacher works as a professional and mentor for students to create best environment for understanding new information. [33]

3.2.3 Humanism

Where Behaviorism was purely sharing knowledge and Constructivism was about creating a suitable environment for the student to educate himself/herself, Humanism is more like giving student the freedom to do whatever comes to mind. Humanism is based on learners experience and the ability to examine prior skills and knowledge which act as the base of learning new things. The learner is seen as a self-imposed person who thinks and understands for what is required to learn. The learner has a goal and free will and motivation to accomplish it. It can be said that a learning process

can be an experience where different kind of means have been used. Feelings, imagination, sounds and other experiences strengthen the learning process. The student is self aware of learning and knows when learning is happening and what would be the best methods for it. [33]

This all requires that student has the motivation and knows how to take responsibility for learning. It also means that the student has to know where to start and how to advance when teacher act more like an observer and supporter. Teacher's role is to create an environment that is pleasant for learning just like in Constructivism. [33]

3.2.4 Cognitivism

Cognitivism was introduced 50 years ago when focus in learning changed to how our mind works and what the processes inside our head are. It more or less rivaled Behaviorism. People can no more considered a passive, programmable animals that react only on stimuli. How the information is processed by different people was the point for study. How the information is perceived, selected, analyzed and stored by learner to acquire new knowledge and skills were also explored. Problems and contradictions in our life gives a reason to start learning. When current skills and knowledge is not enough to solve the problem by nature, we start to find a solution. The learner either tries to search new information or modifies current knowledge to find a solution. New information is seen to be dependent on prior knowledge. [33]

The learner needs to be aware of his/her own skills in learning and what methods are good for learning and how the data is best to analyze. The learner is constantly developing learning skills. It can be said that learner "understands what can be understood and understands what cannot be currently understood. [33]

The teacher should be more like a guide in the learning process and not just deliver information like for example in Behaviorism. The teacher tries to make the student to think and analyze information thus causing learning. There is no need for a detailed planning and goals. Instead seeing the big picture is important. Various learning methods can be used, like pair work, project work and brainstorming. All the time analyzing the information is seen the most important thing. It seems that the goal for learning

process is really to master something but when done right, students could learn faster and better than expected. [33]

3.3 Issues in teaching methods

The traditional way of teaching in a school has been the same for ages. Class teaching is still most used way to teach students. The teacher stands in front of a classroom and students try to understand teacher's message of the lecture. Results vary and for some subjects lecture is very good while in hard subjects there will be students falling behind. This also gives the audience a passive role and the actual learning is hard to measure. [62]

In game development there will always be different kind of students. Some will be more oriented in programming and some are more artists. Then there are always students who know a little bit both and people who have no idea about programming or graphics. This leads to a situation where the teacher just cannot pour the information to into the student's head. There would be people who do not understand a thing while some will get bored because they already know the basics.

That is why in many schools where game development has been taught, project base learning is used. This mimics quite well real time situation in a game company where different teams with different roles develop games. For example one group of students would include a designer, programmer, UX designer, 3D modeler and graphic designer and a business person. In this case everybody needs to know at least something about their subject. So working on a game project should not be the first course on a game development education cycle. Another problem in a project based teaching is the size of the group. If there is a group of 80 students on a course, that would require 10-20 groups to be formed. If each group is creating their own game this will lead to the point where the teacher does not have control of the course anymore. At least it is going to be really difficult to monitor all groups.

This means that the number of groups needs to be cut down which in turn means that all students cannot joint to the course. Budget wise this is not ideal, because the fewer

(106)

students on a course, the less credits there will be given, meaning the school will get less money.

3.4 Degrees to supplement game industry

When a graduated person looks for a workplace in game industry, there is usually some experience required. This so called experience may be more or less an illusion. If around 10 000 hours of game development would make one a professional, then for example 15 years of experience make one a Tiger Woods of the game development. It is not the amount of hours but the quality of the hours spent in game development. In some cases there is not much difference if one has 3 years or 10 years of experience. Working on a domain for a long period a person will eventually get a little stagnant. This means the hours spent on the work will have less quality than a fresh person who may be more motivated but has less experience measured in years. [63]

The problem is to find an employer who is willing to accept a recently graduated person. These newcomers have fresh ideas and they are not got caught by the past. One way to get experience is to start programming and designing on free time. Completing even one project will give experience and a sign that the person has enough skills and patience to complete a project from start to finish. Also a new worker needs to learn the pipeline of the company to be able to work on the game. These pipelines differ between companies and are learned at work place. [63]

But game development is not only programming or designing. Also artists and business people are required and that is why students who are studying information technology are not the only ones who could be part of a game development company as seen in Remedy's list of employees in chapter 3.1.

There are various degree programs that give knowledge and skills to work in a game company. Metropolia offers a variety of degree programs that give graduates a possibility to look for work in game industry. Computer and console hardware has become more powerful and games are getting visually better and bigger. Some games look and feel more like an interactive film where the player can interact with surroundings.

(106)

Metropolia has a degree program of Film and Television where television production is taught including shooting, cutting, scripting along with the production and audio management. This means companies in a need of film cinematics could rely on graduates from Bachelor of Culture and Arts. The same applies to Aalto University's Media Center Lume where various types of education is given on audiovisual media. [73, 74]

Business has an important role in game companies. New games are appearing all the time and only the most popular ones prevail. This does not mean the most sold games are the best ones. These most selling games are maybe marketed properly so that the customers can find them. This all should be written in a business plan and this is where Business School can offer expertise. There are various schools from second level to Universities that offer Business degrees. A game that has a convincing business plan behind does has a chance to success and its operation looks more professional. This also helps when looking for funding and investors. Companies make games to sell them and make money so it is reasonable that there should be a business professional in the team for guidance. [76]

3D Animation and Visualization, Digital Media and Graphic Design are important studies for game industry. Games are often graphically intensive and there is a massive need for content during the development. 3D modelers and animators as well as graphic designers provide this content. Metropolia is one of the few educational organizations which offer full 3D graphic degree program in southern Finland.

Media, Media Engineering and Information Technology are degree programs that create professionals for the game industry. Different kind of programming languages, knowledge in mobile programming and graphic design education are much needed in Game Industry.

Gaming is an experience that puts all our senses into use. Hearing is one of the senses we need in gaming. Sound effects, music and other audio content have an important role too. Sibelius Academy in Helsinki is responsible for higher level music education in Finland. The Academy educates professionals for different fields in Finnish music culture. Gaming is a one exported products of Finnish culture so graduating from Sibelius

(106)

Academy is a way to get involved with game development and production. It might also be a way to get one's own music heard worldwide.[34]

Angry Birds theme music is composed by Ari Pulkkinen who is also responsible for the theme of Trine 2. Angry Birds has been downloaded over 1 billion times. Mr. Pulkkinen's music is played all over the world numerous times in games and Youtube videos. Ari Pulkkinen has commented that by composing game music a composer has a chance to become known globally. Even London Philharmonic Orchestra has recorded their version of Angry Birds Theme. Sibelius Academy and other organizations offering music education degree programs, therefore have an impact in game development. [34,35,36,37,38]

3.5 Learning environments for game development

Digital media, graphic design and similar areas show what game development is all about. But still a degree is not necessary required. Some companies may hire exceptionally talented people if a person suits in needs. Having a degree of course gives an employer a sign that a person has some kind of skills that are required and also other meta skills like ability to work in groups or understanding theory behind practice. Gaming requires a hobbyist mentality and spending free time with game development to become professional. Playing games helps too.

There are many different learning sites, learning environments and tools that help with the learning process. Some of the tools are used by universities and some are independent companies who charge money for example monthly based on the use of their learning material. Also Youtube and Vimeo are full of art and coding tutorials where one can learn essentials but eventually it takes far less time to learn a new subject when there is someone teaching instead of self study. Below a few examples of different learning environments by individual companies and some tools used by Universities are introduced. This is a short glimpse to show how scattered game education can be.

Moodle

Moodle (<https://moodle.org/>) is a well known database based on open source and it is widely used at universities as a course platform. It supports fully online courses but also works along with face-to-face learning. It relies strongly on collaboration and activities and can be used numerous ways as a course platform. It supports different kind of authentication and user roles for course management. It can be said that Moodle is kind of full feature content management system for a course with evaluation and feedback features. It is used on a game development courses along with other courses in different schools. [39]

My own personal feeling about Moodle is that it has too much of everything. The tool is so general that it is made to work in almost every situation, so learning game development with it is not anything special compared to learning for example Biology unless the teacher makes fundamental changes to how Moodle is usually used. Moodle is just a tool and it is totally up to teacher how good the course will be. In Moodle's case it is not enough that the teacher knows game development well. The teacher also needs to be professional in using Moodle to make the course a pleasant experience.

Web based brainstorming and voting tool called Tricider has a thread about Moodle where users can have posted comments about Moodle. Common impression is that Moodle is a very useful platform but navigation may be a little tricky. It is said that Moodle does not encourage students to autonomy any better than other virtual learning environments. Some comments say that Moodle is quite teacher centered and the quality of the course is as good as the coordinator. But Moodle is free and has many user levels and features that support high customization. [64]

Digital Tutors

Digital Tutors (<http://www.digitaltutors.com>) is one of the most known learning websites that offers video-based education for Computer Graphics, Visual effects and Digital Art. Those wanting to learn to make films, games and other digital art are given a massive amount of training within this site. Digital Tutors teaches techniques in 2D, 3D and

(106)

Game development so there is a mix of art and programming with video editing and visual effects.

Digital Tutors videos teach to use most of the main programs that are used in Film and Game Industry. One quite good feature is the Learning Paths, where learner is given a path of tutorial videos to follow in order to get results faster and with better comprehension. Learning Paths is a neat way but it could be enhanced with more game like features to make learning more rewarding. Pricing policy of Digital Tutors is clear. Costs are 45 (USD \$) per every moth. This includes unlimited access to the training videos and projects files. For companies, schools and other organizations there is customized pricing.

Now days many schools have has their funding cut substantially in Finland. This raises a need to plan courses more creatively and in an easier way to compensate for the lack in funding. Every credit given to a student has some financial value to the school. One option for a school to arrange a course might be through Digital Tutors or a similar site. The school could buy 2-3 months account for the site where students start to study independently. The teacher is not much needed until the course is done and it is time for evaluation. Students who use the site material to learn proper tools and skills will pass the course thus reducing the teacher's time and costs for the school. [40]

Viope

Viope (<http://viope.com/>) is a small Finnish company that provides programming learning solutions for educational organizations and as well individuals. Viope's main product is an embedded compiler that makes learning easier to start. It also gives feedback and better support when student instantly sees if the code will compile and what might be the error if it does not. With Viope's system one can learn different programming languages like C, C++, Phyton, Java and PHP. From the point of game development this kind of system is a good tool for learning the basics of a programming language. But quite soon after one has started creating games code is not enough.

It seems Viope does not support any other content except code. This means the system cannot offer support for the full game development pipeline. It is meant to be just

(106)

for the raw, basic programming tasks. When a person wants to start creating content, other tools need to be used and Unity, UDK or similar engines may need to be installed with right programming software. [41]

Game Institute

Game institute (<http://www.gameinstitute.com/>) offers a large learning package of C++ coding and 3D graphic and engine programming with mathematics and A.I. programming. Where Digital Tutors are video based environment for more visual stuff, Game Institute leans to the programming and logic. The Game Developer Package has thousands of pages of information, tens of different presentations and many projects to study. A large amount of learning material is good and again everything is well planned and it looks appealing. Price for the package is 99 (USD \$). Game Institute provides also Unity development with a couple of lessons. [42]

Summary of Environments

Different learning environments have their own scope and approach for game development. Some are more programming oriented while some focus more 3D graphics and art. Usually sites are charging for their service for example asking a monthly fee and then giving unlimited access to the material for a given period. Training material consists mostly of video based tutorials, written instructions, presentations and projects that give user possibility to study the subject. Benefits are that the user can study for example game programming whenever it is needed and can use materials given by the service provider. Lack of support and problem solving alone may be a little cumbersome because the actual teacher may not be available. Usually there is a free trial period available that tries to hook the user to the learning environment thus encouraging the user to stay in the environment.

4 Virtual Learning Environment and Massive Multiplayer Online Gaming

Projects such as Game Cluster get outside funding, which makes it possible to research and find new concepts for courses and learning. Metropolia participates in

Game Cluster project with other school organizations where various tasks are distributed to each participant to help for project succeed. Metropolia's mission is to find and create new learning environments, let them be virtual or physical.

4.1 Definition

Project described in this study had resources to start designing virtual course system that could fit for game development and maybe other courses as well. Learning environments vary a lot, where others are more general and some have specific features for education. There are some common elements in most of learning environments. First of all they have some sort of space that contains information. In a worst case the environment is a pile of html files that without any consistence. Creating a proper architecture for learning environment makes working with it much easier. There are couple of requirements for a learning environment. [43]

Cognitvie learning is partly about solving contradictions. This means the learning environment should have a possibility to answer questions like "Where is an example for this..." or "Provide me an argument opposing this claim..". This kind of functionality would require some kind of search feature and understanding the user's behavior. Guiding the user requires the environment to be a little more dynamic. Storing all information to database makes some features more convenient to use.

Learning environments usually require different roles. Most of the users are students because the learning environment is made for them. But there are teachers, administrators and the author who provide the content and keep it up to date. In some cases if the content is edited, it must be locked so it is not edited by several users at the same time.

Information sources must be visible. Otherwise the base of the information is lacking of credibility. This means that there should be a link of reference to more information when something is put in the learning environment. [43]

(106)

In the future the learning environment will grow and the amount of content will increase all the time. Maintaining information is important. Updating and removing obsolete links and texts should be made as easy as possible to keep content and site alive. In some cases updating a website becomes more expensive than creating it. Update functions should be thought during the development phase and the architecture should support constant rapid updating.

The environment should be actively improved with new features when technology makes it possible. Possibility to social media sharing and audio and video display have become popular during past couple of years. Virtual Learning environment should be a social place just like classroom. Features should be made to support this feeling and usability should be as simple as possible so students do not need to think about it.

An important question is: "Will Virtual Learning Environment improve the quality of teaching?" This is a complex question. Probably not, but it will definitely reduce the costs for the organization caused by teaching. Of course everything is depended on what is being taught and what Virtual Environment is used. Some things work better than others. Some people feel that credits virtual courses them are earned quite easily because nobody is actually watching what students do. Also sometimes students may be little bit lost because they do not know what they are supposed to do next. This leads to a situation where experience of a course is a negative one. If the student needs actively to ask the teacher for guidance the purpose of a virtual self study course is getting a bit obscure. Teacher's presence is sometimes of course needed. Getting face-to-face guidance can solve small problems really fast but this should not be needed all the time.

During Game Cluster project work has been done to create simple, easy to use Learning Environment with easy and slick usability interface. This Environment is being used on a Unity game development course during Spring 2013 and Autumn 2013.

Massive Multiplayer Online (MMO) gaming has become very popular within past decade. History of MMOs go some about 30 years when the first multiplayer games were mostly text based and players interacted with the environment by typing commands.

(106)

Ultima Online opened the more modern representation to MMOs in 1997 and 2004 Blizzard published World of Warcraft. [44,45]

There is couple of variations of MMOs. In some games there are plenty of players online but they are met in a virtual lobby or chat room where game players are gathered and teams or game rules are decided. One example of this kind of game could be Halo or Team Fortress 2. When the game actually starts there are 8-32 players playing the game while others stay in the virtual lobby to seek a game. [75]

Another variation of Massive online gaming is where all players in the same massive world can explore it and compete with each other in different ways. Players in these kinds of games usually accomplish tasks together and explore the world by gaining experience, better items and reputation among other players. Gamers can play against each other or against Non-Player Characters (NPC). Most MMO games have some kind of currency that is used to buy more items or enhance character abilities. In some games real money is needed either to buy a subscription (WOW) or used inside the game to buy new content and items (LOTRO)

Most of the MMO games have a client-server approach. The game's database and functionality mostly works on the server side. The client program is downloaded from a website that requires a registration. Client software acts as a game that handles visual representation of the game. All different players connect their client program to the server thus being able to play with each other and communicate with chat or audio devices.

Tasks On Demand Online (TODO) Virtual Learning Environment and Metropolia Online, Massive Multiplayer Online game are discussed in following sections to explain how the combination of Learning environment and MMO game are working together to create a playable game in a school environment by students.

There were two persons working on the project. I started by myself and designed the concept and programmed the TODO system from ground up. Michael Sundqvist worked as a trainee during fall 2012 when most of the developing was done. He was responsible to program the architecture and the basic functionalities of MMO game.

(106)

Also design and graphic design was involved in the process. Mr. Sundqvist did the code for the server side of MMO game and also functionality to the client software to make it playable. The total development of the TODO system and MMO game took around five months from beginning to the start of the first course. Mr. Sundqvist also assisted as a teacher during the course with some technical issues.

4.2 Tasks on Demand Online (TODO) environment

TODO environment can be found in the address <http://todo.metropolia.fi>

Designing TODO system bases on one's own personal life management. One way to get things done is to write a simple list of things and when the list is visible everything is much easier to fulfill and mark as done when the task is completed. This list could be like following:

- Do Dishes
- Call to do dentist
- Pay the bills
- Water the plants
- Change car headlights

When the list is all the time visible and when there is a moment of free time one can always check from the list for the next task to be done. This method reliefs stress because it is easy to see all the things that must be done. When completing something it gives a good feeling, even if the thing in the list is really minor one like "Take out Garbage". This same "task completing feeling" is put to the learning process in TODO system. This means that a bigger task needs to be cut into smaller chunks for completion. This would lead into a long list of small tasks that students complete in a given order.

Every task is designed to support the next one and thus completing gets easier some small tasks are more theory based and others are practical work of game development.

(106)

TODO system is actually getting some good points from all learning theories. According to behaviorism the learner is seen as a passive person who waits for the signal from the environment. In TODO system every task is a signal. The student should never have a feeling that he/she does not know what to do next during a virtual course. Looking at the list of tasks student immediately know what should do or read. After completing a task another signal is given to a student. When the students feels: "I have completed a task", the following action is to mark the task as "Done". This leads to another signal given by the environment. When the student sees that I have marked a task "Done" so I need to move to the next one." This means student will read the next task available. By repeating the process and steps student proceeds on a course.

While processing single task all other learning theories could be used. Student starts to study the subject with some previous knowledge as noted in Constructivism. Student seeks for new information by him/herself and assimilates it in his/her own way. This way the student can form understanding of the subject his/her own pace. Game development as well many other subjects are sensitive to the previous knowledge. Some people are better with programming while some other are more oriented for example in art. By giving students a chance to proceed on their own pace, nobody feels that the pace of teaching is too fast or too slow. This means that the amount of beginner students falling behind will be smaller and more experienced students will not feel bored because they can proceed in their own pace without any limitations. The process of student's behavior is described below in Figure 8.

(106)

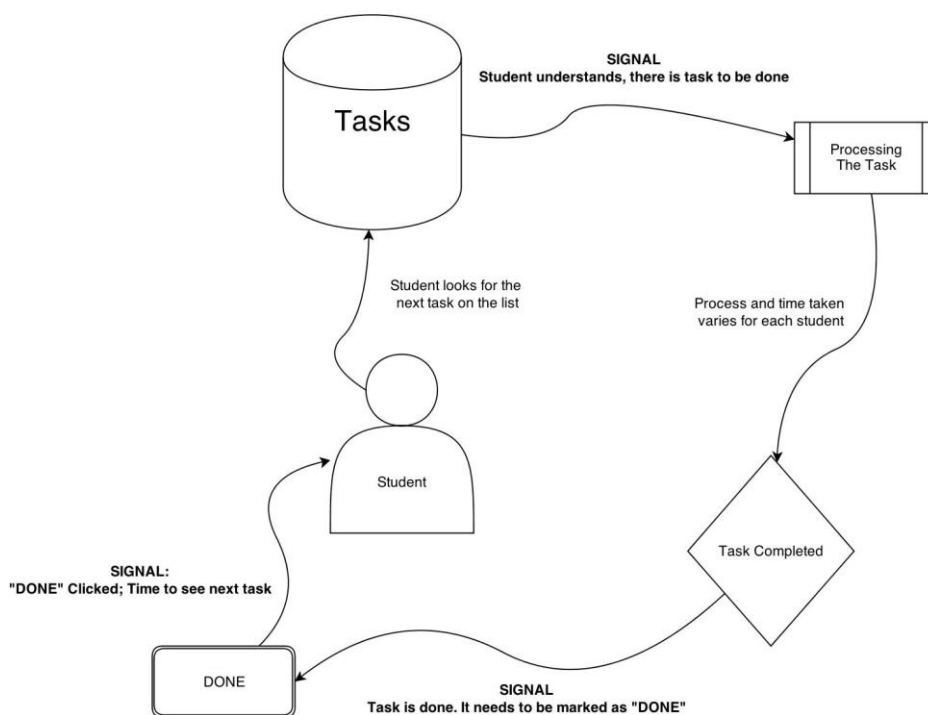


Figure 8: Student progression in TODO environment

The process of TODO system is surprisingly simple but. It is simply the design how working on tasks should be done. The student is guided closely with the tasks and seen as a passive person but when the tasks process is started the student needs to get active in order to learn. Processing leans to an idea that tasks are really small and simple to do and can usually be done quite fast.

TODO system also includes a front page board that acts similar to Facebook's wall. Everything that happens in the environment will be displayed in the feed. This way when someone registers, logs in, does a task, comments a task or does something else, it will be displayed in the board. Because TODO system stores timestamp of every action, it is easy to see which students have done which tasks. The common belief that students work at night was found to be true. It was not uncommon to see people in the middle of the night to make tasks in the TODO environment.

TODO site is also intended to be the download place for the client program of Metropolia Online. This means there needs to be a connection between the game database and the web-pages database for account purposes. First course used on the system is Game development with Unity. In the future maybe other courses could be held as well.

4.3 Metropolia Online, Massive Multiplayer Online (MMO) Game

Gamers spend more and more time on online gaming and almost all modern games have some kind of online multiplayer feature. Students who study programming, design or art and want to work in game development do not have a chance to get experience about developing MMO games because there is no organization that provides this possibility.

Creating a school's own MMO environment, Metropolia has a chance to stand out in competition on game education industry. This way students can get a valuable experience what needs to be considered when developing multiplayer game for masses. This also makes it possible to test various models, graphic shaders and A.I. designs in a multiplayer environment that otherwise would not be possible.

Usually game projects are one of a kind projects that have a clear start and end. However Metropolia Online is a game environment that keeps on evolving by students. This means that every time a student wants to improve the game, he/she can do it and the game can be used on various courses to experiment with programming, art and design.

Metropolia Online can be a good reference for graduates too. Students looking for a work in the industry can show things done on Metropolia Online as a reference. Working version of Metropolia Online can also act as a good reference for the whole school. Students from secondary education may feel Metropolia Online to be one crucial element when applying for a school.

Because World of Warcraft is so well known, it was decided to start with a similar concept so that students would not feel the game to be stange or difficult to understand.

This means that there should be a character in the game world that could be used to travel and explore in the virtual world. This kind of game is also easy to improve. 3D models and programmed features can be easily tested.

4.4 Course goals

Combination of TODO environment and Metropolia Online game was a massive attempt to create a new kind of virtual course. The goal was to split 5 credits course into a two different equal size parts. Both parts use TODO system and are almost entirely virtual. This way there is not limit for students on a course.

The first part was intended to be an introduction to the Unity as a game development tool. It teaches basics of Unity, programming, 3D modeling in theory and in practice. Students need to create a simple mediaeval style village and create some simple features so the player can interact with the game. Students also create their own 3D character that they can use to explore the world. This whole process teaches student the interface of Unity. Also 3D modeling and scripting is included. In addition rigging, animation and texture creation is needed to complete the first part. This way the student notices whether he/she prefers programming or graphics and concentrate on it.

The second part is about MMO development. Students use their prior knowledge from first part to create content and character to the MMO game. Additional features are recommended to implement to the game. Students need to at least put their medieval village to the MMO game. Also implementing their character to the game is preferred. This way virtual world can be expanded rapidly in the end of the course.

Completing the TODO list for both parts of the course will result a basic knowledge of Unity and game development. From there students can start to improve their skills and take part in game projects.

4.5 Course scheduling

The course starts with a contact lecture where goals and rules of the course are told. On the initial course there were around 65 students on the first lecture. In the beginning students are told to register and start use TODO environment. Of course all content for the course is already available and students can process as fast or slow they want. Freedom actually was one of the greatest aspects of the course. Because there were no limitations, some more experienced students completed the first part within 3 weeks while others had to study more and take the time to learn.

Every two or three weeks there was so called 'support class' where teacher was available to help on issues. In most cases there were only a few students in the class because using the TODO system seemed a great way to progress on a course.

The first part was completed in the 3rd period that started in the beginning of 2013. Because the course was marketed as virtual, the first lecture had a huge crowd of people. Quite fast after the first lecture rather many students quit the course right away. My own observation was that many thought that a virtual course does not need much work to be done or they thought the concept of the course would have been different. When students realized there is quite a heavy load of work to be done they chose to seek other options. During the first part about 30 students had to quit because various reasons while about 25-30 students came to the second part. Figure 9 shows estimation of activity during the course's first part. Student activity is based on observation of logging in and done tasks in the learning environment. The amount of students is based on feedback by students and teacher's workload is based on personal observations. It can be said that students were quite active. Activating student's trough learning environment by commenting tasks was easy and fast.

(106)

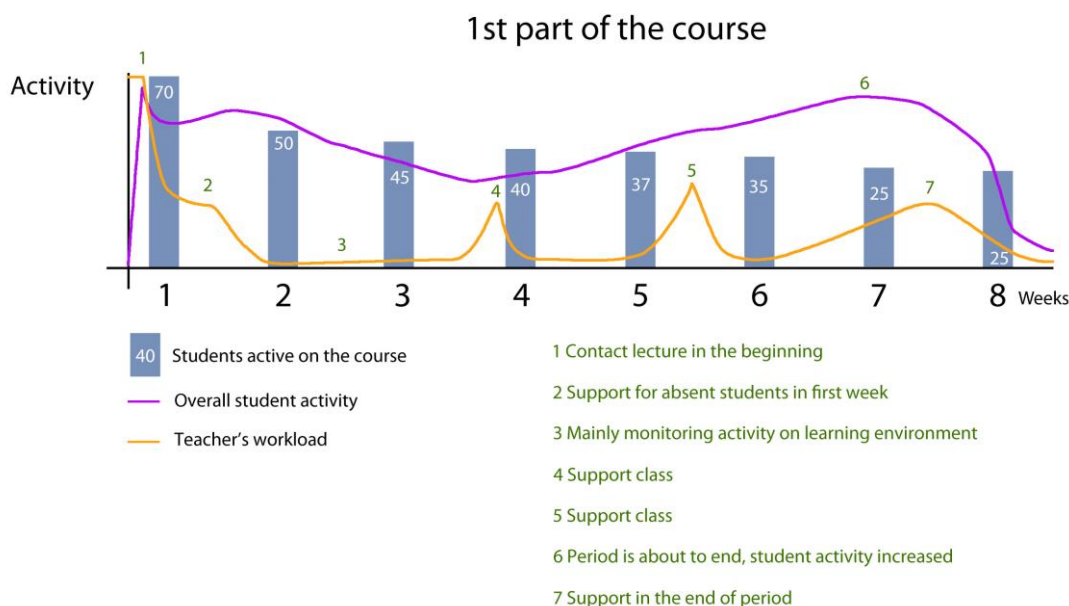


Figure 9: Activity on the course

The second part took almost entire 4th period starting late March 2013. Students started to study MMO files, code and models to be able to put content from first part to the MMO world. Again freedom of the TODO system enabled everybody work on their own pace. Some Villages and characters appeared in the first week while others needed the full period because of other courses, work etc.

Eventually 13 most hardworking students finally passed the course just before summer 2013. MMO game had over 10 new characters and 13 new villages to explore and also some additional features that can be improved. Grading (0-5) of the course was following:

- 0 (not passed) if student cannot create new content to the MMO game
- 1-2 if student can put his/her village to the MMO game
- 3-4 in addition can put playable character to the MMO game
- 4-5 in addition can create some new feature to the game.

For grading and passing the course it was not required to complete all little tasks listed in the TODO environment. Instead bigger goals were given for the whole course thus making grading easier for the teacher.

4.6 Course Challenges

From the teachers point of view the course was not that hard to handle. Because the course content was done well in advance and everything was placed in the TODO system, the teacher just needed to give students the address to the site.

Because there was no possibility to test the site with a big group some minor bugs appeared but those were fixed quite fast and easily. The first part of the course went through without many problems caused by the TODO or students. At some point there was a need to change server because helpdesk wanted to make some adjustments but overall the first part went smoothly. As a teacher I was actually surprised and based on students work they did not have many problems with the system. Of course making tasks and creating a game is hard work but that has nothing to do how the actual learning environment worked.

In the start of MMO part there were some issues with the game server and sub version system. There was a moment when the server shut itself down every night. Helpdesk policy is that students can not have access to the the server side so if server is not running, testing the game gets impossible. This meant the teacher had to be quite often awake to run the server. After some scripts were, the fixed server ran several weeks without any interruption.

Teaching MMO game development through virtual learning environment is a quite a challenge. There are so many things to know and remember to make different aspects to work together. Many students did not have any experience of sub-version control. Some students did not have problems with it while some were afraid to use because they did not want to overwrite someone else's files by accident.

(106)

Because several students were building one single project through the sub-version control, there were differences in file and folder naming conventions. Folder structure and other similar issues may seem minor ones but actually those were quite important because students did not know what folder to use for which purpose.

One issue was also about programming. Some students had programmed features to their village in the first part and these codes did not work well in the MMO environment. This led to multiple errors in the compiling process to the point where the game would not run. That is why it was important to make sure there were no errors until commit was made to the repository system.

Also, some students had their own pipeline to create content to the game. In the future courses there is a need to make a general workflow and file formats to get things work without problems.

One could think that the project of TODO system an MMO game would be a task full of overwhelming problems and challenges including running the course for tens of students. Actually this was not the case and the whole process went quite smoothly.

5 TODO learning environment

TODO Learning Environment is a database driven course system where a single course is divided to a large number of small tasks that guides the student on a course. The following chapter describes how TODO system is designed and what its purpose in game development courses is.

5.1 Original design and specification

TODO bases its design into real life management where various tasks are written as a list and executed when there is enough time and energy. TODO is designed to be a web page of course that holds large lists of various tasks. By carrying out these tasks the student processes on a course and learns new things and skills. Table 2 shows a list of various files and their purpose on the system. Most of the files are connected to

(106)

the database by including mysql.php in the code. Authorizing any page is done by including protect.php file. This way only logged in users can see protected pages. Protecting the pages also gives user information and current status of the user on the course. TODO is a purely virtual environment and students are required to register to see the content of the site. Enrolling on a course is required to be able to execute tasks on a list as seen in Figure 10.

File Name (Page)	Purpose
Index.php	Introduction page
mysql.php	Handles connection to the database on the system
login.php, protect.php, loginform.php, logout.php	Handles logging in and PHP session system using database
register.php	Inserts new user to the database
board.php	Shows activity feed. Content from database.
courses.php	Shows available courses. Handles enrollment to the course through AJAX call.
enroll_course.php	Retrieves AJAX call from courses.php and enrolls user to the course in database.
todo.php	Task listing taken from database. Comments are collapsed and can be

(106)

<p>save_comment.php, todo_done.php</p> <p>Chat</p> <p>download.php</p>	<p>opened using JQuery. Comments are stored to database by AJAX calls to save_comment.php. Completing a task is saved using AJAX calls to todo_done.php</p> <p>Makes database call and saves data to tables</p> <p>Chat is implemented from php-FreeChat (http://www.phpfreechat.net)</p> <p>Latest version of MMO game can be downloaded</p>
--	--

Table 2: Purpose of pages in TODO environment

Admin side works similarly but is a little simpler. Only adding, modifying and removing rows in the database are needed. This is done using various html forms.

(106)

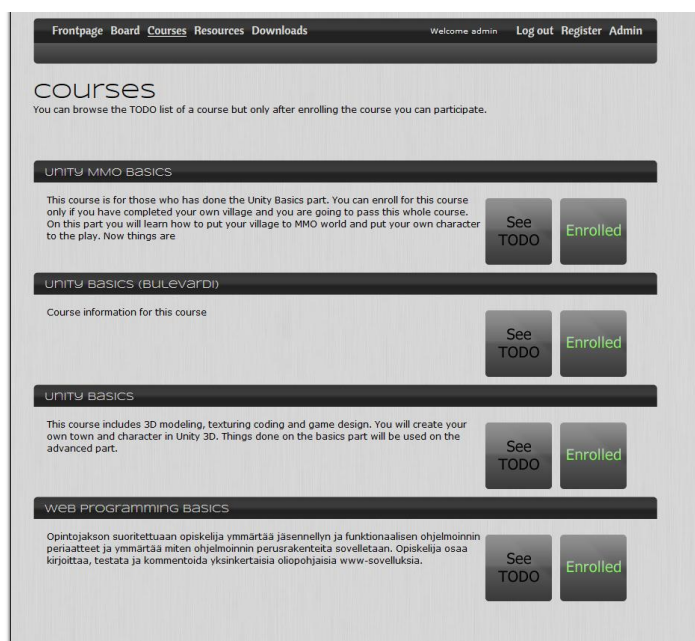


Figure 10: List of courses in TODO environment

TODO environment is divided to the frontend that students use and backend that is used by teachers and other content providers. In backend system a teacher can create, edit and remove content. Content includes courses, sub headers and tasks. One course is divided into several sub headers that cover some specific topic. Each sub header has a list of tasks.

TODO environment is done using PHP, MySQL database and JQuery library. Communication to the database is done using AJAX and PHP functions. This way working on the environment is smooth because no browser refresh is required when a database call is done. Various animations and smooth transitions with AJAX calls do make a difference. The environment feels much more professional and elegant when there is no constant refreshing of the browser.

A chat feature that displays automatically all users who log to the environment to see the list of tasks was added. This gives a small social aspect because the student can see who else is currently working on the course. Chat can be used to communicate and share information live during studying. Figure 11 demonstrates the functions of the task list.

(106)

The list of tasks can and will usually be rather long. From usability point of view it is sometimes said that a web page should not be longer than what the screen can display. But then scrolling down is a better option than having separate pages behind links because the user can not know how much content there is behind it. Scroll bar gives a good visual clue how long an article or the page is. In TODO case users will quite fast get used to scroll the list thus keeping usability easy. Also with consistent workflow users know how to work with the system since there should not be problems with using the environment. [65]

(106)



- ← Menu
- ← Information about progression
- ← Chat area common for all courses
- ← First Sub header
- ← Tasks
- ← Second Sub header
- ← Tasks (not done yet)

Figure 11: Functions of task list

(106)

There are not too many features in the environment which keep user interface simple and elegant. This also makes students to focus on essentials and not to explore the functions of the environment. This way most of the time is used on learning the subject and not the learning environment itself.

The most relevant features are following:

- Registration and login system
- Board for progress management
- Course management and enrollment system
- TODO list and tasks
- Commenting as social feature
- Group pressure for openness
- Easy to use and manage

One of the good points in virtual environment is availability. In theory, a student can access all content anywhere and anytime. The only problem is whether a student start to study in a bus or train or when there is just one hour time if a virtual environment is available. This does not happen often because the student probably does not know what to do next or if there is need for some research that is going to take time.

In TODO system all bigger parts are cut into small tasks that contain links, videos, articles, pictures and other similar content. Every task has an estimated duration that is going to tell the student how long it takes to complete a certain task. Every task has also a difficulty level that tells how difficult the task will be. This means that if the student has half an hour to do something, he/she can instantly go to see the list and pick a suitable article or video that does not take longer than 30 minutes. This way it is easy to start studying even on a bus trip. And completing the task during a way home gives

(106)

a feeling the bus trip was not wasted but something was accomplished something. This feeling is obtained when clicking “DONE” button on the task at home.

Each task has a possibility for comments. This is an important way to communicate, share knowledge and ask questions with other students who probably are struggling with the same things. On a course students started to comment tasks quite a lot and there was very valuable information shared among students. Again, there is nothing new in commenting or a discussion forum. In TODO system it is one of the few available features and it is easy to use. The threshold to start communicating by commenting tasks is low and there is real social activity going on. Various functions are done for each task and summary can be seen in Figure 12.

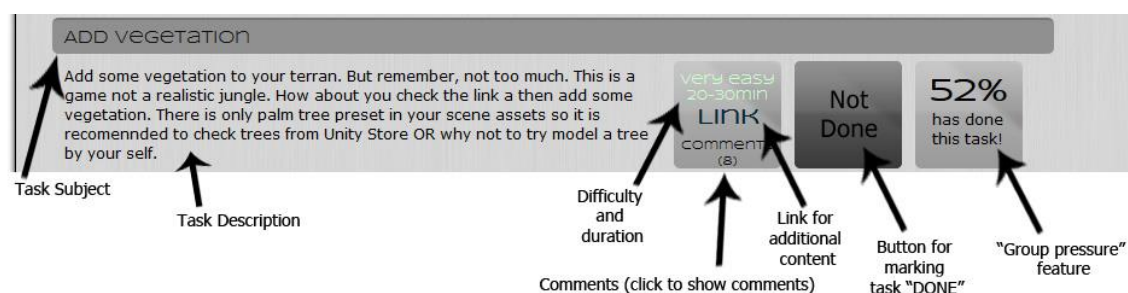


Figure 12: Functions of a single task

If the student wants to see comments he/she needs to click comments button that will display all comments. If there is a comment added to the task within 24 hours, the button will be yellow colored to indicate recent activity. The list of comments show who has commented and when and also the possibility to write new comment is given. Even rows of commenting are little bit darker as seen in Figure 13.

(106)

ADD VEGETATION

Add some vegetation to your terrain. But remember, not too much. This is a game not a realistic jungle. How about you check the link a then add some vegetation. There is only palm tree preset in your scene assets so it is recommended to check trees from Unity Store OR why not to try model a tree by your self.

very easy
20-30min
LINK
comments (8)

Not Done

52%
has done this task!

<i>villehur</i> January 7, 2013, 10:51 pm	For starters I used the BigTree that's in Unity by default. Then I noticed that it didn't have a collision component so I had to add it myself... Still experimenting with things and probably going to do the whole thing again when the first "island" is done. :P
unknown user January 15, 2013, 10:06 pm	I downloaded and imported the "Central European Tree pack" from the Asset store. It's free and pretty neat. But yeah, you could also make your own ;) (which I might attempt once I have a bit more experience with 3ds Max)
<i>miaprohon</i> January 15, 2013, 10:09 pm	uh uh, I guess that's a bug for you Admin! I'm the one who posted the comment above (about the tree pack), though I made that comment after a while and my session was apparently timed out. Identity theft!
<i>admin</i> January 15, 2013, 10:18 pm	Should be fixed now...
<i>pasiva</i> January 16, 2013, 10:14 pm	The very 1st download from the Asset Store did not work and I spent 15 min searching where did the bushes disappear... 2nd time OK with Terrain Assets. Not VERY easy.
<i>Annika</i> January 17, 2013, 9:10 pm	Had some problems with asset store trees here too. They were super small even after some scaling (did not help). Anyways created my own which was not that hard following the tutorial.
<i>rebecac</i> January 30, 2013, 1:26 am	I have been creating different trees and trying the wind zone option. The surprise is that if you want the wind to affect your forest, you need to add the trees one by one. Is there any option to apply the same options to a "painted" forest? :D
<i>admin</i> January 30, 2013, 7:57 am	Yes there is. Someone had a workflow for that. Maybe you need to make prefab from one tree and then use those. I think google is your best help.

Save Comment

Figure 13: Task with couple of comments

Comments written by currently logged user will be darker than others. This way user easily sees what he/she has written. As noted before, writing a new comment does not require to browser refresh thus keeping working on environment faster on a big list of tasks.

5.2 Development process of TODO

Development started with a list of wanted features. Also some user stories were written to cover most of the features. A good thing about the development was that I was doing it alone. This way I did not need to describe features to anyone, I just needed to program them myself. In the beginning there was just an idea of the system but no actual any proof that the TODO system would be a good learning environment. It was first an experiment, whether this kind of system would generate good experience to learn game development.

(106)

After most of the features on paper I created the database and required tables described in Figure 14. The design was supposed to be simple so only 8 tables were required for initial features. Registration, logging in and session management were the first features.

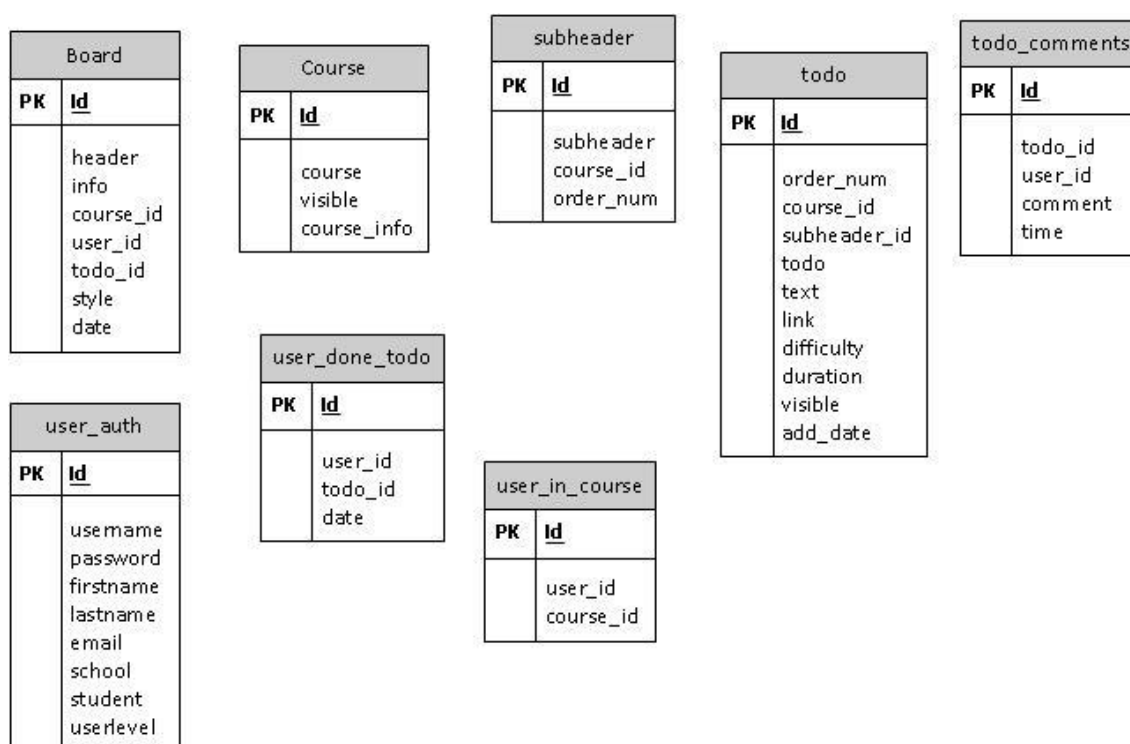


Figure 14: Database tables needed for TODO system

When the database was ready, registration and logging in was working, the backend was the next phase required. Because all course content is managed from the backend there was no need for frontend programming yet. In the backend there is add, delete and edit options for courses, sub headers and individual tasks. General html forms worked well for this purpose.

When the backend was working and all data can be put into the system the frontend needed some focus. Development order for features was created in the order how they appear to student. Usually the user first goes to the list of courses so that was the first one to create. There are two big buttons. If the user is not enrolled for the course, he/she can do it. Also, a button for seeing the TODO list is given and by clicking that, the user sees a list of tasks.

(106)

Course list and task list are fetched from database and then CSS is used to make it more appealing. Various background images, gradients and rollover images were done with Photoshop. CSS and JQuery were used for interaction.

When a user enrolls to the course, there is a row added to user_in_course table that tells what userd_id is enrolled to what course_id. Completing task is done the same way but now information is put into user_done_todo table. If a user wants to comment a task, a row is to put to todo_comments table that stores the task id, user id and the comment with timestamp. This way comments can be retrieved easily.

After the frontend was working there was a need for the board that works as a front page of the system. If a user interacts with the environment by completing a task or commenting it, there is a row added to the board table.

After the beginning of the first course, Information Technology at Bulevardi campus was also interested in the environment and the course content. There was a need to create another course with the same name and implemented a copy feature that copies all content from one course to another. This function will be handy in coming years when courses are held again. There is no need for inserting content all over again.

Because the environment is not too complex it was quite fast and easy to develop in chronological order. All the time I pretended to be a student using the system and if there was some feature that did not work, then I just focused on that one and programmed it to work right.

5.3 Group pressure

Group Pressure is commonly known as conformity is a social influence where one person's feelings, beliefs and actions are changed to fit in a group. This change in thoughts and behavior is affected by social pressure caused by a group. Conformity can also be defined as "yielding to group pressures" (Crutchfield, 1955). There are many examples in real life where group pressure is affecting our behavior. Reading forums on the web gives us the feeling that everybody has the same opinion even if

(106)

there are just a few who are strongly pointing their point of view. Another good example can be found among children. In most cases if a child starts to smoke it is caused by friends through group pressure. A non-smoking child wants to feel social cohesion that is going to lead to smoking. Group pressure may also have different forms like bullying, criticisms and persuasion. [66]

During the design process of TODO system Group Pressure was approached from the positive angle. If most students work for tasks and develop themselves it creates a pressure that pushes the rest to 'fit in'. This way giving up for group pressure can be seen a positive aspect and nobody should feel bad for giving up for the majority's influence.

One of the fundamental features of the TODO system is the Group Pressure percentage. For every task there is a percentage number that indicates how many students of the group had done a certain task seen in Figure 15. This way every student can follow the progression of everybody else as one entity. Calculation is simple. For every task the number of students who have done the task is retrieved from the database and this number is divided by the whole group size on the course. The number itself is simple and without use in most cases but when using it a lot and in right places it brings a whole new feel into the course.

This feature is created to affect the student's behavior in the virtual environment. There are lots of different personalities among us. Those who have a strong competition drive can work really hard and accomplish tasks much faster than many others. In the other end are lazy students who do not care much but they still want to pass the course. Group Pressure percentage encourages students to do the tasks on idea "Because everybody else has done that task too". In other words, if a lazy student sees a task that over 85% of students have done it but he/she has not, he/she might feel "guilty" and hopefully get the task done. This way working on tasks students will drive each other to work on more tasks, because usually nobody wants to be the last one doing the tasks.

For some students group pressure worked. Based on the feedback students liked that they were able to follow what pace of the whole group went. This way they were able to

(106)

adjust their working speed and observe what would be the coming work in coming days or weeks. Of course some students did not bother about the group pressure and worked in their own pace. The experience from using the virtual group pressure was encouraging and it is going to be used again in future courses. Figure 15 shows group pressure percentages.

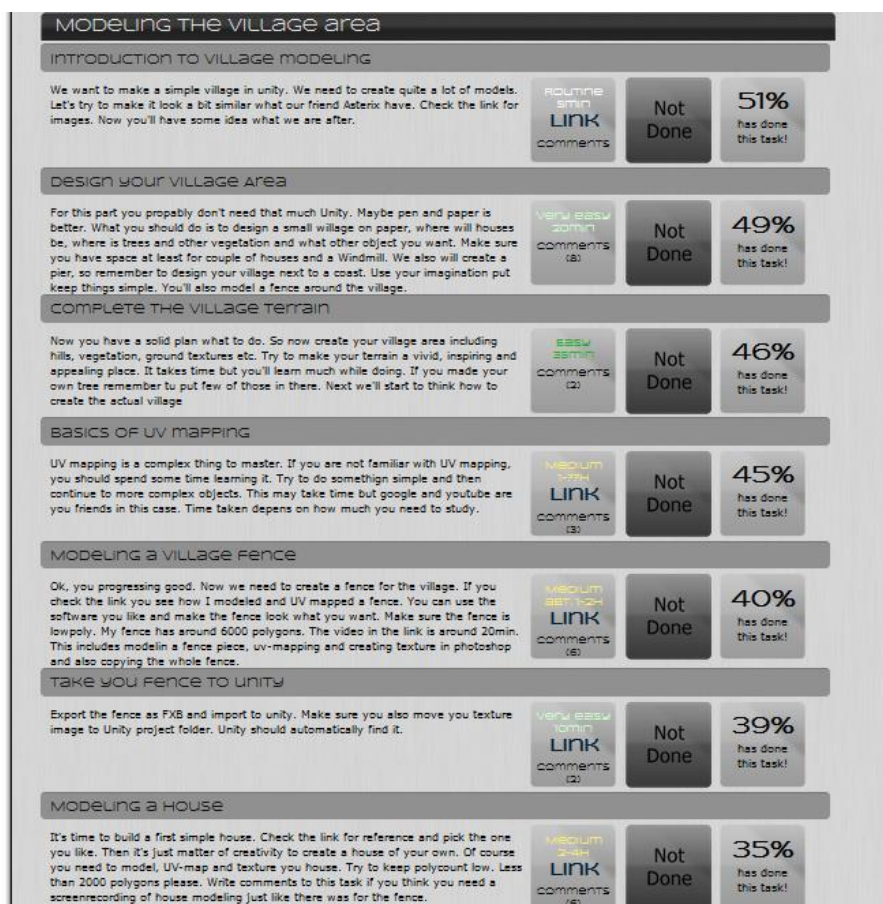


Figure 15: Group pressure percentages

As seen on the right side of the tasks, everyone can check at what task half of the students are going. If student feel, he/she do not want to be in the bottom half, then those tasks should be done as soon as possible or otherwise catching up is going to be more and more difficult if work is not started soon.

(106)

5.4 Creating a course using backend

The administrator needs to login to the backend and give just the name of the course and description for it. Creating a course is made really easy as well everything else. If the teacher wants the course can be left invisible. This way content can be created without it being visible for students. When all content is done and the course is about to begin, teacher just click the course visible and unleash students to study as shown in Figure 16.

ID	Course	Course info	Visible	Remove	Edit	TODO
8	Unity MMO Basics	This course is for those who has done the Unity Basics part. You can enroll for this course only if you have completed your own village and you are going to pass this whole course. On this part you will learn how to put your village to MMO world and put your own character to the play. Now things are	1	Remove	Edit	See TODO list
7	Unity Basics (Bulevardi)	Course information for this course	1	Remove	Edit	See TODO list
5	Unity Basics	This course includes 3D modeling, texturing coding and game design. You will create your own town and character in Unity 3D. Things done on the basics part will be used on the advanced part.	1	Remove	Edit	See TODO list
9	Web programming basics	Opintojakson suoritettuaan opiskelija ymmärtää jäsenellään ja funktionaalisen ohjelmoinnin periaatteet ja ymmärtää miten ohjelmoinnin perusrakenteita sovelletaan. Opiskelija osaa kirjoittaa, testata ja kommentoida yksinkertaisia oliopohjaisia www-sovelluksia.	1	Remove	Edit	See TODO list
10	Insiinööri työ	Tämä kurssi on insinööritöiden seuranta varten. Ilmoittaudu tälle kurssille vain jos olet aikaisessa tehdä insinööri työtä ja valmistua vuoden sisään.	0	Remove	Edit	See TODO list

Figure 16: Course list in backend

After a course is created it needs an administrator for content production. There are two kinds of user levels. Students are on user level 1 and teachers and content managers are on user level 2. For each user the user level can be changed from the user management by editing user information seen in Figure 17. There is also an option to put in a new password if it is forgotten.

(106)

Frontpage Courses Subheaders TODO Users Log out

EDIT STUDENT

[Edit User](#)

Username:

Password:

Firstname: ← Option to renew user password

Lastname:

Email address:

School:

User level: ← User level modification

Are you a student? Yes

Users

ID	Username	Name	School	Student	userlevel	Courses		
1	admin	Antti Laiho	Metropoli	0	2	Unity Basics Unity Basics (Bulevard) Unity MMO Basics Web programming basics	Remove	Edit
36	rajning	Rajini Gaddam	Metropoli	1	1	Unity Basics	Remove	Edit
35	nyvm	Gennikh Glass	metropoli	1	1	Unity Basics	Remove	Edit
27	fafase	Lucas Cosson	Metropoli	0	2		Remove	Edit
28	irakas	Sakari Lukkarinen	Metropoli	0	1		Remove	Edit
34	sys_argv	amaumo anthony	metropoli	1	1	Unity Basics	Remove	Edit
32	Falarier	Jukka Jokelainen	Metropoli	1	1	Unity Basics	Remove	Edit
33	samikuh	Sami Kuhmonen	Metropoli	1	1	Unity Basics	Remove	Edit
37	thwap	Sami Kusma	metropoli	1	1	Unity Basics	Remove	Edit
38	Annina	Annina Salmi	Metropoli AMK	0	1	Unity Basics	Remove	Edit
39	Mikkok	Mikko Karjanmaa	Metropoli	1	1	Unity Basics	Remove	Edit
40	xix	Antti Veräjänkorva	Metropoli - Leppävaara	1	1	Unity Basics Unity MMO Basics	Remove	Edit

Figure 17: User management in backend

Before sub headers to a course can be done, the course needs some administrators. Modifying the course administrators can be added just by selecting users who have user level 2. Figure 18 shows how administrator users can be put admins for different courses.

(106)

EDIT COURSE

Course: **Unity MMO Basics**

Course info

This course is for those who has done the Unity Basics part. You can enroll for this course only if you have completed your own village and you are going to pass this whole course. On this part you will learn how to put

Visible

Yes
 No

Course administrators

Antti Laiho
 Lucas Cosson
 Miikka Mäki-Uuro
 Minna Maltari
 Harri Airaksinen

Assigning course administrators ←

COPY course materials

Copy from:
Unity MMO Basics

Copy to:
Unity MMO Basics

courses

ID	Course	Course info	Visible	Remove	Edit	TODO
8	Unity MMO Basics	This course is for those who has done the Unity Basics part. You can enroll for this course only if you have completed your own village and you are going to pass this whole course. On this part you will learn how to put your village to MMO world and put your own character to the play. Now things are	1	Remove	Edit	See TODO list
7	Unity Basics (Bulevardi)	Course information for this course	1	Remove	Edit	See TODO list
5	Unity Basics	This course includes 3D modeling, texturing coding and game design. You will create your own town and character in Unity 3D. Things done on the basics part will be used on the advanced part.	1	Remove	Edit	See TODO list
9	Web programming basics	Opintojakson suositettuaan opiskelija ymmärtää jäsennellyn ja funktionaalisen ohjelmoinnin periaatteet ja ymmärtää miten ohjelmoinnin perusrakenteita sovelletaan. Opiskelija osaa kirjoittaa, testata ja kommentoida yksinkertaisia oliopohjaisia www-sovelluksia.	1	Remove	Edit	See TODO list
10	Insinöörityö	Tämä kurssi on insinööritöiden seuranta varten. Ilmoittaudu tälle kursseille vain jos olet aikaisessa tehdyssä ilmoittautumisessa ja valmistus vuoden aisaan.	0	Remove	Edit	See TODO list

Figure 18: Applying users as course administrators

Sub headers have got their own forms for creating and modification. Every sub header needs to belong a one course and those can be ordered as well as seen in Figure 19. This way content management gets easier if the teacher wants to reorder some sections for the course. In the list of courses only those courses where user is currently administrator are listed. If teacher knows all sections of a course he/she can put all sub headers once and then start to produce tasks on them. At the same time this is an easy way to organize course content. Sub headers can of course be created afterwards and reordered if needed.

(106)

ID	Subheader	Course	Order No.		
13	Controlling 3rd person camera and character	Unity Basics	6	Remove	Edit
12	Programming	Unity Basics	5	Remove	Edit
8	Introduction to system	Unity Basics	1	Remove	Edit
9	Getting started	Unity Basics	2	Remove	Edit
10	Creating a terrain	Unity Basics	3	Remove	Edit
11	Modeling the village area	Unity Basics	4	Remove	Edit
14	Open the Door	Unity Basics	7	Remove	Edit
15	Changing Scene	Unity Basics	8	Remove	Edit
16	Creating a real character	Unity Basics	9	Remove	Edit
17	Sign	Unity Basics	10	Remove	Edit
18	Additional features	Unity Basics	11	Remove	Edit
19	Building a game	Unity Basics	12	Remove	Edit
21	Controlling 3rd person camera and character	Unity Basics (Bulevardi)	6	Remove	Edit
22	Programming	Unity Basics (Bulevardi)	5	Remove	Edit
23	Introduction to system	Unity Basics (Bulevardi)	1	Remove	Edit
24	Getting started	Unity Basics (Bulevardi)	2	Remove	Edit
25	Creating a terrain	Unity Basics (Bulevardi)	3	Remove	Edit
26	Modeling the village area	Unity Basics (Bulevardi)	4	Remove	Edit
27	Open the Door	Unity Basics (Bulevardi)	7	Remove	Edit
28	Changing Scene	Unity Basics (Bulevardi)	8	Remove	Edit
29	Creating a real character	Unity Basics (Bulevardi)	9	Remove	Edit

Figure 19: Sub header management in backend

When sub headers are done it is time to create the actual content of the course as shown in Figure 20. Creating tasks is again just typing html forms. For a single task there are a little bit more options than for course and sub headers. The teacher needs to write a topic for the task and order number and then select the right course and sub header so the task is displayed on right course and in right section. After that the teacher writes the actual content of the task. This includes just description text and possibly a link to an article or other material. After that difficulty rating and duration is given. These are based on teacher's evaluation how difficult the task will be and how long it takes from student to complete it. This information tells the student in the front end whether he/she has time to do this task and does he/she have enough skills to complete it. After all information is given, submitting the form puts the content to database and it is displayed in the frontend for students to start working on the task.

(106)

Frontpage Courses Subheaders TODO Users
Log out

ADD TODO

TODO

Order number

Courses
Unity Basics

Subheaders
Controlling 3rd person camera and character (Unity Basics)

Text

Link

Difficulty
Routine

Duration

Visible
 Yes
 No

TODOS

ID	TODO	Order No.	Course	Subheader	Text	Link	Diff.	Duration	Vtc.	Date	
31	Design your Village Area	2	Unity Basics	Modeling the village area	For this part y...		2	20min	1	10.12.12. 21.32	Remove Edit
19	Knowing you environment	2	Unity Basics	Getting started	This is going L...	Link	2	Max. 60min	1	05.12.12. 9.37	Remove Edit
18	Installing Unity	1	Unity Basics	Getting started	You will need U...	Link	2	15min	1	05.12.12. 9.35	Remove Edit
14	Todo links	2	Unity Basics	Introduction to system	Each Todo may h...	Link	1	2min	1	05.12.12. 9.06	Remove Edit
15	Check the Board	3	Unity Basics	Introduction to system	The board works...		1	3min	1	05.12.12. 9.11	Remove Edit
16	Comment this task	4	Unity Basics	Introduction to system	You can also pu...		1	3min	1	05.12.12. 9.22	Remove Edit
17	Almost Done	5	Unity Basics	Introduction to system	Now you are fam...		1	3min	1	05.12.12. 9.26	Remove Edit
20	Hey Friend, you have a problem?	2	Unity Basics	Getting started	No worries, the...	Link	1	5min	1	05.12.12. 9.42	Remove Edit
21	How about 3D	4	Unity Basics	Getting started	You need to lee...		1	5min	1	05.12.12. 10.01	Remove Edit
23	Stuff have 3 dimensions	5	Unity Basics	Getting started	Now you task is...	Link	3	15min	1	05.12.12. 10.14	Remove Edit
24	Creating a new project	1	Unity Basics	Getting started	Projects in Uni...	Link	2	5min	1	05.12.12. 10.25	Remove Edit

Figure 20: Creating tasks in backend

Editing and removing content of a course is rather simple because all content is shown as a list. The easiest way to work on course materials was to keep two browser windows open. One window was back end side where content is created and modified. The other window was opened to the front end thus giving the actual look of the course and what students are going to see.

5.4.1 Course content

TODO system supports content that can be split into small and simple parts. Content creation needs to be started well before the course will start because planning all small tasks is not done fast. But when there is a plan of all tasks then input to the system goes rather fast.

On the first course Unity Basics teacher needs to know or study the subject well and create all course content once, so that he/she is able to split everything into small pieces and write instructions how everything should be done and what common mistakes should be avoided. In this case there was a need to study different workflows between 2D and 3D programs and Unity. Also work on the programming side of Unity was needed as well as on creating tasks that help students to understand basic C# programming.

When workflows were ready the models, characters, textures, and lot of code were created completely ready. Used time was recorded because when creating course from scratch there cannot be too much content. First part was about 2,5 credits so students need to complete it in about 60 hours. I solely created all content within 45 hours and it seems it takes at least twice the time from students to do same by following instructions.

After all content were done, some of the content was done again but now with instructions that includes screenshots, written explanations and screen recording videos. Also many links to external sites and tutorials were searched to complement one's own learning material. Just for practical reasons there was no need to create some tutorial that someone else has already done and which was freely available. No need to invent a wheel again.

The process creating content for a course to the TODO environment could be done in following phases:

- Plan course sections

(106)

- Create every single code, model and asset required for the course. Record time taken
- Search for additional content like tutorials, articles and hints that help students on completing the tasks
- Start from scratch and do all content again with instructions and proper teaching material.
- Write required phases in small parts on word document. These are the tasks. Every task requires following info
 - Topic
 - Description
 - Link
 - Difficulty
 - Duration
- After word document is complete with all tasks, copy and paste them to TODO environment.
- Make course visible to students

The course content was put as a list format to word document where it could be easily copy pasted to the html forms and from there to the database. In the future there are plans to create a feature that all content can be put to excel and all data can be inserted to database just by uploading an excel. This speeds up the content inserting phase. Though all content needed to be written to forms by hand, creating almost 100 tasks were done in less than 15 hours.

5.4.2 Task content

A single task content should vary a lot. This means that there should be really simple tasks like “Read this article” or “Watch this video” that are more theory based and then more practical tasks like “Program lever to rotate when clicked enter”. This means that

(106)

students see a versatile list of tasks with different topics, durations and difficulties. This way list brings variability to studying and the student does not get bored so easily. Although the student sees a difficult task he/she can still accomplish easier and faster tasks.

A single task should contain good introduction text to the task and a link to external site is preferred. Difficulty and duration tells student that how much time and effort is needed to complete the task.

Task does not need to be just for studying. Comments section can be used for various purposes. With tasks there could be questions such as: "How much do you have experience in C# programming? Answer in comments". Because commenting is easy, students easily write some comments and this way social cohesion is increased while using a virtual environment. Feedback and other possible questions tell the teacher a lot how students feel and what they like about the course and which tasks have been difficult. Later on based on this information teacher can add tasks to supplement some topic.

5.4.3 Social activity

Because social media is such a common element in modern society, virtual learning environments should support it as well. Content sharing, liking, commenting and discussion should be one of the first things when developing these kinds of environments.

TODO environment currently has a commenting possibility, chat and board for an activity feed. Because learning in a virtual environment is not limited to time and place, having various tools to communicate and monitor others students activity is important. Connection to Facebook and Twitter or other mainstream social applications are not that important because many people want to keep studying and personal life separate. That is why social features are built into the TODO system. This way everybody know they will not share their personal information but only information about their studies.

5.5 User experience

Most of modern games have a tutorial phase in the beginning. This way the player can learn the basic functions and purpose of the game. Easy learning curve usually brings better user experience later on. The same approach is working on a TODO system. For the first course there were four simple tasks created just to introduce user to the functions of the TODO system. The first one was the introduction of the link in the task, the second was the board, where all activity is listed. In the third one the user is suggested to write a comment to the task and then mark the task as done. The fourth task is just introduction to the rest and explaining the Group Pressure feature.

After these four steps the user is ready to start studying and can continue to the actual tasks of the course. User experience is intended to be a little different each time the student logs in to the system. Because on one course there are usually couple dozen students and all the time usually someone is working on a task, commenting something or marking something done. This means that every time a student logs in to the system there will be new things on the board and there are new comments and the Group Pressure percentages have changed as well. This gives the feeling the student is working for the course as part of a group and it brings the feel of having a small community even if it is a virtual one.

It is up to the student and his/her personal preference do he/she want to answer a comment, participate in chat or just do the tasks on one's own pace. This way the student is not forced to do anything just encouraged.

Good user experience requires well designed user interface. There are two important aspects in user interface. First one is that because there are not many functions in TODO system, not many buttons are required. Therefore the buttons and information can be designed a little bit bigger than usual. This mans there are no hidden menus and the student does not need to find the required function.

Another important aspect of user interface is lack of icons. Many programs and websites have lot of icons. In many cases a small square icon does not tell the user what the button does. This requires that the user will need to move the mouse on the button

(106)

to get more information or try it to see what happens. In TODO, all buttons are text based because it instantly tells the user what the button is supposed to do. Figures 21 and 22 show an example of PEPPI system first with its current menu buttons that are based on icons and after that icons are replaced with text.

The screenshot shows the PEPPI system interface. At the top, there is a navigation bar with the TUUSI logo and the PEPPI title. Below the navigation bar, there is a main content area divided into two columns. The left column contains a section titled "Viimeksi tapahtunutta" (Recently happened) with a list of events. Each event entry includes a yellow arrow icon, the event name, the date, and the person responsible. The right column contains several sections: "Tehtävät" (Tasks) with "Ei tehtäviä" (No tasks); "Viestit" (Messages) with "Ei viestejä" (No messages); "Ohjeita käyttäjille" (Instructions for users); "Toteutussuunnitelman tallennus" (Implementation plan storage) with a link to "Lue ohjeet" (Read instructions); "Onko järjestelmässä virheitä?" (Are there errors in the system?) with a link to "Lähetä viestiä Helpdeskiin" (Send message to Helpdesk); and "Päivityslokki" (Update log) with a link to "Yhteenveto korjauslistalla olevista virheistä ja puutteista ja päivityslokki" (Summary of errors and deficiencies in the correction list and update log).

Figure 21: Peppi system with original menu

The screenshot shows the PEPPI system interface with a text-based menu. The navigation bar at the top includes the TUUSI logo and the PEPPI title, followed by a menu with text-based buttons: "Etusivu" (Home), "Henkilöt" (People), "Opetussuunnittelu" (Curriculum planning), "Toteutukset" (Implementations), "Oppimateriaalit" (Learning materials), "Resurssit" (Resources), "Raportit" (Reports), and "AHOT". The main content area is identical to Figure 21, showing the "Viimeksi tapahtunutta" section and the right-hand navigation and information sections.

Figure 22: Peppi system with text based buttons. Image is 'photoshopped'.

(106)

It is up to personal preferences which one is better. In Peppi's case there is a lot of space in the menu bar so why not to make use of the space. Big buttons were also put to the TODO system. This way user instantly sees the most important things and does not need to think or look for a button or action. Big buttons also are better also when students are studying using mobile phones.

5.6 Student progression and Teacher's role

When all course material is put to the system and the student is given permission to do tasks, everything gets rather simple. Student does not need to worry anything else except logging in to the system, enrolling on the course and starting to do tasks. In most cases student will start progressing fast in the beginning. This is mostly due to the novelty of the system but at some point working with the system will get steady and the pace of doing tasks will reduce a little. If tasks are done right, student will use his/her previous knowledge and skills to complete next tasks. Tasks are getting more and more difficult from different areas but doing tasks should be fun and completing tasks by clicking "DONE" should give a "Go Up a Level" feeling thus encouraging to do more tasks.

The Student can do tasks where ever he/she feels good tasks are available. Some tasks require previous tasks to be done but some are theory based can be seen or read despite other tasks. Because the student sees the difficulty and estimate of the duration of a task, he/she can start from easier ones and gradually continue to the more difficult ones. Of course the whole tasks list is designed so that student should be able to go through tasks in straight order without making any selective decisions.

The Student can all the time see how many tasks are done and the percentage of the tasks done. Every time when logging in, the student can see from the Board who has done which task which is one way to keep group pressure on.

From teachers point of view most of the work is done before the actual course. When students start to work on tasks teacher will work as a supervisor and monitor percentages and comments and give guidance on the chat. Teacher can put tasks in to the

(106)

system that indicates deadlines; for example “This task should be done in the end of week 4”. This way students know are they in right pace with the course.

Because TODO system works well basically with any number of students, it does not matter for teacher how many students there are on a course. Even if a student wants to join on a course later, it does not generate work for the teacher. Latecomers can be accepted and instructed to log in to TODO system and start doing tasks.

If students give feedback that some task is difficult or there are problems, the teacher can add, remove or ease tasks during the course. New tasks will appear to their slot and students can instantly benefit from the new information. This way there is no written communication, like emails or announcements needed because everyone sees new tasks they have not done yet. Monitoring the course makes the teacher able to modify the speed and direction of the course on the fly.

Some contact teaching is preferred because on those classes the teacher can quite fast solve problems that might take longer time in virtual environment. On the initial course attendance on support classes was quite low and most of the students were happy to work with the environment.

6 Massive Multiplayer Online (MMO) platform

Online gaming is one of the most common ways to play games these days. Almost every new game has a multiplayer game feature. Some games do not have single player mode at all and they are relying only on online gaming. Such as WOW, LOTRO, EVE Online and World of Tanks that are really popular.

If Metropolia had its own online game that could be developed by students, Metropolia would have the ability to provide experts to companies who are willing to create pure Online games. That is why a research was made within Game Cluster project to find out whether this kind of system could work within school environment.

(106)

Metropolia Online is a working platform for game design, programming and art that works purely online. Students develop it and are able to learn from others. Following chapter is going to describe the basic idea and design of the game. Also technical decisions are explained and main features of the game are listed.

Figure 23 shows the overall design of MMO game and TODO learning environment and how they are connected. The main focus is on the master server that keeps MMO game running and stores repository of the latest version of the game. Students retrieve the game files to local computer using version control system and develop it further using Unity game engine.

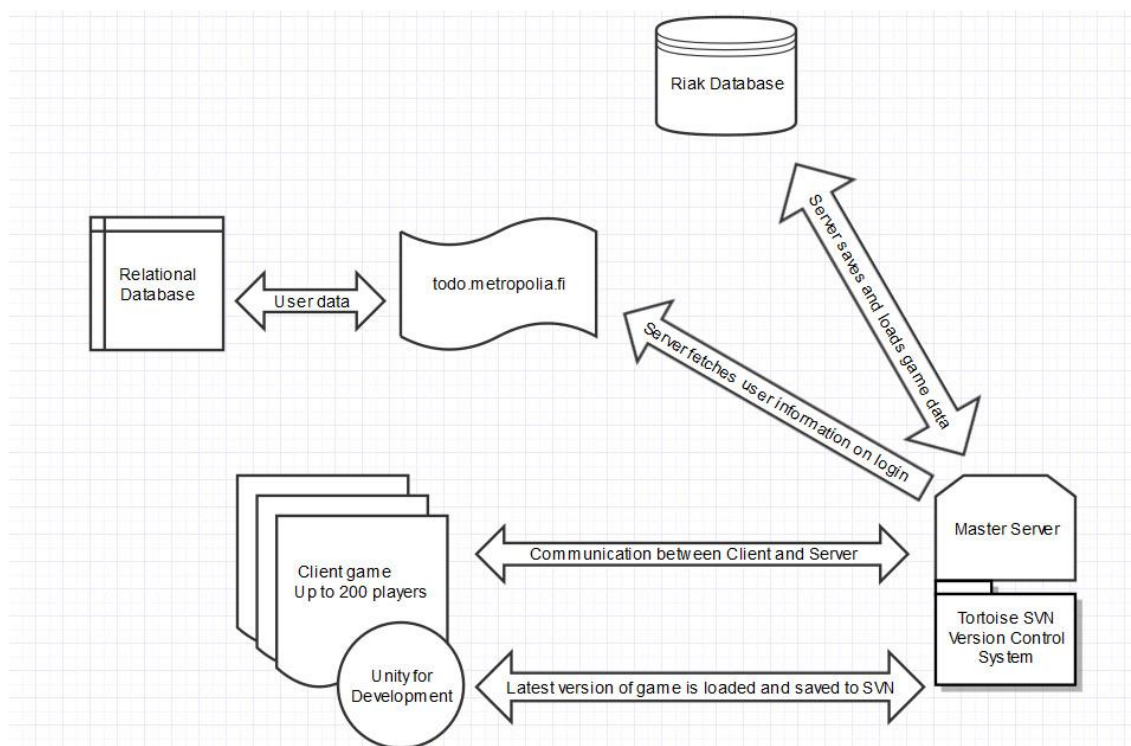


Figure 23: System architecture

Running client game in Unity creates connection to the server, meaning fetching login information from the database for TODO environment. This way logging to the learning environment and to the game can be done with the same account. Riak database handles live game information and modifications for example location of characters. More detailed and technical list of the system are found in chapter 6.2.

6.1 Metropolia Online Background and Design

The Game Cluster project has made it possible to research on the Massive Multiplayer Online games and development process. Game development requires so many various abilities that there should be a platform which could be used to test and enhance all of them when needed. Working on different projects is a good practice for students but if they have a certain feature or idea they would like to test it would require an environment specially dedicated for that.

In recent years different game engines have developed a lot and because they are easy to use, they can be used almost in any game development situation. Good examples of game engines are Unity and UDK. Unity was chosen to be used in this project. The reason was that other game development courses at Metropolia use Unity as the main tool. Prototyping various design ideas in Unity is quite easy and there is no need to start to program a game engine that would require a lot of effort and time. [46, 47]

In the beginning the whole Metropolia Online project was fully an experiment as well as a research process to find out whether we could develop a system that can be used in various courses. The Actual design process and gathering ideas was done during summer 2012. Visiting Siggraph conference in August 2012 was one part of this designing process. After summer the design was clear and Michael Sundqvist was hired to help with the programming part. Development cycle was set for 6 months because the first Unity development course would start in January 2013 and the MMO part in at the beginning of March 2013. Everything was good on paper and in theory the designed online game with MMO game should be achievable within the time frame.

The process of developing Metropolia Online along with the TODO system was easy and hard at the same time. The idea of making an online game is something that should not be taken lightly because of amount of work is big. Browsing the web questions such as "How can I make an MMO game?" can be found on forums. There are many Massive Multiplayer gamers who are willing to do their own game. For the players the game itself may look rather simple but often there is usually much more effort made than a normal player can probably understand.

(106)

Understanding all aspects required in development was quite difficult. After describing the design and required features Michael was able to start developing the groundwork for the game while I was working on the TODO system. With only two people working on a project made the managing process easy.

The basic design for Metropolia Online was purely copied from popular Massive Multi-player Online Role Playing Games (MMORPG). The game would consist of a server and a client that can be downloaded from web site. the server would run on a dedicated virtual server with a couple of Riak databases. Riak is discussed in more detail in chapter 6.1.4. Client program is the actual game that can be downloaded from web site. The game can be played with it on a local computer with online connection. This means that after signing in the player can choose a character, join the world and can explore everything freely. If and when there is more people in the game at the same time, the game shows another player's location and movement in real time. If all this could be done before the course starts around March 2013, there would be a possibility to start developing more features for the Online game on the course and we would have a proof of concept.

6.1.1 Server - Client approach

Usually MMO games have one server or many servers where a downloadable client game will connect. The client game is a simpler graphical representation of the server's current status and the content that is in the database. The load and requirements coming from clients are distributed usually to a cluster of servers and databases that handle the data. They then send back information to the client to show how the game looks after the change.

Basic move command will work through the following steps:

1. Player presses move button on keyboard
2. Client sends movement command and coordinates to the server. Coordinates are sent to server 15 times per second

(106)

3. Client moves local character on screen
4. Server analyzes the command and checks which clients are in the same scene.
5. Server send response to these clients and those client programs reflect to the original movement. Other clients see the moving player but the character is a proxy model. It acts more like a ghost but is moved in asimilar way as the original. This way an illusion of all visible characters is made.

In this project database connection was not needed because the server did not need it. However if the user picks up an item or there is a change for example in health parameter, a database request is made. With server and client approach there is always some kind of lag. This means that it will take some time to take the command from one client through server to another client. This is sometimes compensated with a technical solution but Metropolia Online does not need such solution in this initial version. Every 5 seconds the location of the player is stored to the database. When the player quits and comes back to the game the last location is used to continue the game.

6.1.2 Unity

Unity (<http://www.unity3d.com>) is a game development ecosystem. It is popular and highly used among game development companies. Unity has 2 million registered developers and 0,4 million are active every month. Around 50% of developers use Unity for game development. Important things for developers are rapid development time and flexibility. These properties match well with Unity and that is why it is a suitable tool in a school environment where fast prototyping and learning is important. [46]

Unity is used by all companies from small developers to big international organizations. Rovio's Bad Piggies was done by Unity which proves that there is demand for Unity developers in Finland, too. Unity was chosen as a tool in game development courses because it is easy to use and has many other good qualities as well. [48]

(106)

Unity uses mentality “Develop once, deploy everywhere”. This means that the game needs to be developed inside Unity and it can be published most modern platforms. Games can be published on various platforms such as PC, Mac, iOS, Android and Windows Phone. This is an important feature for a developers who want to achieve a wide audience for their game. Programming is done by using JavaScript, C# or Boo and languages can be mixed with different scripts. Languages are used and run on the Open Source .NET platform Mono. [46]

Unity has an asset store where content can be purchased to speed development. Additionally various plugins, tools and service are available through the web. One example is Mixamo’s auto rigger and animation service where the user can rig a character in a few minutes when usually rigging takes several hours in a modern 3D program. Mixamo can be found on <http://www.mixamo.com/>.

The basic version of Unity is free, giving students the possibility to use it at home for learning and game development. The professional version of Unity gives additional features costing 1140€.[46, 47, 48]

6.1.3 Photon

Photon (<http://www.exitgames.com/>) is a development framework for Massive Multi-player Online games. When the development of Metropolia Online was started, a working framework was needed to handle the database and server side processing. Photon server was chosen for this task and work was started to make it compatible with Unity and required features.

Core of the photon is C++ but the actual server code is C#. This way programming the server side would be a little easier. The server handles the data management and connection towards to the database when a request comes from the client game. In theory any database could work with Photon but MySQL was chosen in the first iteration. Server would run on a windows server and client program would be edited inside Unity.

The problem with Photon was that programming the server could not be done inside Unity and it would require separate environment for development. This would be much

(106)

more difficult for students and learning the server programming could not be possible in a sensible timeframe.

Quite a lot of effort was made to get Photon framework, MySQL database and Unity to work together and there was actually a really well working concept where multiplayer activity was possible. After a while more research was done to find a more suitable framework to work within Unity. The requirement was to find more simple solution that would not need so much server side code and which could also be taught to students if needed. After little more research uLink was discovered and it was an easy decision to change framework. uLink made the server side much easier to understand. [50]

6.1.4 uLink, noSQL and Riak

uLink (<http://muchdifferent.com/>) is designed to work only with Unity while Photon was more like a general approach that could work in various situations and development environments. But because we had chosen to use Unity as a main tool, uLink was a good choice because it is designed only for Unity and everything else is left out.

Quite fast we found that there were many features in uLink that made development easier. uLink is fully C# code and it can be edited within Unity editor. The port from Photon to uLink required some extra work but not all code was required to write again. Instead the server side handling became quite a lot simpler and lot of code rows were deleted. uLink uses the same design as Unity's built-in system but it is more efficient and has more features. uLink has over four times bigger API than Unity's own Network system. This means uLink has some readymade building blocks that can be utilized easily which means that less code needs to be written. uLink has similar features as Photon had with the easier of use added. Another benefit compared to Photon was that Mono editor can be used on server side and code from client and server can be re-used within Unity. [51]

uLink supports well First Person Shooter (FPS) games and also various MMO games. Basically any multiplayer game type can be created combining Unity and uLink. During the change from Photon to uLink the database was also changed. Relational databases such as MySQL do not scale well when a lot of data and fast requests need to

(106)

be made. Change was made to NoSQL because it is an answer to the requirements needed regarding the scale and agility. It supports cloud based development and modern environments that are required when creating MMO games. [52]

With relational databases the structure needs to be determined before the data can be put to the database. This schema is required and changing it may be difficult during the development. This kind of database does not fit well in a MMO development because data can vary a lot and there will be lot of information that is changed by users all the time. In time database will get rather large and handling the schema would become harder. NoSQL does not need schema to be determined before and it scales well for any situation. This leads to faster development and it can be integrated to the code easier. Also database administration is not needed that much.

Flexibility is the keyword in NoSQL. Traditional relational models data must be read from multiple tables and then the results must be combined. Also writing data to database must be done to several tables. NoSQL has a different approach. Document-oriented database can take storable data and form a JSON format document that is then stored to the database as one object. This JSON document may contain a row of information from several traditional tables. Adding data to JSON documents does not need any changes to the structure making modification easier later on. [67]

Relational databases are usually run on a single server that does not scale well and brings various problems to the design. In the case of Metropolia Online three NoSQL Riak (<http://basho.com/riak/>) nodes were created. Riak is a distributed database architecture that uses noSQL and replicates data to all nodes and is available all the time. Riak uses key-value pairs in its structure where value can be basically anything from single string to a complex object. [52]

Even if one of the nodes crashes or loses data, Metropolia Online would not lose anything because all database nodes are identical. Scaling works well, because adding new Riak nodes is easy so expanding the game is not a problem. If a new node is added to the ring of Riak databases current nodes immediately start to copy data into the new one. After all data is copied to the new node, tasks can be shared to the new node and efficiency is increased. Riak has a built-in load balanced which automatically

(106)

stresses the right node to retrieve data for fast results. If one node shuts down or gets an error during storage process, other nodes will take the workload from the broken one and when the node is fixed it will continue its work. [52]

After the completion of uLink and Riak nodes there was a well scalable, agile and easy to use Massive Multiplayer Online game architecture that could be edited within Unity.

6.2 Final Technical structure

The technical structure for Metropolia Online require various servers and ports to be opened for communication from outside school network. The following list sums up the technical requirements for MMO game that should scale at least up to 200 simultaneous players. This was a rough estimation if one user used approximately 250kB/s of bandwidth. Figure 23 in beginning of chapter 6 shows how databases, server and clients are connected to each other. Below is a technical list of servers, databases and connections to show how the system is built.

- **3 pcs Riak Nodes (Database servers)**
 - Static IP addresses
 - Operating system Linux, preferred 64-bit ubuntu/debian
 - Remote control for software installation
- Open ports required for communication between Riak nodes
 - TCP: 6000-7999
 - Epmc listener: TCP 4369
 - Handoff_port listener: TCP 8099
- Ports open for client program
 - Web_port: TCP:8098 (This way connection between todo.metropolia.fi and client game is made when new user is registered to the game.))

(106)

- pb_port: TCP:8087 (This way Master Server (below) will connect to database)
- **Master Server**
 - Current estimation is that maximum of 200 users can be used simultaneously. This means that bandwidth by every user may be around 125kB/s when position information is delivered between users.
 - Additional bandwidth required for other tasks may be another 125kB/s resulting 250kB/s of total bandwidth.
 - Static IP
 - Operating system Windows Server 2008 R2 + SP1
 - Access from networks outside school through port UDP 7100
 - FTP account for transferring files to the server
 - Remote control for installing software
- Required ports
 - Server listener port: UDP 7100 (Players connect through this port)
 - Database port: TCP: 8087 (Through this are connected to Riak-nodes)
 - HTTP, FTP, SSH ports, if needed at some point

For school purposes three Riak nodes are probably too much but adding or removing them is easy and fast afterwards. The more nodes are available the better is the performance of the database is. Master server is running as virtual server and can be accessed remotely if it is crashed or shut down.

6.3 Sub-version Control

Tortoise SVN (<http://tortoisesvn.net/>) was chosen as the Version Control system for in Metropolia Online, because Metropolia Online is a one big project that is modified by tens of students nearly the same time. When students create new content, modify a

(106)

scene or write new code, everything needs to be shared. Updating everything manually would not be wise so a version control system was needed for the project.

Again we aimed to find a simple approach for the version control. Tortoise SVN was chosen because it has an easy interface and the functions are simple to use. Tortoise is an extension to windows explorer giving similar tools that students usually use. Only commands that can be currently used are available and only a couple of commands are needed to update and retrieve versions from the repository (REPO). This way it is easy and fast to teach to students to use version control system. [54]

Version Control was set up to the server so anyone could get access to it when needed. At the beginning of the course students download the latest revision on their computer and then start updating content to it. After something new is made for the project, a commit process is made to the repository and everyone can retrieve recently made updates. Every time someone commits something to the repository modifications need to be commented with name and descriptions so that the teacher can track who has done which modifications and resolve problems if something is broken in the game.

One important thing in school environment is that no broken revision is ever committed to the server because when some other student retrieves the content, there may be several errors appearing and all students have to fix the same problem before they can continue to develop the game. That is why all code must be compiled without errors and tested before committing to the REPO.

An additional feature was made to sub version control. Every time a commit is made to repository, there would be a database insertion to the TODO system that displays current revision number, date and comments given. This again brings on one little aspect that integrates TODO system to the MMO game development. [54]

6.4 Main features

To make the game logic and functions better understandable for the students, the game required some basic features where improvements and additions can be made.

(106)

There were only a few of them but they were important for development to start. These features were defined during the design process well before the actual development was started. First there was one important requirement, namely that the whole system should work. This means that if a player is in the game world it can see all other characters. This was the first requirement which makes whole game to resemble MMO.

Before a player gets to the game a login screen is displayed as shown in Figure 24. User needs to log in with the same account that is used on TODO learning environment. This way the game recognizes the player identity and can retrieve player's characters from the database.

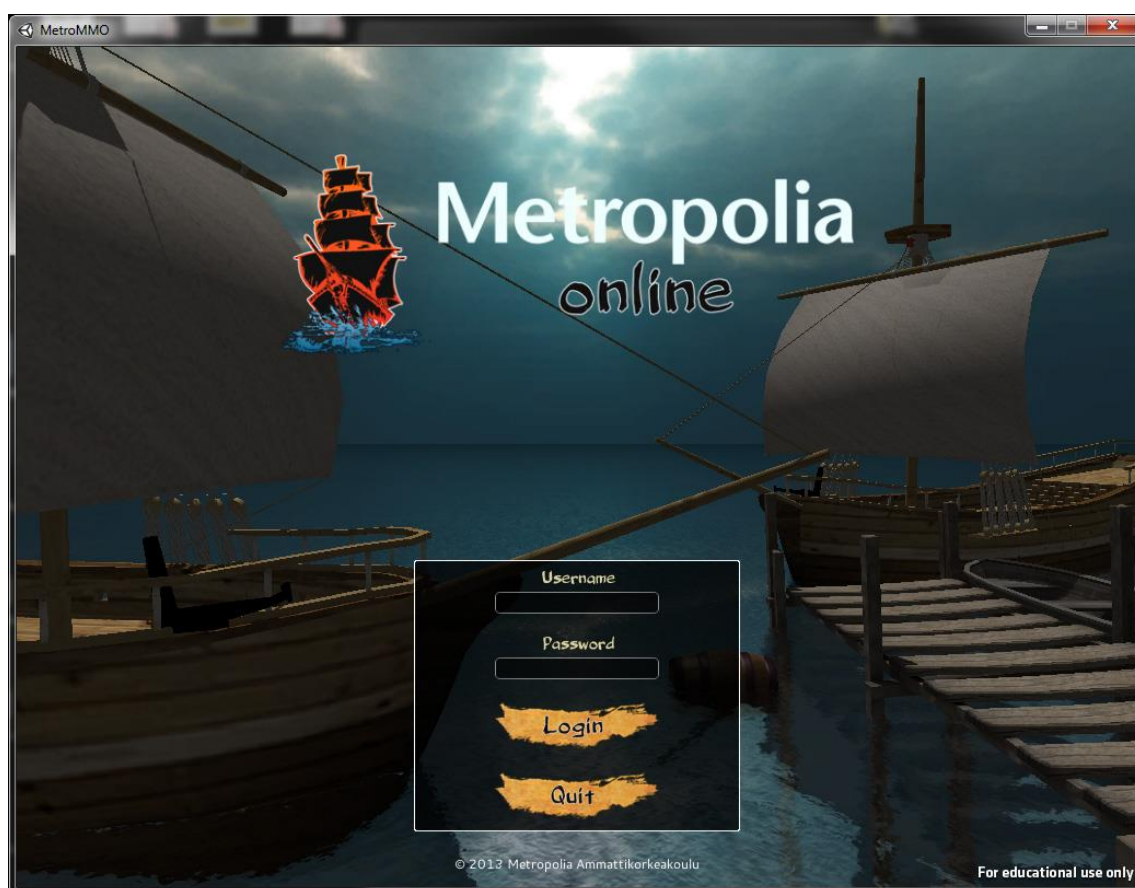


Figure 24: Metropolia Online login screen

The original design started so that players could model, texture, rig and animate their own characters to the game. This way students could create their own avatar they can use in the game. At the same time students would learn the process that is required to

(106)

get a working character to the game. From game point of view we could get a lot of characters to the game rather fast. After logging in, a character creation screen is displayed to the player where characters can be created, selected and removed as shown in Figure 25. Again the functionality was planned to be very simple because students can later improve it.



Figure 25: Character management within Metropolia Online

After character creation the player can join the game and is sent to the Start Village. This Start Village is a common gathering place where all new characters will appear. From here exploration to other places starts. If a character enters the World Map scene, the character model is changed to a boat thus being able to sail to other scenes.

Scene Changer class was created to enable players to go from one scene to another. This means that an invisible object is put on the scene and when collision is detected between character and the object, the current scene is closed and a new scene is loaded. Character's new spawn point is set in the previous scene when collision happens. The name of the new scene and location object are variables in the invisible collision object. This way the same objects can be re-used as often as wanted in a easy

(106)

way. Changing the scene is dynamic meaning chat history will not be erased and all characters already in the new scene are naturally visible to the new player.

One important and common feature is in-game chat. Because the game is fully a multi-player game, some kind of communication system is required and chat is naturally the best and easiest solution. Chat can be seen in bottom left corner of Figure 26.



Figure 26: Metropolia Online with chat available

Quitting the game would store location of the player to the database so the player can continue from the previous location next time he/she logs in. After all the game is not very feature rich but the work under the hood is done and more development can be done with students.

6.4.1 Game world design

Games created in Unity are constructed of scenes. One scene usually contains models of one part of the game and also scripts that create the logic and functionality. Each scene can be seen as one level of a game. One scene is usually a main menu while another one can be whatever the game is going to need. This scene structure had to be

(106)

taken into account when designing the structure of Metropolia Online. In the start Metropolia Online has a scene for login purposes and after that the player sees a scene that is for character creation.

Although we have a version control system set up for the game it does not solve all problems. Version control does not solve a situation where one student is editing the same scene with another student at the same time. This is due to a binary format of the scene which version control cannot handle. This means that every student has to have their own scene they can edit and then submit to the repository.

The game world was designed to have one scene called Start Village. This is the place where all new players will be spawned. From there players can travel to the pier where a trigger is first set to change the scene. The player gets into a scene called World Map. The idea of World Map is taken from Sid Mayer's pirates game seen in Figure 27 where the player travels by boat between different places.



Figure 27: Sid Mayer's Pirates [68]

The same idea is utilized in Metropolia Online. Going to the pier changes player model to the ship and takes it to the ocean. The player can now freely travel to some other scenes made by students. Other scenes have some kind of indicator to tell where the player should go as seen in Figure 28. Near the models there are again invisible colli-

(106)

sion detectors that will close the world map scene and open another one. These scenes are independent and created by students.



Figure 28: Exploring World Map by ship in Metropolia Online

This design works well because students can work on their own scene and then make a commit to the version control system without interrupting other students work. The only time when students need to modify the same scene is when they apply their model and scene changer to the world map but that is done only once so editing the world map by everyone is not constant. Figure 29 shows a simplified structure of scenes in Metropolia Online. Arrows describe how the player can travel from one scene to another.

(106)

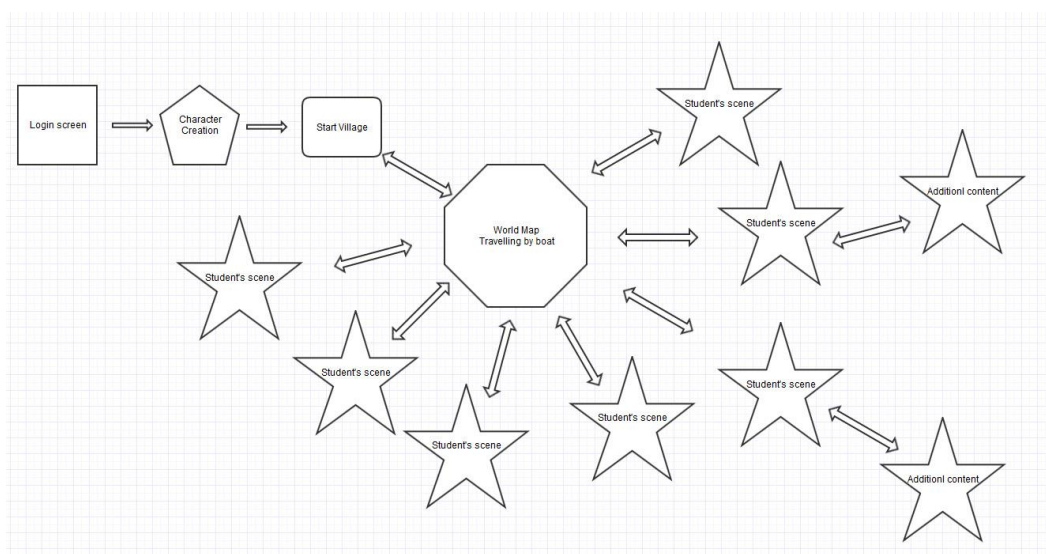


Figure 29: Scene structure of Metropolia Online

With this kind of scene structure total freedom is given to students to create any kind of world. The only thing that is required is to create some kind of pier that tells player this place is going to lead back to the world map. Working on one's own scene could later be a part of a portfolio and a student can experiment with any model and programming techniques he/she wants.

6.4.2 Village creation

One requirement for passing the course was to create one's own island and village for the game. On the first part of the course everyone practiced their skills and learned to use Unity as a game development tool. The second part of the course was to implement the local village with a character to the MMO game. This way the game will expand rapidly during every course by several villages and possibly several new characters.

Putting a village to the MMO game is quite easy. A locally made village can be copied to the MMO folder structure and then two prefabs are required to add to the scene. Dragging GameInitializer and ChatWindow prefabs to the scene makes network features available. Easy usability is the reason why uLink was chosen for the network platform for the Metropolia Online. GameInitializer makes all objects aware of each other

(106)

thus giving the MMO feeling. ChatWindow prefab naturally adds dynamic chat to the scene.

After these two prefabs are in scene connection to the World Map a scene is needed. This is done by adding a trigger to the pier that is going to change the scene to the World map. The trigger has two variables. One is the name of the scene and the second is the name of the Spawn Point seen in Figure 30. The character will start to play from new scene from the location of this Spawn Point.

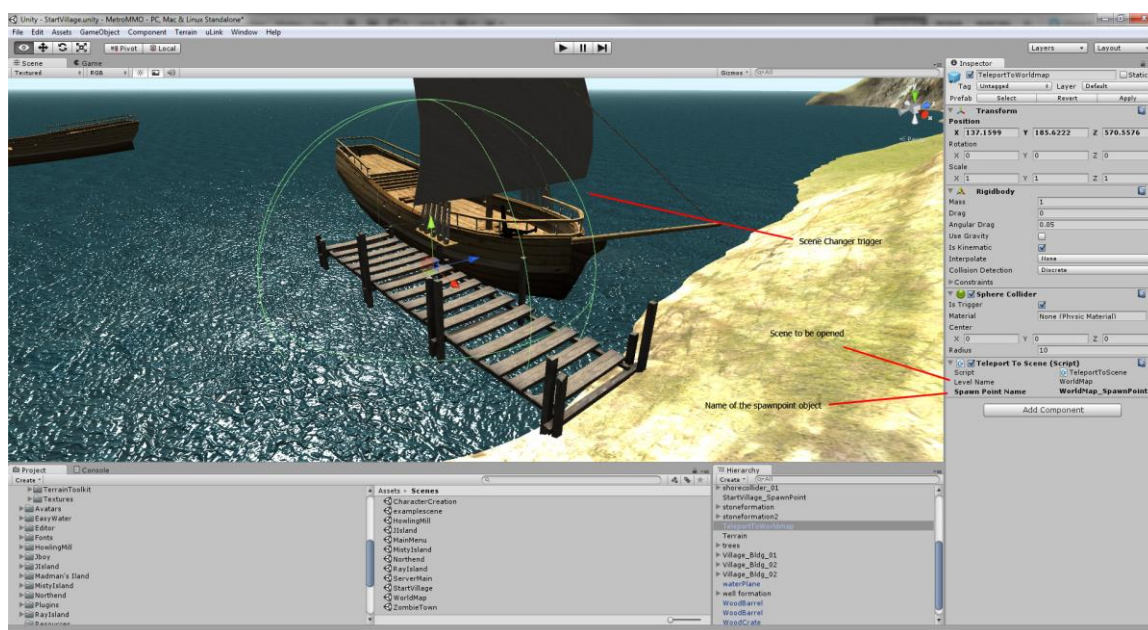


Figure 30: Properties of scene changer

Scene changing has to work both ways. The player has to have the ability to come and go between world map scene and other scene so a trigger has to be added to the world map scene, too. Spawn point must also be added to the village.

6.4.3 Character Creation

Creating a character to the Metropolia Online is a little more complex. Just modeling, UV mapping and texturing the character is a task that does require some experience in 3D graphics and that is out of the scope of this study. The character model needs to be

(106)

rigged for animation purposes. For this phase there are three working methods to do the rigging.

The time consuming and difficult method is to rig the character inside 3D program and then animate the character by hand using preferred 3D program. This is too difficult for most of the students and the purpose of the course is not about rigging this method is not recommended.

Second method is to use Mixamo (<http://www.mixamo.com/>) autorigger that offers automatic, easy to use rigging tool. With Mixamo students can take their 3D character and use automatic rigger to put bones on their character. The process is mostly automated thus saving time and making the process easier. Mixamo runs on a browser and after rigging, the user can download or buy motion capture files to animate the character. Mixamo also offers some free scripts to use in Unity. The rigging and animation process is simple and scripts work basically by dragging and dropping inside Unity.

Mecanim

(<http://docs.unity3d.com/Documentation/Manual/MecanimAnimationSystem.html>) is rather a major addition to Unity and it is the third way to do rigging. Mecanim is a recently added feature to Unity that brings rigging and animation flow tools easily available. Rigging using Mecanim is a little more difficult and a longer process than Mixamo but gives better results when mixing animations inside the game. Mecanim supports animation file recycling from one character model to another. This way all animation clips working on one character can be applied to other characters made by students. Metropolia Online has support for both ways thus giving students the possibility to choose either option. [69]

Single Character in Metropolia Online has two versions. The first one is the actual playable character that is controlled by player. Second character is identical with different scripts applied and acts as a proxy model. This model is displayed for other players through network connection. It can be said that other players just see 'a ghost' of the original character.

(106)

When applying right scripts to the character and adding them to correct folder, the character becomes available in the character creation scene where characters are made. It takes about 30-60 minutes to rig a character, apply scripts to it and make it work on Metropolia Online by using Mixamo rigging system. It was not compulsory to create a character during the course but it was highly recommended.

During the course additional features were gathered and some students were interested in developing them. Flying devices controlled by character was made as well as theme music to the main menu and wandering monsters to the start village. Other special features were also added by students to their own villages. This was a good way to get some lively aspects to the game. Creating some special feature to the game was also voluntary but of course it also gives the students a better grade.

7 Projects results and future development

This chapter provides a brief status of Metropolia Online and also future development ideas are listed. TODO environment has received some interest and there are also plans for future development. A brief report of the feedback of the first course can be found at the end of the chapter.

7.1 Current state of the Metropolia Online and Future Development

Metropolia Online server has been running several months without interruptions. During the course students created 13 different villages and characters to the game. Currently there are not many issues with the game that would require attention thus new features and content can be created to the game without any worries. Version control system is working and connection to TODO system is also working properly when creating a new version.

The game can be accessed through <http://todo.metropolia.fi> by registering and then downloading a zip file from Download Section. Unzipping and running the exe file runs the game. After logging in the player needs to choose a character to the game. After

(106)

giving a name to the character and choosing a suitable model from dropdown, the player can join the game and is spawned to the start village.

Another way to access the game and its development side is to see instructions from the TODO learning environment and installing version control system and Unity and then downloading the latest version from the repository. This way anyone can access to the source code, models and textures of the game on client and server side. Currently the revision number is over 150 which means students have done lots of updates during the course.

During autumn 2013 a new Unity course will start taking along other Universities of Applied Sciences along to the game development. These schools are part of the Game Cluster Project and their students have the possibility to take part in the development. This way we can test how TODO environment and Metropolia Online work within demanding situations. It is possible that adding more and more characters, villages and additional features will increase the game size so much that it is not sensible to use the same version every year. After the first course the repository is around 0,8 GB and downloadable client game is around 150MB. All content added to the game will increase these and in the future, additional repositories may be needed for simpler usability. That is why students should keep their models simple, textures optimized and other useless resource out from the repository. The teacher is of course responsible for guiding the students.

In the future two possible big modifications could be made to Metropolia Online. Version control could be moved to Bitbucket providing a more modern system for version controlling. Also an integrated wiki could be available for development purposes. This would make students familiar with the tool that is mostly used in the industry. It seems student licenses are free on Bitbucket giving unlimited storage capacity and good tools for managing source code and game versions. [55]

Bitbucket (<https://bitbucket.org/>) is a service that gives possibility to save projects in the cloud. While the current version control system uses SVN, Bitbucket uses Git for version control. Benefits for this change for students would be that Git is more used in the industry. This way students can get valuable experience of GIT during studies. Manag-

(106)

ing repository and workflows should stay the same when SourceTree software is installed on computer. SourceTree (<http://www.sourcetreeapp.com/>) is a free Git client for Windows and Mac computers. Additionally, Bitbucket offers some useful social applications such as wiki and chat available alongside the version control system. This way students can store valuable information about development process.[55, 70]

Another modification for Metropolia Online would be moving to Photon Cloud. This way a server would always be available in the cloud and students would not need to worry about server side code and all logic could be made on the client side. Also maintenance would be done by the cloud so Helpdesk at school does not need to worry about the servers. Photon Cloud has a similar API as Unity's own network solution. If database connection is needed a separate server could be created for that. The system would look simpler for students and development could be faster because not so much work needs to be done on studying the all code. Photon Cloud and uLink has quite a similar API thus making the change would not be an overwhelming task. More information about Photon Cloud can be found on <https://cloud.exitgames.com/> [71]

7.2 TODO environment improvements

Recently it has been realized that Finland could have lot to achieve by exporting education knowledge. Rovio's Chief Marketing Officer Peter Vesterbacka has stated that Finland should boldly take game education to the international markets. TODO environment could have the basis for bringing Finnish game education available to everyone. [56]

Minor updates to the TODO system will be made. Adding a file to the task comment is an addition that gives students to show screenshots of their game. Task list might collapse and displaying only undone tasks is a feature that should be added. Also help button to each task would be helpful. If a student clicks help button an email is automatically sent to teacher and information is added to backend side. This way when a student has a problem he/she can easily ask assistance and teacher immediately knows what the problem is and who is asking for help. This help to gather more information about learning environment and improves quality. Lot of usable features can be

(106)

added to the TODO system. It is just a matter of time and effort what features are going to be under development.

By proper funding and talented people a game like education system could be created for everybody on the web. Because game development is not a single skill, but rather a sum of lots of different talents such as modeling, graphics and programming as well as business skills. An education system that providing courses from different fields of game development could be created. These courses could be designed like courses in TODO system. Small tasks are given to students and by completing these tasks, the environment gives virtual experience points to the students. Getting these experience points by studying various levels, achievements or skills can be obtained. This way a student could, for example, be a level 2 3D modeler, level 4 graphic designer or a level 6 programmer with a speciality in artificial intelligence. When this is added to a MMO game development environment, a new educational paradigm could be achieved. This kind of learning environment could also benefit companies when finding new employees. They can see what kind of skills a student has through the learning environment.

This kind of system could have business potential not only in Finland but also worldwide. If tasks and courses in the system would be approved by famous game companies, the educational system could have a better reputation. Users could be charged by a monthly fee or by taking courses.

7.3 Feedback from the course

Around 65 students enrolled for the course and from those around 30 passed the first part. Many students hang on the course just for a couple days or a few weeks and then quit. This was the probably because combining 3D modeling and programming is a mix that not everyone can handle. That is why many did not have time to start learning so many new things. The first part of the course was passed by around 30 and from those, 18 students started the second part. Finally 13 passed the second part completing the whole course.

(106)

during the course there were small problems with some tasks, MMO game and TODO system but overall everything went much better than was expected. All in all there were 149 individual tasks. Numerous comments, hints and link were given by students commenting various tasks. Feedback was given by over 50 students through TODO environment. The course was successful and TODO environment is reliable and works well as a virtual learning environment.

An extensive custom feedback form was created to cover the whole course and teaching methods. The course was almost purely virtual and only TODO environment was used for teaching and communication.

Students gave feedback during the course and in the end. By gathering all information a general big picture of how the course could be improved can be formed. During the course mostly positive feedback was given:

- I personally like this kind of learning, small steps with good instructions.
- Hope it will go like this all the way to the end.
- It's amazing how much can be accomplished in a very short time.
- I'm looking forward learning more about this tool.
- Well, I think more courses should be done like this.
- Seeing that green DONE proves I've actually accomplished something in my life.
- It's so much more motivating when you can actually see your progress here.
- Haha, this system is awesome! I can even hear the "LEVEL UP!" sound every time I get the task done. :o

By gathering all feedback positive and negative things came up.

(106)

Positive

- As learning environment TODO system is great. Completing bigger tasks by doing small baby steps is a good idea.
- Many students would like to see more courses done this way. Self study with a help of learning material was a good choice for this kind of a course.
- Everyone could have their own pace when learning how to use Unity
- By doing tasks you know what to do next and you could easily follow others progression by seeing Group Pressure numbers. There were no “What should I do next?” moments
- The % number drove some students forward
- Getting village to the game was not hard
- It was great to see your own character in a game.
- MMO development was a great addition. Good that Metropolia is reacting to the global trend.
- Students enjoyed the course

Feedback was mostly positive and students generally had fun with the learning environment. Many said that self studying is a good way because students have different needs and this way they can concentrate on the things they feel is best. Tasks list gave a good flow for the studies giving everybody something to do all the time. Some students said that they followed the group pressure percentage and tried to stay ahead of the rest of the group. at the beginning of the course some students were skeptical whether they would be able to create all things required on the course. Maybe the subject was not that hard or instructions were good but according to the feedback creating a village to the MMO game was not that hard. Generally students were happy about how the course went.

Negative/Improvements

- Too much 3D modeling

(106)

- More programming
- Better instructions
- Better instructions to MMO
- Server problems
- MMO development
- Sometimes there were confusions with MMO

Most of the 3D modeling was done at the beginning of the course. This way students provide sensible content to their game and learn to import 3D models to Unity. Some students do not have experience in 3D graphics so providing 3D models for them could ease the course and make it more enjoyable. This way students who are more interested in programming and design can more or less skip 3D modeling. Programming tasks are of course all the time available, they can be done just by using simple box objects. Students followed task the list quite slavishly. No major skipping occurred which implies some programming tasks could be put in the beginning of the list. This gives programmers a more interesting beginning to the course.

Some student said that instructions for the tasks were good but some did not like them. It can be said that there is always room for improvement in instructions and guidance. There were some server problems due to some bugs in the code. This is an important thing in the future. If server is students feel the system is unreliable. Currently the server is running better and in the future things should work without crashing. There were also opinions that MMO game is not useful but there were also students who thought that MMO game development is taught at school. In future more MMO like features should be taught for students to help them better understand how the whole system works. Overall this is a learning process for everybody and the next course will hopefully be better.

8 Conclusions

People play more games than ever. Especially online gaming has become popular giving people the opportunity to contribute to the same game together. It is possible this enthusiasm to gaming could help us to solve real world problems in the future.

Finnish game industry has strong roots in the history and currently it has a good reputation. Developers and investors also pay attention to Finland. Increased popularity in game development means that better education is needed to provide workers for the industry. Finnish education system is probably going to see some changes in the future and creative teaching methods should be approached.

Game development is not one skill but a combination of a big skillset. Let it be designing, 3D modeling, graphic design, programming or business sense some skill is needed to work on a game company. Usually different teams of specialists work on specific task to complete game features. Universities can provide study programs for various field of game development.

The project carried out in this thesis is a virtual learning environment that relies on a task list. By completing small tasks, students progress on a course and they can monitor others progression at the same time. The teacher's role is to give support and monitor activity on the learning environment and courses can be held with less effort. Compared to traditional methods there could be a large group of students on a course that could not be easily taught by using traditional contact teaching. The new learning environment uses an experimental feature that tries to force students to make tasks using group pressure.

Additionally, a Massive Multiplayer Online game was developed for the project. By developing features to the game, students can test and study programming, 3D graphics and art design. The game is based on a server client approach with the use of database cluster that stores the game data. Commands given by players in client program are sent to the server for process. Updates to the game are made by students and version control system has been set up for sharing the latest revision of the game.

(106)

The project and initial course was a success and will be used again during game development courses in the future. The idea is that students from other schools that are participating the Game Cluster project can also take part in the MMO development.

Student feedback has been positive after the first course and future development is planned to make the project to continue in coming years. This project is also a showcase to prove that working on an MMO game can be done with a small team and developed by students in a school environment. It also gives the school the possibility to teach modern MMO development techniques and offers students a chance to experiment new technology they invent.

Sources

- 1 Presentation by Jane McGonigal in Los Angeles, Siggraph 2012
- 2 Video Presentation by Jane McGonigal
http://www.ted.com/talks/jane_mcgonigal_gaming_can_make_a_better_world.html
- 3 Elementary study plan in Finland
http://www.minedu.fi/export/sites/default/OPM/Koulutus/koulutuspolitiikka/vireilla_koulutus/perusopetus/liitteet/asetusehdotus_1_2.pdf
Accessed 18.7.2013
- 4 First Finnish games
<http://dome.fi/pelit/ajankohtaista/onko-tama-ensimmainen-suomalainen-peli-ikina>
Accessed 18.7.2013
- 5 Sanxion
<http://www.tdubel.com/emulaattorit/sc64pelit2.html#sanxion>
Accessed 18.7.2013
- 6 Finnish game industry
<http://www.neogames.fi/wp-content/uploads/2013/05/Finnish-Games-Industry-2013.pdf>
Accessed 18.7.2013
- 7 TEKES
<http://www.tekes.fi>
Accessed 18.7.2013
- 8 TEKES influence
http://www.tekes.fi/u/vaikuttavuuden_taskuesite.pdf
Accessed 18.7.2013
- 9 Presentation KooPee Hiltunen, Neogames, Espoo
20.5.2013

(106)

- 10 Invest In Finland
<http://www.investinfinland.fi/main.php>
Accessed 18.7.2013
- 11 Neogames
<http://www.neogames.fi>
Accessed 18.7.2013
- 12 IGDA
[http://igda.fi/?paHYPERLINK "http://igda.fi/?page_id=4"ge_id=4](http://igda.fi/?paHYPERLINK%20\)
Accessed 18.7.2013
- 13 Manse Project in Tampere
<http://www.slideshare.net/slatva/the-growth-of-tampere-game-industry-19068918>
Accessed 18.7.2013
- 14 Suomen Pelinkehittäjät Ry
<http://www.pelinkehittajat.fi>
Accessed 18.7.2013
- 15 Game know-how attracts international publisher to Finland
<http://www.goodnewsfinland.com/archive/news/game-know-how-attracts-international-publisher-to-finland/>
Accessed 18.7.2013
- 16 Remedy Open to Doing Another iOS Game
<http://toucharcade.com/2011/12/20/remedy-open-to-doing-another-ios-game/>
Accessed 18.7.2013
- 17 Aalto University New Media
http://www.aalto.fi/fi/studies/education/programme/new_media_game_design_production_master/
Accessed 18.7.2013
- 18 Kajaani University of Applied Sciences
<http://www.kamk.fi/fi/Hakijalle/AMK-tutkintonuoret/Insinööri/Tietotekniikka/Rakenne>

(106)

Accessed 18.7.2013

- 19 Kouvola Region Vocational College
<http://www.ksao.fi/koulutustarjonta/aikuiskoulutustarjonta.html?koulutus=192>

Accessed 18.7.2013

- 20 Kouvola Region Vocational College brochure
https://dl.dropboxusercontent.com/u/14560280/Pelisuunnittelija_2013.pdf

Accessed 18.7.2013

- 21 North Karelia Municipal Education and Training Consortium
<http://www.pkky.fi/Resource.phx/pkky/amo/outokumpu/english/generalinfo.htx>

Accessed 18.7.2013

- 22 Game Education in North Kaerila
http://www.carelian.fi/games/game_education_in_north_karelia

Accessed 18.7.2013

- 23 Sataedu, Audiovisual communications basic examination
http://sataedu.fi/fi/koulutukset/perustutkinnot/audiovisuaalisen_viestinnan_perustutkinto_pk_0

Accessed 18.7.2013

- 24 Tampere University of Applied Sciences, Game production
[http://www.tamk.fi/cms/tamk.nsf/\\$all/3D56EC79722CFA66C2257619004F9A65](http://www.tamk.fi/cms/tamk.nsf/$all/3D56EC79722CFA66C2257619004F9A65)

Accessed 18.7.2013

- 25 Turku University of Applied Sciences, Information Technology
<http://www.turkuamk.fi/public/default.aspx?contentid=141675>

Accessed 18.7.2013

- 26 Kymenlaakso University of Applied sciences
<http://www.kyamk.fi/Hakijalle/Nuorten%20haku/Tekniikan%20ja%20liikenteen%20ala/Tietotekniikka/>

Accessed 18.7.2013

- 27 Game Cluster Project Plan

(106)

- 28 Game Cluster Project Presentation, Cursor
- 29 The Finnish Games Industry, Samuli Syvähuoko, Presentation
19.3.2013
- 30 Game development Team
http://www.digital-church.com/html/week_53.html
Accessed 18.7.2013
- 31 Game programming know-how requirements
Heini Puuska, Miikka Mäki-Uuro, Presentation
- 32 Learning Theories Paradigms
<http://www.learning-theories.com/paradigms>
Accessed 18.7.2013
- 33 Oppimiskäsitykset
<http://oppimateriaalit.jamk.fi/oppimiskasitykset/oppimiskasitykset/behavioristinen-oppimiskasitys/>
Accessed 18.7.2013
- 34 Sibelius Academy
<http://www.siba.fi>
Accessed 18.7.2013
- 35 Trine 2 Main Theme, Video
http://www.youtube.com/watch?v=Znw_wi39nsQ
[Accessed 18.7.2013](#)
- 36 Angry Birds Main Theme
<http://www.youtube.com/watch?v=clnb7jilvQM&list=PLB0D0E095D79FCC47&index=1>
[Accessed 18.7.2013](#)
- 37 Angry Birds Theme by London Philharmonic Orchestra
<http://www.youtube.com/watch?v=Qbrgo3vPBdQ>
[Accessed 18.7.2013](#)

(106)

- 38 Interview of Ari Pulkkinen, Composer
<http://areena.yle.fi/tv/1381195>
Accessed 18.7.2013
- 39 Moodle Features
<http://docs.moodle.org/20/en/Features>
Accessed 18.7.2013
- 40 Digital Tutors
<http://www.digitaltutors.com>
Accessed 18.7.2013
- 41 Viope
<http://viope.com/>
Accessed 18.7.2013
- 42 Game institute
<http://www.gameinstitute.com/>
Accessed 18.7.2013
- 43 Virtual Learning Environments
Will Virtual learning environments improve teaching?
P.Dillenbourg, EUN Conference 2000
Accessed 18.7.2013
- 44 Ultima Online
http://www.uoguide.com/History_of_Ultima_Online
Accessed 18.7.2013
- 45 World Of Warcraft
http://www.wowpedia.org/Timeline_%28World_of_Warcraft%29
Accessed 18.7.2013
- 46 Unity
<http://www.unity3d.com>
Accessed 18.7.2013

(106)

- 47 UDK Game Engine
<http://www.unrealengine.com/udk/>
Accessed 18.7.2013
- 48 Unity PR
<http://unity3d.com/company/public-relations>
[Accessed 18.7.2013](#)
- 49 Mobile game developer survey leans heavy towards iOS, Unity
http://www.gamasutra.com/view/news/169846/Mobile_game_developer_survey_leans_heavily_toward_iOS_Unity.php
Accessed 18.7.2013
- 50 Photon
<http://www.exitgames.com/Photon/Architecture>
Accessed 18.7.2013
- 51 uLink
<http://www.muchdifferent.com/?page=game-unitypark-products-ulink/>
Accessed 18.7.2013
- 52 What is NoSQL?
<http://www.10gen.com/nosql>
Accessed 18.7.2013
- 53 Riak
<http://basho.com/riak/>
Accessed 18.7.2013
- 54 Trotoise SVN
<http://tortoisesvn.net/about.html>
Accessed 18.7.2013
- 55 Bitbucket
<https://bitbucket.org/>

(106)

Accessed 18.7.2013

- 56 Rovion Vesterbacka: Miksei Suomi valloita maailmaa koulutuksellaan?
<http://suomiareena.fi/?p=855>

Accessed 18.7.2013

- 57 Job Roles in Computer Games
http://www.creativeskillset.org/games/careers/article_2768_1.asp

Accessed 18.7.2013

- 58 Computer games developer
<https://nationalcareersservice.direct.gov.uk/advice/planning/jobprofiles/Pages/computergamesdeveloper.aspx>

Accessed 18.7.2013

- 59 Top skills to become a game developer: what South African developers say
<http://devmag.org.za/2011/01/29/top-skills-to-become-a-game-developer-what-south-african-developers-say/>

Accessed 6.8.2013

- 60 Game Development Process
<http://www.e-games.tech.purdue.edu/GameDevProcess.asp>

Accessed 6.8.2013

- 61 Discussion with Remedy's employee (Antti Veräjänkorva)
6.8.2013

- 62 Common Teaching Methods
<http://www.hec.gov.pk/InsideHEC/Divisions/LearningInnovation/Documents/Learning%20Portal/NAHE%20%28Presesation%29/COMMON%20TEACHING%20METHODS.pdf>

Accessed 6.8.2013

- 63 Why the "experienced" old dog game developer probably hasn't learned the tricks you really need.

(106)

<http://thebossmonster.blogspot.fi/2010/07/why-game-development-experience-is.html>

Accessed 6.8.2013

- 64 What do you think of moodle?
<https://tricider.com/en/brainstorming/AQOa>

Accessed 6.8.2013

- 65 How Long Should Your Web Page Be
http://webdesign.about.com/od/layout/qt/page_length.htm

Accessed 6.8.2013

- 66 What is Conformity
<http://www.simplypsychology.org/conformity.html>

Accessed 6.8.2013

- 67 What is NoSQL Database & Why NoSQL
<http://www.couchbase.com/why-nosql/nosql-database>

Accessed 7.8.2013

- 68 Picture of Sid Meier's Pirates!
<http://www.aeropause.com/wordpress/archives/images/2009/01/sidmeierspirates.jpg>

Accessed 7.8.2013

- 69 Mecanim Animation System
<http://docs.unity3d.com/Documentation/Manual/MecanimAnimationSystem.html>

Accessed 7.8.2013

- 70 SourceTree
<http://www.sourcetreeapp.com/>

Accessed 7.8.2013

- 71 Photon Cloud
<https://cloud.exitgames.com/>

Accessed 7.8.2013

(106)

- 72 Pula työntekijöistä painostaa pelialaa yrityskauppoihin
<http://www.digitoday.fi/viihde/2012/03/27/pula-tyontekijoista-painostaa-pelialaa-yrityskauppoihin/201226153/66>
Accessed 13.8.2013
- 73 Mediakeskus Lume
<http://lume.aalto.fi/fi/esittely/>
Accessed 14.8.2013
- 74 Metropolia, Elokuvan ja television koulutusohjelma
<http://www.metropolia.fi/?id=1090>
Accessed 14.8.2013
- 75 Team Fortress 2 lobby
<http://www.tf2lobby.com/>
Accessed 14.8.2013
- 76 Importance of a Business Plan
<http://www.daymondjohnacademy.com/importance-of-a-business-plan.php>
Accessed 14.8.2013