Empowering Students to Create Better Virtual Reality Applications: A Longitudinal Study of a VR Capstone Course

Tuukka M. TAKALA, Lauri MALMI, Roberto PUGLIESE, Tapio TAKALA
Department of Computer Science, Aalto University, Finland
e-mail: {tuukka.takala; lauri.malmi}@aalto.fi, pugliesr@gmail.com, tta@cs.hut.fi

Abstract. In this paper we present our experiences of teaching an annually organized virtual reality (VR) capstone course. We review three iterations of the course, during which a total of 45 students completed the course and 16 VR applications were implemented. Our comparative analysis describes the students' evaluation of the course, the applications created by them, and their development experiences. The results suggest that our gradual improvements on the course and the utilized software paid off, as the latest of the compared course iterations produced the best feedback and the highest quality VR applications. Our learning assessment analysis reveals that our course is effective in teaching VR application development and having students meet their personal learning goals. We also bring forward our RUIS toolkit that was used in the course with success, and present evidence on how better software toolkits can affect the development experience and allow students to create more impressive applications. Finally we share the lessons learned during five years of teaching the course, introducing several practical considerations for VR course organizers regarding pedagogics, software, and hardware.

Keywords: virtual reality, 3D user interface, project course, computer science education, software toolkit.

1. Introduction

In the past decades virtual reality (VR) applications have been developed predominantly by researchers and industry professionals, mainly because the required hardware has been costly. Recently the number of hobbyist VR developers has increased however, since affordable interaction devices such as Wiimote and Kinect have become available to consumers. The release of Oculus Rift head-mounted display (HMD) in 2013 has attracted substantial public attention; over 100,000 units of the developer kit have been sold (Rose, 2014). Facebook, Google, HTC, Samsung, and Sony are all investing in

VR and are launching consumer VR products. This heightened interest in VR also calls attention to virtual reality as a teaching subject, because there are more VR related job vacancies and more enthusiasm among students.

VR courses have been organized since 1990s in Aalto University, Finland. The teaching has been conducted with commitment and proper resources: between 1998 and 2005 a four-wall CAVE (Cruz-Neira *et al.*, 1993) was utilized, after which a two-wall virtual environment has been used. In 2011 the authors of this paper took over Aalto University's VR course, reorganized it, and have since been teaching it annually. Students of our course concentrate on learning VR concepts and application development by creating working VR applications in cross-disciplinary student groups, where the students can have diverse backgrounds.

In this paper we scrutinize our endeavors in enabling students to create impressive VR applications within the bounds of a VR course and in a manner that supports their learning. We present the structure of our course, its learning goals and outcomes, and the software toolkits that have been used. We also analyze how the utilized toolkits affected the student-created VR applications and the students' development experience. Moreover, we discuss how we have improved the course over the years, and examine how these improvements have affected the course results, student experience, and their feedback on the course.

2. Related Work

Sherman and Craig (2002) described VR as "a medium composed of interactive computer simulations that sense the participant's position and actions and replace or augment the feedback to one or more senses, giving the feeling of being mentally immersed or present in the simulation". Burdea and Coiffet (2003) defined the "three I's" of VR as immersion, interaction, and imagination, underlining the importance of human imagination in VR.

There exists a wealth of research about using VR applications to assist education and training (Lee and Wong, 2008; Seymour, 2008; Stone, 2001). Literature about teaching VR is rare however. We did an extensive search for papers about teaching VR, and found only a handful of them. Those papers are summarized below. Specifically, we did not find any longitudinal studies that would have presented an in-depth exploration of a VR course's evolution over several years.

2.1. *Existing Literature about Teaching VR Courses*

Bell (1996) was the first to report the design and implementation of a VR course. His course emphasized teaching VR concepts over implementing VR applications. This was influenced by the lack of VR hardware (expensive at the time), and hence the students practised VR development with standard desktop computers.

One of the first scholars who raised concern about issues in teaching VR was Burdea (2004), who claimed that VR education has been lacking at the university level, also stating that "VR cannot be taught adequately without a specialized laboratory". He pre-

sented a comprehensive list of VR courses organized around the world in 2004, estimating that only 3% of world's universities and colleges have VR courses. Burdea suggested that the scarce and fragmented offering of VR courses has resulted from shortage of faculty with first-hand knowledge in VR, high price of VR equipment, and unclear career path for people who are specialized in VR. Nevertheless, he asserted that the job market for VR specialists is growing, as more and more profitable VR applications are created in domains such as entertainment, military training, medicine, oil industry, and others.

Examining Burdea's list of VR courses, Herbelin and Cíger (2008) argued that "several courses labeled virtual reality systems or virtual worlds are in fact almost exclusively on computer graphics". One such course was presented by Zara (2006), and another by Cliburn (2010), who incorporated VR elements in the final assignment of his computer graphics course. Herbelin and Cíger also described their student workshop about VR that focused on immersion, skipping computer programming and technical development altogether. They noted that VR courses are often taught by computer science faculties, while there are several human factors involved in VR: physiological responses, usability, sense of presence, and other psychological aspects. They proposed that both the technical and human aspects should be treated with equal importance in VR education.

Häfner *et al.* (2013) presented their VR course that had an emphasis on soft skills that are required in interdisciplinary team-work within an industrial VR project. They showcased student projects and feedback from a span of three years. This was one of the few publications with a longitudinal view on a VR course that we came across, although it merely included quotes from students and numeral ratings of course elements without a deeper analysis or comparison between the years. Häfner *et al.* stated that student group formation and task specification design are key factors for a successful VR course project, and that students should have freedom to be creative. They also pointed out the importance of choosing a good VR toolkit for the students to use.

Miyata *et al.* (2010) introduced their VR project course that underlined group work and the design phase of VR applications. They described several applications that had been created in the course. They also presented students' ratings of their learning experience and the course (averaged over the years), and a more detailed evaluation from one course year, demonstrating that throughout the course the students gradually learned personal and teamwork skills.

Zimmerman and Eber (2001) reported one iteration of their interdisciplinary VR project course, where computer science and art students worked together in mixed groups. Their course consisted of lectures on VR technology and artistic topics, teaching laboratory sessions, and group presentations. In hindsight Zimmerman and Eber noted that the course could have lasted longer than its designated six weeks. Additionally, one VR workstation and one VR toolkit license was not enough for the 16 students, who had to take turns during the VR application development.

Stansfield (2005) presented a VR course that included capstone course elements. She argued that VR is a good subject for capstone courses, because of its broad, multidisciplinary nature that includes hardware, software, and human aspects.

There have also been attempts to teach VR via online courses: Pantelidis and Auld (2003) reviewed their experiences about teaching online VR courses and mentioned their

obvious shortcoming as "the lack of hands-on, guided experience with VR hardware and software". Cliburn *et al.* (2010) evaluated their online VR lesson units that could be integrated into existing computer graphics or human-computer interaction courses. Faculties could consider such integration if their prevailing computer science curriculum is overcrowded and a new course cannot be added. Cliburn *et al.* pointed out that this approach cannot provide the same depth as a full VR course.

VR and game development have much in common, and the two fields influence each other (Zyda, 2005). Thus it seems sensible to apply knowledge from game development education to VR development education. Our interest lies in capstone courses, which are common in computer science curricula. Durel (1993) defined them as "crowning cours-es" with the objective of integrating a body of relatively fragmented knowledge into a unified whole. In the field of computer science education there exists a number of papers describing capstone courses where the primary focus is game development (Linhoff and Settle, 2009; Parberry *et al.*, 2005; Ritzhaupt, 2009). Zagal and Sharp (2011) presented a survey of game development capstone courses, which was answered by 37 instructors. Their results contain statistics about capstone courses and the common features between them, acting as a baseline for other capstone course instructors.

In conclusion we summarize that the existing literature on VR courses emphasizes taking into account the multidisciplinary nature of VR, which involves a wide range of technical and human aspects. The importance of students' access to VR equipment and hands-on experience is also underlined. From the reported VR courses, most include a project assignment, where VR applications are developed by students who are working in groups. Only very few papers report on longitudinal development of VR education, as we will do in this paper.

2.2. *Challenges in Virtual Reality Application Development*

Virtual reality research has been conducted since late 1960s, when the first modern VR hardware was built (Sutherland, 1968). Even with decades of research and advances in the field, VR application development still involves many challenges (Green and Jacob, 1991; Steed, 2008; Wingrave and LaViola Jr, 2010).

In an earlier publication (Takala, 2014) we have the examined challenges of VR appli-cation development, which we summarize here: 1) VR development is very multifaceted and it incorporates programming, 3D mathematics, visual and audio assets, ergonomics, and human-computer interaction. 2) The technology is new and no standard hardware platforms for VR exist (Wingrave and LaViola Jr, 2010). 3) A VR application commonly embodies a 3D user interface (3DUI), where the user operates in a spatial 3D context (Bowman *et al.*, 2004). In contrast to developing 2D WIMP-interfaces (windows, icons, menus, pointer), 3DUIs are new to users and lack high-level "building blocks" and estab-lished interaction metaphors (Wingrave and LaViola Jr, 2010). Finally, 4) 3DUI testing cannot be automated (Wingrave and LaViola Jr, 2010) and needs to be carried out with the intended hardware setup. This differs drastically from developing interfaces for mouse and keyboard, where practically any computer can be used for testing.

VR application development is characterized by these challenges, which student and

hobbyist developers confront head-on. It follows that on a VR project course the instructor must steer the students through these development challenges by frequently observing their work and actively offering guidance when it is required.

3. Research Questions and Methodology

This paper presents an observational, longitudinal study of our VR course with a non-experimental research design; i.e. data regarding VR applications, development experience, and course feedback was gathered over several years without experimental manipulation. We conducted a detailed analysis on the collected data retrospectively, after multiple VR course iterations had already been organized. Between the courses we utilized the most pressing student feedback and our observations of their work to improve the course. Hence our study has aspects of design-based research (Juuti and Lavonen, 2006), but does not adhere to it in a stringent manner.

We investigate the following research questions in our study: *how does the utilized VR toolkit affect the created VR applications (RQ1) and the development experience (RQ2)? How did the changes in our VR course affect the student feedback (RQ3)? How do the students of our VR course evaluate their own learning (RQ4)?*

To answer RQ1 we measured various complexity indicators for the student-created VR applications, and created qualitative descriptions of the most representative applications. For exploring RQ2 we gathered data about our students' development experiences, using a 3DUI questionnaire that has been introduced in our earlier paper (Takala *et al.*, 2012). The 3DUI questionnaire included a set of Likert ratings and questions about the VR application development process and its difficulties. To address RQ3 we collected student feedback using Aalto University's standardized feedback form, which is the same for every course in the university.

Data concerning RQ1, RQ2, and RQ3 was collected from a total of 45 students who completed our VR course between 2011 and 2013. RQ4 was formulated later and studied by having 25 students of the 2015 course take part in an additional learning assessment. Details of the analyzed data items and the results of our study regarding RQ1, RQ2, RQ3, and RQ4 are presented in sections 6.1, 6.2, 6.3, and 6.4, respectively.

4. RUIS - Virtual Reality Software Toolkit

Generally speaking, VR toolkits provide reusable software components for creating VR applications and reducing the amount of low-level programming. Reality-based User Interface System (RUIS) is a VR toolkit developed by us in Aalto University. RUIS has been continuously developed during the years that we have organized our VR course. While our toolkit is not a learning platform in itself, we created RUIS with the core principle of easing VR application development for hobbyist developers.

RUIS fulfills a set of requirements that address VR development challenges, facilitates teaching VR development, and lowers the barrier of starting VR development. These requirements and the thinking behind them are presented in our prior paper (Taka-

la, 2014), which also includes a comparison of other freely available VR toolkits that are well suited for educational use.

4.1. *RUIS for Processing*

Processing is a free, easy-to-use computer graphics development environment, where applications can be run just by clicking a play button. In 2011 the students of our course used an early version of RUIS for Processing, where two hand-held Nintendo Wiimotes with attached color LEDs acted as six degrees of freedom (6DOF) controllers (Takala *et al.*, 2011). For the 2012 course we introduced a new version of RUIS for Processing, where the Wiimotes were substituted with PlayStation Move controllers that were tracked with a PlayStation Eye camera.

4.2. *RUIS for Unity*

Unity 3D is a widely used game development software suite, which according to a survey conducted by Zagal and Sharp (2011) was utilized in 55.6% of game development capstone courses as one of the toolkits.

 After the 2012 course it was apparent to us that Processing was not an ideal platform for VR development: it was slow at rendering 3D graphics and did not have a scene-graph, a thorough 3D programming library, an integrated physics engine, an ingrained audio engine, or a built-in 3D model importer. This led to students working at a very low abstraction level, which took away time and effort from creating more immersive virtual worlds and interfaces.

 We solved these issues by porting RUIS to Unity 3D, which provides the above-mentioned features that Processing is lacking, and contains a powerful, yet easy to use development environment. Unity Editor allows scene management where game objects, scripts, and other content can be created and modified using a graphical interface. Scenes can be run by pressing a play button, just like in Processing. From 2013 onwards the students of our course have used RUIS for Unity, which supported Kinect, PlayStation Move controllers, and various stereo 3D displays. A detailed introduction to RUIS for Unity is presented in our earlier paper (Takala, 2014).

 Students and anyone with less than $100,000 of annual gross revenue can use Unity Personal on their computer.

5. Virtual Reality Course

In the end of 2010 our faculty professor in Aalto University asked us to take over a previously organized virtual reality course and redesign it. The first author of this paper acted as a head teaching assistant throughout the three iterations of the course that are presented here. Before being assigned to take over the course, he had been researching

VR for four years. The head teaching assistant had previous teaching experience from two computer animation courses and one human-computer interaction course, but no formal pedagogical studies.

5.1. *Learning Goals and Pedagogical Framework*

Virtual reality is a rapidly developing area in computing based on theoretical knowledge of computer graphics, which is considered necessary prerequisite information for it. However, while we started to develop the new version of the course, we concluded that there were no clear theoretical models available how VR applications should be developed. Rather, the course should be practice oriented. We wanted our students to gain a hands-on learning experience of developing VR applications, and therefore leaned heavily on *active learning* methods (Bonwell and Eison, 1991) and a *student-centered* learning approach rather than emphasizing teacher-centered, lecture-based learning (King, 1993). Moreover, since capstone courses are very common in computer science education, we decided to build our course around a capstone project and thus incorporated some aspects of *project-based learning* (Thomas, 2000) into it.

We expected that many of our students would have an existing interest in computer games and VR technology, and we hoped to further encourage their enthusiasm and imagination. Therefore we gave students the freedom to plan and implement any VR application of their choosing, which also Miyata *et al.* (2010) had allowed to their students. This creative freedom also implied that the students would need to learn about both application related and technology related skills on their own. We decided to provide mainly such guidance that was of generic nature, rather than explicitly teaching application specific information.

The learning goals of the course included: 1) understanding basic VR concepts (e.g. immersion, 3D interaction, motion tracking, VR input & output devices) on such a level that allowed discussing and reporting on the application to be developed, 2) acquiring an ability to design a purposeful and immersive VR application, 3) obtaining the skills to realize such an application in a competent manner, and 4) improving group work skills so that the students would be better prepared to face similar team work situations in the future. The required level of immersive features and other expectations related to the VR application are described in Section 5.3.2.

Our purpose was to provide a holistic course that involves the whole VR application creation process from planning to implementation. We wanted students to experience different aspects of VR application development, highlighting those areas that are distinctive to VR applications – immersion, virtual world, and multi-sensory feedback. To achieve this, we gave our students access to VR input devices, motion trackers, and stereo 3D displays. Relying on that technology, the students had to design and implement a 3DUI for their VR application. They also faced tasks that are common to computer game development. This included creating and finding content (graphics and audio), utilizing a physics engine, and programming application logic as well as 3D graphics algorithms.

The course was mainly intended for students of computer science and related fields, but each year we advertised the course also to new media students in our university's School of Arts, who were allowed to participate. This was done to enable cross-disciplinary collaboration.

5.2. *Course Structure*

There were no strict prerequisites for attending the courses, but having a bachelor degree and a completed computer graphics course was strongly recommended. Passing the course was awarded with 4 ECTS (European Credit Transfer and Accumulation System) credits, which corresponds to roughly 120 hours of study. All iterations of the courses lasted for 19 weeks.

Each course followed the structure that is presented in Table 1: all lectures were given in the beginning of the first academic quarter, one lecture per week. The course featured 6 two-hour lectures covering different topics: introduction to the course and VR, input and output technologies, audio and acoustics, interaction and navigation, practice assignment, and project assignment. The latter two lectures also covered VR development with RUIS toolkit, and included example videos of VR applications that were later shared in the course homepage. Students were encouraged to acquire additional knowledge about the development environment (Unity or Processing, depending on the year) on their own through online tutorials.

In the final two lectures we asked the students to socialize, discuss their ideas, and start forming groups that would last for the remainder of the course. This way the students handled the group formation by themselves, and each year the assistants only needed to help a few students to find a group. The initial group sizes varied between two and four students, as we recommended a group size of three. The first task for the

Table 1

Structure of our VR course

Phase	Face-to-face teaching	Deliverables	Duration
1 Introduction	Lectures about VR: • Introduction to the course and VR • Input and output technologies • Audio and acoustics • Interaction and navigation	N/A	4 weeks
2 Group Formation	Lectures about VR development: • Practice assignment & RUIS • Project assignment & RUIS Two working sessions	List of group members Practice assignment	3 weeks
3 Project Assignment	Ten working sessions Demonstration event	Project plan VR application Live presentation	11 weeks
4 Assessment	N/A	Project report 3DUI questionnaire answers (Feedback form answers)	1 week

groups was a *practice assignment*, which introduced VR application development to the students.

The practice assignment was followed by a *project assignment*, which was the central part of the course: each student group wrote a project plan about a VR application that they wanted to create and then spent the majority of the course developing it. Near the end of the course we organized a *public demonstration event* where the groups presented their VR applications to an audience consisting of the students' friends and invited faculty members. The final project assignment task for each group was to write and submit a *project report*. After this the students answered a mandatory 3DUI questionnaire and could fill an optional course feedback form.

The VR applications were intended to work in Aalto University's virtual environment called Upponurkka (Lokki *et al.*, 2006), which has two stereo-walls. This environment was also equipped with inexpensive 6DOF controllers through all three years, and a Kinect in 2012 and 2013.

We organized weekly working sessions where the students were welcome – but not obligated – to participate for receiving feedback, testing, and further developing their applications. Working sessions were situated in a teaching laboratory that contained the VR hardware reserved for the students. These sessions were organized from the start of the practice assignment till the end of the course. Additional, daily working sessions were organized one week before the demonstration event for those students who wanted to polish their application. With the exception of a small number of emails exchanged between the students and the course staff, most of the communication occurred in the weekly working sessions.

The role of the professor in the VR course was to handle student enrollment, give half of the lectures in the beginning of the course, and register study credits in the end. The teaching assistants took care of everything else: giving the other lectures, organizing working sessions, assessing and giving feedback on students' deliverables, and keeping the course on track. In 2013 there were three teaching assistants while there were only two in 2011 and 2012.

5.3. *Course Deliverables and Assessment*

Instead of many small assignments, our course was designed around a big project assignment where each student group developed a VR application. This was preceded by one practice assignment.

5.3.1. *Practice Assignment*

After the student groups were formed, they started to work on the practice assignment, where students used our RUIS toolkit to create a basic VR application with object manipulation. The purpose of this was to familiarize the students with hardware and software aspects of VR development, and give hands-on VR experience before the groups decided on the topic of their project assignments. Three weeks of the semester and two working sessions were dedicated to the practice assignment.

In 2011 and 2012, the practice assignment requested students to create a photo viewer where the user could view and organize photos within a 3D space by moving and rotating them. With RUIS for Unity this task became too trivial to implement, and for the 2013 course we modified the practice assignment; each student group was asked to create a more general VR application, where 3D objects could be manipulated and scaled with a two-handed interface.

The practice assignment was not graded, but to be allowed to continue with the course each group had to pass it. The requirement for passing was that all the features (moving, rotating, and scaling of objects) had to be working properly.

5.3.2. Project Assignment

Development of an interactive VR application was the main focus of the course: in total 11 weeks out of 19 were reserved to the project assignment, which included writing a project plan, implementing the VR application, demonstrating it to a live audience, and writing a project report.

The students were free to come up with their own VR application idea. In order to spark the students' imagination, we also provided them with example ideas, such as an application for architecture, 3D modeling, virtual soundscapes, etc. Before starting with VR application implementation, each group wrote a project plan. We required the project plan to be 1-2 pages long outline containing a description of the application, its 3DUI, implementation, responsibilities of each student, and an explanation why VR technology was necessary in the implementation.

To ensure that the student groups would put effort into creating a VR application with proper immersion and interaction, we required that they included at least three of the following features in their application: *3D stereographics, head tracked view rendering, two-handed interaction / full-body interaction, multiuser application, haptics, infrared heat lamps, olfactory output, or tangible user interface*.

We offered our own RUIS toolkit for the students to use, but they were allowed to employ any VR toolkits of their choosing. Since RUIS includes the first three features from the above list, there was a substantial incentive for students to use it. We placed this incentive because we wanted to receive developer feedback for improving RUIS. Consequently, there was only one case where a student group resorted to something else than RUIS: in 2012 one group utilized Kinect and Unity instead of RUIS for Processing, which at the time supported only PlayStation Move controllers.

The students were encouraged to use free 3D models, textures, and sound assets from Internet, and even create their own content if they were so inclined. We let the students know that we expected their VR applications to utilize a fair amount of audio-visual assets.

The voluntary, weekly working sessions were organized so that each group could show their progress and develop their application further. During the working sessions we took the role of a coach, giving positive feedback in case of solid ideas, and voicing concerns about plans or implementations that seemed lacking. The teaching assistants would move between the student groups, trying to answer their questions and address their issues. Occasionally the assistants suggested improvements and explained vari-

ous interaction methods that could be implemented. If the students had a bug that they could not overcome, one of our assistants would attempt to fix it together with the students during the working sessions. Thus our focus was to maximize the students' time spent on creating results, by not leaving them to struggle alone with technical issues. Due to the informal and open nature of the working sessions, students were also exposed to each other's projects, sometimes trying out a fellow group's application and providing feedback.

Early on in the course we informed the students that near the end of the project assignment we would organize a public, two hour event where the students' VR applications would be demonstrated. Each group would present their application and willing members of the audience could use it. In game development education, live demonstration events similar to ours have been used to motivate the students (Linhoff and Settle, 2009) and to elicit competition (Ritzhaupt, 2009). Our aim was to inspire the students to create applications that they could proudly exhibit to others. This also acted as an implicit motivator for creating competition between the groups. The demonstration events were participated by the students, staff from our faculty, and any friends that the students had invited.

The final task for the groups was to write a 4-6 page project report, which was due one week after the demonstration event. We required that the project report described the purpose of the VR application, how well the initial goals were met, work allocation between students, and self-evaluation about what was achieved.

5.3.3. *Assessment*

At the end of the course, the students were given a grade on a discrete scale from 0-5, which is the standard grading scheme in Finnish universities. In this scheme 0 means failing the course, while 1 is the lowest and 5 is the highest passing grade.

In courses like ours it is not possible to explicitly quantify how well learning goals were met, as projects are not graded with exams. We did not assess learning outcomes on an individual level, but rather on a group level, evaluating our interaction with each group and their deliverables. Hence, all student group members received the same grade for completing the course. This decision was influenced by our intent of organizing a cross-disciplinary course with both computer science and art students. With the exception of general VR knowledge, it would have been difficult to try to choose shared learning outcomes to be measured between individual students who had such varied backgrounds and roles in the groups.

Grading for the groups was decided by a panel consisting of the course professor and the teaching assistants, who discussed each group's performance in the course. Our grades for the VR applications were based on what we witnessed in the demonstration event and in earlier interactions with the groups during the working sessions. The course learning goals were reflected in the grading, where we considered the purpose of the application, its immersive and interactive features, quality of implementation, student effort, and how well the application served its purpose.

Rather than grading different aspects of the VR application and calculating a weighted sum, our grading scheme was based on the notion that a VR application is more than

the sum of its parts. At the start of the grading process the panel recollected the course learning goals. Then the panel members independently determined overall grades for the VR applications holistically by taking into account the learning goals. The panel went through the groups one by one, each panel member stating their grade for the VR application and their reasoning. After going through all the groups and averaging the grades for each VR application, the panel reviewed the averages and rounded them up or down in a manner that ensured a consistent grading among the groups from the subjective perspective of the panel. In retrospect we can state that it would have been more informative for our students if we had scored the different aspects of the VR applications. On the other hand, such feedback was given verbally to the student groups during the working sessions.

The final course grade was based on both the grade given by the panel and the quality of the project report. The latter one could affect the final grade with one point in either direction except lowering grade 1 to zero or raising grade 5.

After deciding on students' course grades, we sent them a link to the mandatory 3DUI questionnaire, explaining that we had decided the grades and that they would not be disclosed until the questionnaire was answered by everyone. This was done in order to get as objective answers as possible, by letting the students know that their answers could not influence their grade, and by preventing each student's awareness about their grade to impact their answers.

6. Results

In total, 18 students enrolled the VR course in 2011, 13 students in 2012, and 21 students in 2013. The number of students who completed the whole course, were 14 (2011), 12 (2012), and 19 (2013). There were two teaching assistants in 2011 and 2012. We added a third one in 2013 to provide more support for the students in the working sessions. Students' background was similar each year: on average over the three years 76% of the students were studying computer science, 7% new media, and 17% other fields. As for pursued degrees, 13% were pursuing a Bachelor's degree, 79% a Master's degree, and 8% a Doctor's degree. The students had studied 4.4 years in the university on average when taking the course (median 4.5). Only 13% of all the students were female, which is close to the overall gender ratio of computer science students in our university.

In 2011 and 2012 the students formed five project assignment groups, and in 2013 there were six groups. Independently of us, the students assigned specific roles to each other within the groups, depending on their skills: some focused on programming, others on creating graphics or audio content. Group members were also fully responsible of task assignment and other organizational duties, and there were no cases where the teaching assistants would have had to intervene in internal organization of a group.

All student groups of each year passed the practice assignment. Each student group delivered the project plans and reports on time and as requested. We called for changes in a project plan only once, when a group wanted to use Arduino boards to give electric shocks to their application users.

6.1. *Course Outcomes: Student-Created VR Applications*

It was not a surprise that every year a vast majority of the VR applications were games; the only exceptions were a 3D interior design application in 2011, one virtual sound-scape application in 2012 and another one in 2013.

In 2013, we pushed the students to innovate by requiring that they use both Kinect and PlayStation Move in their VR application. Every application from 2013 used PlayStation Move controllers for more precise interactions (shooting, selecting & manipulating objects, and triggering events). Kinect's full-body tracking was used in four applications to control the player avatar, and in three of the applications certain poses were used to trigger commands. A video of students' VR applications from 2013 course is available online[1].

Because RUIS toolkit evolved over the years, RQ1 can be addressed by comparing the VR applications from each course iteration: Table 2 presents statistics about the VR application complexity for each compared year, including the number of sound effects, complex 3D models (other than basic primitive shapes), and complex features that were non-trivial to implement. We identified the following complex features within the applications: artificial intelligence, gesture recognition, interactive sound synthesis, multiple user interfaces, virtual drawing, and 3D menu. The complex features were additional in the sense that they were not required by the course, and instead stemmed from the students' own creativity.

According to our complexity metrics of Table 2, the VR applications from the 2013 course were considerably more complex in each category when compared to the earlier years. Significant differences between the courses were discovered with a Kruskall-Wallis test for the number of sound effects ($\chi^2(2) = 7.5$, p = 0.02) and number of complex 3D models ($\chi^2(2) = 6.4$, p = 0.04). Post-tests with Tukey-Kramer correction revealed that there were significantly more sound effects and complex 3D models in 2013 than in 2011.

In 2011 there were no multiuser applications while in 2013 all of the applications were intended for multiple users. We confirmed this to be a statistically significant difference using a Fisher's exact two-tail test (p = 0.003, 3x2 contingency table) followed by a pair-

Table 2

Statistics about the students' VR applications from each course year

	2011	2012	2013
Number of applications	5	5	6
Student group size (mean ± standard deviation)	2.8 ± 0.4	2.4 ± 1.1	3.2 ± 0.4
Sound effects	1.8 ± 2.2	2.0 ± 2.3	7.4 ± 3.8
Complex 3D models	1.8 ± 2.0	2.2 ± 1.5	12.5 ± 9.0
Complex features	0.6 ± 0.5	1.0 ± 0.7	1.5 ± 1.0
Multiuser support (% of applications)	0%	60%	100%
Music	0%	20%	67%

---

[1] Virtual reality games by students (2013): `http://youtu.be/Ptq9F1nnUaI`

wise comparison. The difference in the number of multiuser applications is at least partly explained by the fact that there were only two 6DOF controllers available in 2011, while in the later courses the students could employ up to four PlayStation Move controllers.

Fig. 1 features photographs of the most and the least successful VR application pairs from each course year, in terms of grade and the opinion of the teaching assistants. The
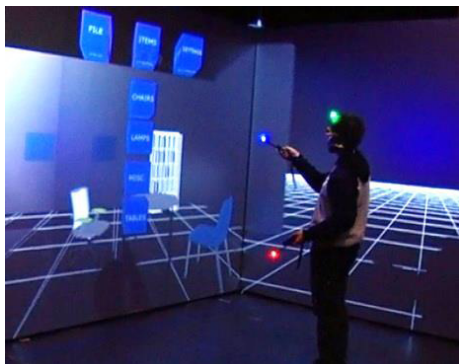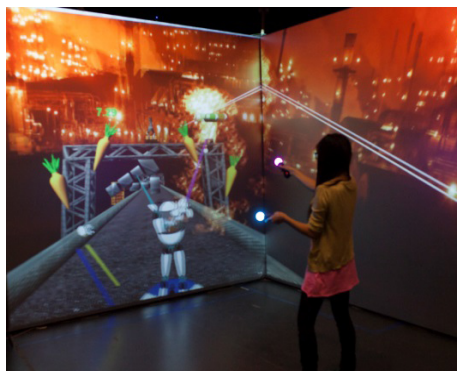

(a) Room Planner, 2011.   (b) Spiderman, 2011.

(c) Air Supremacy, 2012.   (d) Fishtank, 2012.

(e) Battletruck, 2013.   (f) Astro Surfer, 2013.

Fig. 1. Examples of student-created VR applications.

most successful applications (a, c, e) are on the left and the least successful (b, d, f) on the right side. We use this pairwise comparison to compactly illustrate the overall quality range of the students' deliverables for each year.

The same pair-wise comparison is present in Table 3, where we list each application's immersive and interactive features, student group size, and their grade. Each of the six applications' success, as categorized here, was reflected in the audience participation and interest during the public demonstration events.

In the following we describe these applications in more detail to highlight their differences both in terms of using VR technology, as well as their qualitative nature. Names of the applications have been changed to provide better anonymity for our students.

Room Planner application from the 2011 course (Fig. 1a) enabled the user to utilize a drop-down 3D menu to select pieces of furniture and then place them into a 3D scene. The application included a powerful 3DUI where one Wiimote was used for viewpoint control and the other was used for furniture manipulation. Almost every button on the two Wiimotes was mapped to a distinct action in the 3DUI.

Spiderman game from 2011 (Fig. 1b) was a simple game with three levels, each level consisting of a number of spheres arranged between a start and a finish zone. The objective of the game was to use hand-held grappling hooks to latch onto the spheres, swinging from start to finish like the comic book hero Spiderman through Manhattan.

Table 3

Comparing the most and the least successful application pairs from each year that the course was organized

Year	Application	Immersion features	Interaction features	Group size	Grade
2011	Room Planner	● stereo 3D ● head-tracking ● 5 complex 3D models ● 1 sound effect	● two-handed 3DUI ● 6DOF object manipulation ● 3D dropdown menu	2	5
	Spiderman	● stereo 3D ● head-tracking	● two-handed 3DUI	3	2
2012	Air Supremacy	● stereo 3D ● 3 complex 3D models ● 1 sound effect ● rumble effect	● two-handed 3DUI ● multiuser application ● 3DOF shooting	4	4
	Fishtank	● stereo 3D ● artificial intelligence	● two-handed 3DUI ● multiuser application	2	3
2013	Battletruck	● stereo 3D ● 22 complex 3D models ● 10 sound effects ● music ● artificial intelligence	● full-body interaction ● multiuser application ● 6DOF shooting ● driving interface	3	5
	Astro Surfer	● stereo 3D ● 14 complex 3D models ● 9 sound effects ● music	● full-body interaction ● 6DOF shooting ● multiuser application	4	3

It was not possible to fall and lose the game, and as such there was very little excitement or motivation. The graphics consisted only of graphical primitives and there was no audio. The students who created Spiderman admitted in their project report that they did not accomplish as much as they wanted, because they were busy with other work during the VR course.

In 2012 a group of students created a game called *Air Supremacy* (Fig. 1c), where one player used two PlayStation Move controllers to pilot a warplane through an obstacle course of hoops, while a second player acted as a gunner, shooting down enemy planes with a rifle peripheral that had a third controller attached to it. Both the obstacle course and the terrain 3D model were randomly generated upon game start. Air Supremacy was fun to play and the best looking application from all the applications created using RUIS for Processing in 2011 and 2012.

In *Fishtank* game from the 2012 course (Fig. 1d) two players competed to catch as many fishes as possible with virtual nets. Each player held two PlayStation Move controllers whose relative position oriented the net, with one controller representing the open end of the net and another representing the back end. Unfortunately this two-handed controlling scheme did very little to add to the immersion, and just one PlayStation Move would have been sufficient. In their project report students behind Fishtank acknowledged that the interaction was clumsy and did not feel natural.

Battletruck game from 2013 (Fig. 1e) was the most entertaining and the best looking application created in the first three years of our VR course's history. In this co-operative game one player acted as a driver of a truck and another player was a bazooka soldier standing on the truck's pallet. The player driving the truck had to locate and pick up explosives by navigating through a hilly countryside that contained roads, forests, military bases, and enemy tanks. The game's goal was to take the explosives to the enemy's main base and demolish it. The truck's only defense was its bazooka soldier passenger, Kinect-controlled by the second player. The game required skill and co-operation between the two players.

Created in the 2013 course, *Astro Surfer* was a game (Fig. 1f) where the player was represented with a Kinect controlled full-body robot avatar that automatically moved forward on a straight road that was filled with obstacles. The player used two PlayStation Move controllers to aim and shoot a gun and a flamethrower. There was no need to avoid the obstacles in the game, as the player could not be harmed or slowed down by them. The students reported that the gameplay of Astro Surfer did not have a clear goal aside from finishing the level as fast as possible by collecting power-ups. The game offered little motivation and started to feel repetitive quickly.

6.2. *3DUI Questionnaire Results*

Between 2011 and 2013 our 3DUI questionnaire was answered by all 45 students of the three VR courses of that period. Their answers can be used to investigate RQ2 by comparing how the students' VR development experience varied through the three course iterations, during which different RUIS versions were used.

6.2.1. *Student Satisfaction with the VR Application*

Each student answered whether they would present their VR application to a possible employer. In 2011 50% of the students said yes, in 2012 that percentage was 67%, and in 2013 it was 89%. A significant difference was found with Fisher's exact two-tail test (p = 0.04, 3x2 contingency table) when comparing these answers between the courses. Pairwise comparisons revealed that there were significantly more yes-answers in 2013 than in 2011. This indicates that students in 2013 valued their applications as more presentable than students in 2011.

6.2.2. *Development Difficulties among Students*

Our 3DUI questionnaire contained a list of statements that describe different development difficulties. In the past we have used the same statements to explore difficulties between experienced and inexperienced developers (Takala *et al.*, 2012), and difficulties between our students and other VR developers who had answered our 3DUI questionnaire (Takala, 2014).

The VR course students rated each statement on a seven point Likert scale (where 1 indicated strong disagreement and 7 indicated strong agreement), basing their ratings on the experience of developing their VR application during the course. Each statement was constructed in a way where a higher rating signifies that more difficulty was experienced.

The first 8 statements concern difficulties in the beginning of VR application development. In the context of our VR course, the term *3DUI toolkit* is interchangeable with *VR toolkit* in the below statements:

A1. Getting input device drivers to work was difficult.

A2. There were too many steps required between connecting the input device for the first time and successfully streaming data from the device into my application.

A3. Device input data was too low-level for quickly getting started with my 3DUI application.

A4. Lack of documentation or tutorials about the 3DUI toolkit made the development difficult.

A5. The 3DUI toolkit had a steep learning curve.

A6. The development was difficult because the 3DUI toolkit had a bad programming interface.

A7. Programming in general was difficult.

A8. Creating mathematical algorithms required by my application's 3DUI was difficult.

Significant differences between the three compared course iterations were found with a Kruskall-Wallis test in ratings of statement A5 ($\chi^2(2) = 6.5$, p = 0.04) and statement A6 ($\chi^2(2) = 13.8$, p = 0.001). Post-tests with Tukey-Kramer correction revealed that statement A5 was rated as significantly more severe in 2012 than in 2011, and A6 was rated as significantly more severe both in 2011 and 2012 when compared to 2013.

The questionnaire also had the following eight statements concerning difficulties that can occur later in the VR application development:

B1. Input device performance was poor.

B2. There were bugs in the 3DUI toolkit that I used for developing my 3DUI application, making the development difficult.

B3. Lack of proper 3DUI building blocks made it difficult to develop my 3DUI application.

B4. Each added 3D interaction feature increased application complexity, making the development difficult.

B5. Constant testing and re-implementation was required, making the development difficult.

B6. Testing of the application's 3DUI could not be carried out properly with just mouse and keyboard, making the development difficult.

B7. Teamwork was difficult.

B8. Legal status of using unofficial drivers and libraries for commercial purposes was unclear.

When comparing the ratings of statements B1-B8, a Kruskall-Wallis test found a significant difference between the three compared courses only with B3 ($\chi^2(2) = 8.1$, $p = 0.017$); a post-test with Tukey-Kramer correction showed that the lack of proper 3DUI building blocks was rated as more severe for the toolkits used in 2012 (mainly RUIS for Processing) when compared to 2013 where students used only RUIS for Unity. A summary of the statistically significant differences for all the statements can be found in Table 4, which indicates that the significant results are due to different VR toolkit versions.

Fig. 2 shows the Likert ratings of the development difficulty statements, revealing that statement B6 was consistently rated as the most severe issue over the years.

Table 4

Statistically significant differences in development difficulty statements between different VR courses

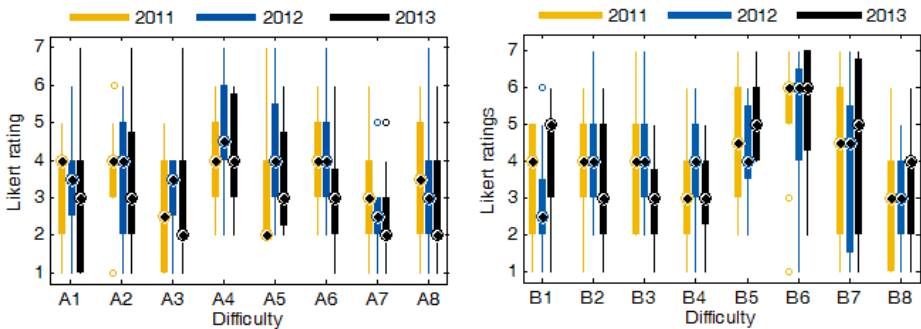| Difficulty	Less severe in	More severe in	$\chi^2(2)$	p
A5. The 3DUI toolkit had a steep learning curve	2011	2012	6.5	0.038
A6. [...] 3DUI toolkit had a bad programming interface	2013	2011, 2012	13.8	0.001
B3. Lack of proper 3DUI building blocks [...]	2013	2012	8.1	0.017

Fig. 2. Boxplots of severity ratings for early (Left) and later (Right) development difficulties.

6.3. *Course Feedback*

Aalto University's standard student feedback form included an overall rating to the course (5-point Likert scale from 'poor' to 'excellent') and difficulty rating (5-point Likert scale from 'very easy' to 'very difficult'). There were separate 4-point Likert scales with a 'Not applicable' option for rating the lectures and the project assignment (from 'very bad' to 'very good') and statements regarding teaching (from 'strongly disagree' to 'strongly agree'). The form also had open-ended questions asking about the good and bad sides of the course. Giving feedback was voluntary and done anonymously.

The feedback form was filled by five students in 2011, two in 2012, and 15 in 2013, shedding light on RQ3. The varied number of students giving feedback was affected by many factors that changed over the years: number of enrolled students, general enthusiasm, and possible confusion between the voluntary feedback form and the mandatory 3DUI questionnaire.

The students' overall rating for each course and their difficulty are presented in Fig. 3. The 2013 course had the highest mean (3.86 / 5) and median (4.0 / 5) overall rating from the three courses. This also meant that our 2013 course was rated among the top third of all (73) computer science courses organised in study year 2012–2013 at Aalto University. Considering the challenging nature of our course, this was a very good achievement from our perspective.

Fig. 4 shows students' Likert ratings for lectures and project assignments, as well as their agreement on a Likert scale about various statements regarding the course. Due to the low number of responses we could not find any statistically significant differences between the course instances.
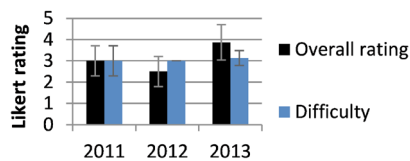

Fig. 3. Mean overall rating and difficulty for each year's course as rated by students' in the feedback form.
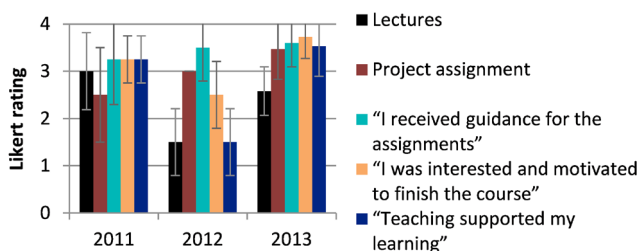

Fig. 4. Students' mean Likert ratings for lectures, project assignments, and three statements about the course.

We dissected the students' answers to open-ended questions from 2011 and 2013. Answers from 2012 were not available due to the fact that only two students gave feedback. We found four different comment categories from the answers: either positive or negative comments about the course or technology used within (VR toolkits and hardware). These comments are summarized in Table 5.

There were more positive individual comments than negative comments (10 versus 5) about the course in 2013. The opposite was true for the 2011 course (2 versus 4). The 2013 course received the best feedback in answers to open-ended questions, with comments like "*One of the best courses in Aalto University I have ever taken!*" and "*Liked it very much and even proved to be very useful in a job-interview. All-in-all a very interesting and engaging course!*" Particularly the hands-on teaching and technical support from the assistants received praise: "*The project was very fun and support from the assistants was exemplary*" and "*Help offered by the staff was superb*". In comparison, the most praising comments from 2011 were less enthusiastic: "*Good course. Interesting lectures and fun exercise.*" and "*Having a whole system like RUIS available is really cool.*" Feedback from three of the 2013 students specifically mentioned the teaching assistants and their help as a positive factor, while there were no such mentions in 2011.

Table 5

Summary of comment contents regarding the course and its technology in 2011 and 2013. Quotations are directly from student feedback; the other sentences are stylized by us

Year	Comment subject	Pos. / neg. comment	Contents of comments
2011	Course	+	+ "*interesting lectures and fun exercise*"
			+ good advice in the working sessions
		–	– project assignment should have partial milestones
			– the course felt unorganized
			– "*don't mix in students from arts-school*"
	Technology	+	+ RUIS toolkit
		–	– problems with 3D programming
2013	Course	+	+ "*the course was interesting and fun*"
			+ "*learned a lot of useful stuff*"
			+ help offered by the assistants
			+ working sessions were useful
			+ "*project was great*"
			+ free choice of project topic
			+ "*material for the course worked well*"
		–	– project assignment should have partial milestones
			– "*the lectures didn't teach much*"
			– too little teaching about Unity
			– the virtual environment was available only once a week
	Technology	+	+ students could borrow Kinect
			+ 6DOF controller simulation with mouse in RUIS
		–	– inability to test applications at home with intended hardware
			– inaccuracy of Kinect tracking
			– problems with Unity
			– "*RUIS was barely documented*"

6.4. *Learning Assessment*

We consider that from a teaching point of view it is not meaningful in this kind of multidisciplinary and practice-oriented course with group work to evaluate individual students' learning results using an exam or a pretest/posttest setting. Therefore the method of assessment used in the course focused on group level achievements and no other assessment was used.

To answer RQ4 and get insight into the individual level learning we decided to use a different type of approach. Because there had been no significant changes in the course since 2013, we asked 25 students of our latest 2015 course to self-evaluate their learning. The students filled a learning assessment form that contained Likert statements (Table 6) and open-ended questions.

6.4.1. *Likert Statements Regarding Learning*

Each statement was rated on a seven point Likert scale, where 1 indicated strong disagreement and 7 indicated strong agreement. The form also contained a control question "I learned about French cuisine"; 24 students rated this statement with 1 and one student with 2, which suggests that the students associated rating of 1 with no learning in the subject of the statement. Despite this we chose the 'neutral' rating of 4 from the middle of the scale as a more robust threshold to indicate whether learning occurred according to the students' opinion.

For each statement we used a Wilcoxon signed-rank test with Bonferroni correction to examine whether the median of ratings differed from 4. Results are displayed in Table 6: with each learning item L1-L7 the ratings' median was above 4 with statistical significance ($p < 0.05$) and a large effect size.

When tested whether any of the statement medians were above 5 with statistical significance, only statement L7 passed the test with Bonferroni correction ($p = 0.0007$, $z = 3.9$). The effect size was large ($r = 0.78$) also in this case.

Table 6

Likert statements, medians of their ratings, and statistics for Wilcoxon signed-rank test to determine whether medians of ratings are higher than 4. The two-tailed p-values are before Bonferroni correction

Statement	Median	p	z	Effect Size r
L1. I learned about basic VR concepts	6	0.00007	4.0	0.79
L2. I learned about VR application implementation	6	0.00013	3.8	0.77
L3. I learned about working with VR technology	6	0.00018	3.7	0.75
L4. I learned about content creation for VR	5	0.00561	2.8	0.55
L5. I learned about 3D user interfaces	5	0.00010	3.9	0.78
L6. I learned about Unity 3D	6	0.00249	3.0	0.60
L7. I am now better at contributing to VR application creation than before the course	7	0.00002	4.3	0.86
L8. I had VR knowledge before the course	3	0.02651	-2.2	-0.44

6.4.2. *Open-Ended Questions Regarding Learning*

We carried out a qualitative content analysis on the students' responses to the open-ended questions about their goal setting and learning. We highlighted sentence by sentence the concepts, goals or results that were mentioned and created initial categories for them. Thereafter the data was revisited several times and some categories were refined or merged to reduce their total number and remove overlap. Moreover, we defined upper level categories that were clearly related to one or more low-level categories and combined them into logical areas, such as 3DUIs, VR, or Devices. Separate categorizations were created for goals and learning results, and they allowed us to identify the main findings for the following two questions:

> **Question 1:** *In your own words, what were your own goals for the course? What did you want to learn?*

This question opened a rich picture of students' expectations on the course. By far the most common goals were related to learning about *3D user interfaces and input devices*. A clear majority of students mentioned either one or both of these motivations in their responses, quotations of which are presented in italics:

- *I wanted to get practical experience in using and programming for 3DUI and VR gear.*
- *I wanted to explore the possibilities and limits of 3D stereoscopy graphics within a virtual reality experience.*
- *I wanted an experience in using the new devices.*

The next major theme, mentioned by half of the students was *learning about Unity* as a development tool and environment.

- *I had only used Unity 3D a little in the past and felt really inclined to get better at using it.*

As an overall picture we found that students' motivation revolved largely around practical skills, learning to use the modern 3D input devices and the development environment. However, more generic goals concerning VR and designing VR applications were mentioned, as well.

- *I wanted to learn about VR technologies and how I can develop for it currently.*
- *I was hoping to learn some basic VR concepts and familiarize myself with challenges of using VR compared to more traditional output devices*

> **Question 2:** *Next, discuss how well your goals were met. What kind of theoretical things (concepts, principles) and practical things (tools, practices, software) did you learn?*

The main area of learning outcomes mentioned in the response was tools and technology. Learning to use Unity was mentioned by most students. Also the RUIS framework was commonly reported. Few students also reported learning C#.

- *Unity and C# were the most useful things I learned. I learned how scenes, objects, scripts etc. work in Unity. RUIS was useful, but it abstracted away all the techni-*

cal details: I can't use Move / Kinect / OR without it. I don't remember learning
any theory.*
- *I think, I met the goals really well and especially developing VR applications with
the Unity.*

Another common theme covered learning to use devices, implementing software based on them, and discovering their limitations.

- *Regarding the goals, most valuable was the early idea concepting and hands-on
approaches with the Move and Kinect, what kind of clicks and gestures worked
and didn't work.*
- *The most relevant learning from the course was to see and experience (from developer's point of view) the current state of the much talked about consumer VR.*

In general we can say that students' goals and their achieved learning outcomes matched well. Several students explicitly mentioned that their goals were met. There were very few signs of disappointment.

- *The goals were met well. We implemented some theoretical things, that we learned
during the course, but we also experimented with new ways and new possibilities
to control stuff in 3D space.*
- *I did end up learning something about Unity as well, though maybe not as much
as I had hoped.*

7. Discussion

Our course is relatively affordable to organize. All the required software can be used for free by the students: Unity, RUIS for Unity, and Blender 3D, which was used by some students to create 3D models. Kinect sensors cost $200 apiece, and PlayStation 3 with PlayStation Move controllers can be obtained for $300. The most costly item in our course was the Upponurkka virtual environment with two stereo-walls and a price tag of roughly $10,000. However, such a system is not necessary for organizing a VR course, as HMDs will suffice for the most purposes. For example an HTC Vive HMD and two 6DOF controllers can be bought for $800, which offers similar quality to older HMDs that have cost several thousands of dollars in the past. Thus a VR workstation consisting of a $1000 computer and an HTC Vive will cost only $1800.

In our course the project assignment requirements ensured that each year all the student-created VR applications fulfilled the three I's of VR (Burdea and Coiffet, 2003): immersion was achieved by employing stereo 3D, often paired with head tracked view rendering, and a 3DUI based on the use of motion controllers. All the applications were also interactive, and exhibited various degrees of imagination.

7.1. *Course Evolution over the Years*

Feedback from students revealed that lectures were not the strong suite of our course, while the project assignment was – especially in 2013. We addressed the complaints

about the lectures by reducing their number from six to five in 2013. Otherwise the course structure and teaching have remained the same till 2015. The most prominent course improvements between 2011 and 2013 have come in the form of updates to the RUIS toolkit and new hardware that has resulted in more impressive student applications.

Likert ratings of the development difficulty statements reveal that in 2012 students rated RUIS toolkit to have a significantly steeper learning curve when compared to students of 2011, i.e. in 2012 it was more difficult to learn. In 2012 students also rated RUIS to have a significantly worse programming interface than students in 2013. We believe that these differences are explained by the several new features that were added to RUIS for Processing before and during the 2012 course. These additions were described only with commented code examples, and otherwise our documentation was lacking, which seems to have affected the toolkit learning experience for the students. Many of the added RUIS for Processing features were already present in Unity 3D, which is very well documented and has a large user community. When we ported RUIS to Unity 3D for the 2013 course, we inadvertently reduced the documentation issues that had troubled the students in 2012.

Overall the 2013 course received the best feedback from students in terms of Likert ratings and answers to open-ended questions. We speculate that the following reasons contributed to this:

1) Unity 3D software suite was easier to use, had more expressive programming library, and had a simpler asset pipeline than the Processing environment of earlier years.

2) We included a third teaching assistant who had passed the course earlier as a student and was very experienced with Unity 3D. This allowed us to give more face-to-face technical support for the student groups during the working sessions.

3) Several students were aware of Oculus Rift and the hype around it in 2013, and the students seemed more enthusiastic about VR than before.

4) It is possible that the teaching assistants' accumulated experience of running the course for three years had a positive impact in the 2013 course and its student feedback. The head teaching assistant had taught in the two previous courses and another assistant in one previous VR course.

Our analysis of the development difficulty ratings between the three VR courses indicate that the students had the least amount of development issues in 2013; there were three difficulty statement comparisons where the 2013 course had significantly less difficulty than an earlier course. Each of those difficulties concerned the VR toolkit, which suggests that RUIS for Unity is more convenient for developing VR applications than RUIS for Processing. A big part of this convenience can be attributed to Unity 3D being a lot more suitable for complex 3D application development than Processing. Unity 3D is also able to produce much better graphics with faster frame rates. In 2013 students created applications with very impressive graphics when compared to the earlier years. In our opinion even the graphically weakest applications from 2013 looked better than the best looking applications from 2011 or 2012. We assume that this is one reason why significantly more students in 2013 expressed to

be willing to show their VR applications to a possible employer when compared to students from 2011.

It is interesting to contrast the student-created VR applications, which were the most important course outcomes. Looking at the photographs, descriptions, and complexity metrics of our students' VR applications, it is clear that in 2013 the VR applications were more complex and contained more interactive elements than in the previous years. The utilization of audio content is also notable: in 2013 every application included audio, while in the earlier years that was the case only for some applications. These observations demonstrate how better software tools can enable students to use more of their creative potential and build better applications.

Having only one computer equipped with the intended VR hardware was a problem at the working sessions, especially in 2011. The student groups had to wait for their turn to test their application properly. The situation improved with the new versions of RUIS in 2012 and 2013, where multiple groups could simultaneously utilize PlayStation Move controllers in the working sessions. This was because the controllers' motion tracking was done with a PlayStation Eye camera connected to a PlayStation 3 console, which was capable of serving controller data to multiple computers via Ethernet. Burdea (2004) has also utilized such a hardware multiplexing scheme with magnetic trackers in their teaching laboratory.

Each year a few students lamented about not being able conduct proper testing and iterative development at home, because they did not have the same VR hardware that was present in our university's virtual environment, where the application was intended to be run. Thus most of the interaction testing had to be performed in the weekly working sessions. In 2013 we alleviated this issue by borrowing spare Kinect devices to the student groups, who could use them outside the scheduled sessions.

Table 7 summarizes the evolution of our course, presenting its makeup and outcomes, which are related to research questions 1, 2 and 3. Course feedback results are not clearly reinforced by statistical tests ($p > 0.05$) due to too few answers, while the other results about VR application complexity and development experience are.

7.2. *Improvements for the Later Courses*

With our original project assignment arrangement, every year many student groups waited too long before starting to work industriously with the VR application development. Usually more than half of the work was done just two or three weeks before the project assignment deadline. Since 2014 we have attempted to alleviate this by adding one partial milestone session in-between the VR application development period, where all student groups must present a working prototype. We also started to keep a list of participants who attended the voluntary lectures and the working sessions, with the hope that the task of filling the list will remind students about their level of participation, and perhaps motivate them to attend more often.

During the first three iterations of the course, each year a few students reported experiencing a sense of unfairness, as one or two members of their group had participated

Table 7

Changes in the course and the utilized software and hardware juxtaposed with the resulting student-created VR applications and the students' experience

	2011	2012	2013
Changes in the course	• First course iteration	• A teaching assistant from the previous year was replaced with another	• Reduced lectures by one • Added one more teaching assistant to working sessions • Required students to use both Kinect and PS Move • Borrowed Kinect sensors to students
Input devices	• 6DOF Wiimotes	• PS Move	• PS Move • Kinect
Software	• RUIS for Processing	• RUIS for Processing (with added features)	• RUIS for Unity
VR application complexity (RQ1)	• Very few sound effects and complex 3D models • No applications with music or multiuser support	• 20% had music, 60% were multiuser applications	• More sound effects and complex 3D models • 67% had music, all were multiuser applications
Students' ratings about development experience (RQ2)		• RUIS learning curve was steeper than before	• Better programming interface in RUIS • More 3DUI building blocks in RUIS • Students were more willing to present their application for a possible employer
Course feedback (RQ3)		• Lectures received poor ratings	• Course rating was the highest • More positive feedback comments

very little in the project assignment. Since 2014 we have addressed this issue of social loafing with the following means:

1) Each student group is required to keep a shared spreadsheet where students mark the number of weekly hours that they have spent working on the course. This way the group members are able to monitor how the workload is distributed within the group.

2) At the end of the course a group can use peer assessment to adjust the grade of individual students by ±1 point as long as the sum of adjustments within a group is zero. In 2014 two groups and one in 2015 have utilized this possibility.

Hardware and RUIS for Unity have also improved: Oculus Rift HMD and Razer Hydra controllers have been available to our students since the 2014 course, and support for Kinect 2 was added for the 2015 course. In 2015 we stopped using Upponurkka's projection walls in favor of Oculus Rift HMDs due to financial reasons.

7.3. *Pedagogical Observations*

Results of the 2015 learning assessment address RQ4 and indicate how well our course performs in teaching VR application development according to our students. It should be noted that the course, its structure, and teaching methods had remained the same since 2013. Only one junior teaching assistant had been replaced with another. Students' rating of Likert statements shows that according to their self-evaluation they learned about 3DUIs, VR technology, VR application implementation, and basic VR concepts. The strongest and most univocal agreement among students was that after the course they were better at contributing to VR application creation. Answers to open-ended questions reveal that overall the students felt that their learning goals were met by the course's learning outcomes.

In general we have concluded that the pedagogical choices made when the course was initially revised for 2011 have been successful. The model taken from project-based learning, where students work in teams and self-organize their working has definitely been useful. Moreover, the project assignment has given student considerable freedom to choose their goal and the technology they apply. This has emphasized their own responsibility of acquiring necessary knowledge of applying the technologies, and thus activated them during the course. The course staff role has focused more on guidance, rather than transmitting information through lectures and closed exercises. In the future, we plan to strengthen the pedagogical design by better applying constructive alignment (Biggs and Tang, 2011), thus making the course learning goals and grading criteria more explicit for students.

7.4. *Practical Considerations for VR Course Organizers*

Here we present our experience and lessons learned as pragmatic considerations, which can serve other VR course instructors.

Organizing a VR course with practical assignments requires preparation. VR hardware and software must be acquired and tested beforehand. Besides theoretical VR knowledge, a VR course teacher should have a basic understanding of human-computer interaction and immersive technologies such as motion trackers and stereo 3D displays. Experience in playing or developing computer games is also beneficial, as our results suggest that student groups are most likely to create games if given a free choice over different VR applications.

Rapid development in computer graphics and VR hardware sets further challenges for VR course organizers. Lecture material and teaching laboratory equipment should be kept up to date when possible. Otherwise the most avid students who follow the VR industry could be sorely disappointed.

There should be some prerequisites for the students as well, so that they will not be overwhelmed by the amount of knowledge required to develop a VR application. We recommend that before taking a VR course, computer science students should have an already completed course in computer graphics or game development. In an interdis-

ciplinary course with art students, it is desirable that the art students have previously completed courses about 3D modeling tools or user interface design.

Our 3DUI questionnaire results show that students consistently rated the most severe development difficulty to be the inability to properly test their application's 3DUI with just a mouse and a keyboard of their personal computers. Likewise, limited access to the VR workstation was the most commonly mentioned issue in our students' comments. With the rise of affordable VR technology, VR course organizers can make equipment widely available and even borrow it to the students, so that they can develop and test applications at their own convenience. This would have been inconceivable in the past, when VR peripherals cost thousands of dollars apiece.

It is unlikely that there ever will be one perfect VR toolkit that makes others obsolete. Different toolkits have their own advantages in different VR application domains. Similarly, there is no one perfect toolkit for teaching VR; the choice of a toolkit depends on the VR course, its learning goals, teaching methods, available hardware, and background of its students.

Whether you are organizing a VR or a game development course, the adopted software toolkit affects the quality and the complexity of the applications created by students. With certain toolkits it is necessary for developers to "re-invent the wheel" when it comes to 3D selection and manipulation of objects, or even lower level features like display management. In some VR courses this might be appropriate; in others a VR toolkit with high-level features is more suitable. For our course where the goal has been to develop a functional, immersive VR application, Unity 3D software suite combined with our RUIS for Unity toolkit has provided much better results when compared to RUIS for Processing that we used earlier.

If a VR course employs a VR toolkit that does not require knowledge about compilers or linkers, teaching assistants are spared from spending time on helping students to set up the development environment and build configurations. Getting started with VR application creation can be as easy as installing the development software suite with an automatic installer and running an example project.

When evaluating different VR toolkits for course use, instructors should also take into account available documentation and the size of developer community for each toolkit. Good documentation and a lively community help students tremendously.

Based on our experience it is obvious that some kind of a 3D authoring tool is required when developers are building 3D worlds for VR applications or computer games. The most striking difference between working with Processing and Unity 3D is that the former offers a mere code editor, while the latter provides an integrated asset pipeline and a 3D scene editor that help importing content and composing 3D scenes. The increase in students' VR application complexity with RUIS for Unity is evidence about the importance of such tools for a VR toolkit. This notion is supported by Varcholik *et al.* (2009), who listed scene management and asset pipeline as secondary components in their list of requirements for a 3DUI research platform.

Our experience of teaching with the aid of the Upponurkka virtual environment is that CAVEs are well suited for VR education; they allow the whole class to see VR applications being demonstrated and multiuser applications can be used by several stu-

dents simultaneously. When organizing a VR course that primarily relies on HMDs, we recommend that a projector or a large-screen TV is used to mirror the HMD's view, so that the students can see the VR application from the user's perspective. This is especially important when demonstrating VR applications during a lecture, or when students are testing their own applications in working sessions. Relying only on HMDs discourages the creation of multiuser applications, because each HMD needs its own computer, which requires implementing a more complicated VR application that runs on a network of computers.

8. Conclusion

Consumer VR is a fast-growing industry and VR hardware is becoming increasingly affordable, which has implications for VR education. In this paper we shared our experiences of teaching VR application development with freely available software and off-the-shelf VR peripherals over the duration of five years. We reviewed existing literature on VR courses and reported the structure of our own course, its learning goals, and outcomes.

The course evolution was described, and three different course iterations were compared with each other using student feedback, questionnaire answers, and applications that the student groups had created. We analyzed the student deliverables quantitatively and qualitatively, with the results indicating that the latest of the compared course iterations was the most successful in terms of course feedback and the quality of the students' VR applications. Particularly the project assignment and support from teaching assistants received praising comments from students on that year.

The additional analysis of learning assessment results from 2015 showed that according to the students our course is successful in teaching VR application development and making them meet their personal learning goals.

At the heart of our course is the RUIS toolkit, which we have created to simplify VR application development for students and hobbyist developers. We reported the improvements to RUIS that we implemented over the years, presenting and statistically comparing student-created VR applications from each year. This comparison introduced evidence of how software toolkits can affect the development experience and the quality of applications created with them; the latest RUIS version reduced development difficulties, increased the complexity of the applications, and heightened the students' willingness to showcase them.

Finally we described how we have further improved our course since 2013, and summarized our teaching experience from the five years of organizing the course. This summary consisted of several practical considerations for VR course instructors, many of which are applicable also for game development and human-computer interaction courses: We discussed what to take into account when choosing a software toolkit and how different toolkits can support different learning goals. We also highlighted the importance of providing students sufficient access to VR hardware and presented further hardware related considerations for VR course organizers. Likewise, we emphasized

the importance of students having enough face-to-face support from teaching assistants during hands-on working sessions and introduced other points about teaching VR application development.

Our conclusion is that a VR course can be affordable for educators and uncomplicated for students.

References

Bell, D.H. (1996). Teaching virtual reality. *ACM SIGCSE Bulletin*, 28(2), 56–61.

Biggs, J., Tang, C. (2011). *Teaching for Quality Learning at University*. McGraw-Hill International.

Bonwell, C.C., Eison, J.A. (1991). Active learning: creating excitement in the classroom. *School of Education and Human Development*, George Washington University, Washington, DC.

Bowman, D.A., Kruijff, E., LaViola Jr, J.J., Poupyrev, I. (2004). *3D User Interfaces: Theory and Practice*. Addison-Wesley.

Burdea, G., Coiffet, P. (2003). *Virtual Reality Technology*. John Wiley and Sons, Hoboken, New Jersey.

Burdea, G.C. (2004). Teaching Virtual Reality: Why and How? *Presence: Teleoperators and Virtual Environments*, 13(4), 463–483.

Cliburn, D.C., Miller, J.R., Doherty, M.E. (2010). The design and evaluation of online lesson units for teaching virtual reality to undergraduates. In: *Frontiers in Education Conference (FIE)*, IEEE.

Cruz-Neira, C., Sandin, D.J., DeFanti, T.A. (1993). Surround-screen projection-based virtual reality: the design and implementation of the CAVE. In: *Proc. Conference on Computer Graphics and Interactive Techniques*, 135–142.

Durel, R.J. (1993). The capstone course: a rite of passage. *Teaching Sociology*, 21(3), 223–225.

Green, M., Jacob, R. (1991). SIGGRAPH '90 workshop report: Software architectures and metaphors for non-WIMP user interfaces. *Computer Graphics*, 25(3), 229–235.

Häfner, P., Häfner, V., Ovtcharova, J. (2013). Teaching methodology for virtual reality practical course in engineering education. *Procedia Computer Science*, 25, 251–260.

Herbelin, B., Ciger, J. (2008). Teaching and learning immersion and presence. In: *11th Annual International Workshop on Presence*. Padova, Italy, 305–313.

Juuti, K., Lavonen, J. (2006). Design-based research in science education: one step towards methodology. *Nordic Studies in Science Education (NorDiNa)*, 4, 54–68.

King, A. (1993). From sage on the stage to guide on the side. *College teaching*, 41(1), 30–35.

Lee, E.A.-L., Wong, K.W. (2008). A review of using virtual reality for learning. In: *Transactions on Edutainment I*. Springer, 231–241.

Linhoff, J., Settle, A. (2009). Motivating and evaluating game development capstone projects. In: *Proceedings of the 4th International Conference on Foundations of Digital Games*. ACM, 121–128.

Lokki, T., Ilmonen, T., Makela, W., Takala, T. (2006). Upponurkka: an inexpensive immersive display for public vr installations. In: *Virtual Reality Conference, 2006*, 315–315.

Miyata, K., Umemoto, K., Higuchi, T. (2010). An educational framework for creating VR application through groupwork. *Computers & Graphics*, 34(6), 811–819.

Pantelidis, V., Auld, L. (2003). Teaching virtual reality courses online. In: *GLOBAL-ED*. Melbourne, Australia.

Parberry, I., Roden, T., Kazemzadeh, M.B. (2005). Experience with an industry-driven capstone course on game programming. *ACM SIGCSE Bulletin*, 37(1), 91–95.

Ritzhaupt, A.D. (2009). Creating a game development course with limited resources: An evaluation study. *ACM Transactions on Computing Education (TOCE)*, 9(1).

Rose, M. (2014). *Oculus Rift has sold more than 100,000 dev kits*. Gamasutra.
`http://gamasutra.com/view/news/220125/Oculus_Rift_has_sold_more_than_100000_dev_kits.php`

Seymour, N.E. (2008). VR to OR: a review of the evidence that virtual reality simulation improves operating room performance. *World Journal of Surgery*, 32(2), 182–188.

Sherman, W.R., Craig, A.B. (2002). *Understanding virtual reality: Interface, application, and design*. Elsevier.

Stansfield, S. (2005). An introductory VR course for undergraduates incorporating foundation, experience and

capstone. In: *ACM SIGCSE Bulletin*. ACM, 197–200.

Steed, A. (2008). Some useful abstractions for re-usable virtual environment platforms. *Software Engineering and Architectures for Realtime Interactive Systems-SEARIS*.

Stone, R. (2001). Virtual reality for interactive training: an industrial practitioner's viewpoint. *International Journal of Human-Computer Studies*, 55(4), 699–711.

Sutherland, I.E. (1968). A head-mounted three dimensional display. In: *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*. ACM, 757–764.

Takala, T.M. (2014). RUIS: A toolkit for developing virtual reality applications with spatial interaction. In: *Proceedings of the 2Nd ACM Symposium on Spatial User Interaction*. ACM, New York, NY, USA, 94–103.

Takala, T.M., Pugliese, R., Rauhamaa, P., Takala, T. (2011). Reality-based user interface system (RUIS). In: *Proceedings of the IEEE Symposium on 3D User Interfaces 2011*. 141–142.

Takala, T.M., Rauhamaa, P., Takala, T. (2012). Survey of 3DUI applications and development challenges. In: *Proceedings of the IEEE Symposium on 3D User Interfaces 2012*. 89–96.

Thomas, J.W. (2000). A review of research on project-based learning. *San Rafael, CA: Autodesk Foundation*.

Varcholik, P.D., LaViola Jr, J.J., Hughes, C. (2009). The Bespoke 3DUI XNA Framework: a low-cost platform for prototyping 3D spatial interfaces in video games. In: *Proceedings of the 2009 ACM SIGGRAPH Symposium on Video Games*. 55–61.

Wingrave, C.A., LaViola Jr, J.J. (2010). Reflecting on the design and implementation issues of virtual environments. *Presence: Teleoperators and Virtual Environments*, 19(2), 179–195.

Zagal, J.P., Sharp, J. (2011). A survey of final project courses in game programs: considerations for teaching capstone. In: *Digital Games Research Association Conference*.

Zara, J. (2006). Virtual Reality course – a natural enrichment of Computer Graphics classes. In: *Computer Graphics Forum*, Wiley Online Library, 105–112.

Zimmerman, G.W., Eber, D.E. (2001). When worlds collide!: an interdisciplinary course in virtual-reality art. In: *ACM SIGCSE Bulletin*. ACM, 75–79.

Zyda, M. (2005). From visual simulation to virtual reality to games. *Computer*, 38(9), 25–32.

T.M. Takala is a doctoral student, who will graduate in 2016 from Aalto University. He has worked full-time with augmented and virtual reality since 2007, designing and implementing AR and VR applications with a focus on 3D user interfaces and full-body interaction.

L. Malmi is Professor of computer science at Aalto University, and Vice dean of education at Aalto School of Science. His main research areas include computing education research and educational technology, especially in the context of programming education.

R. Pugliese received his Doctor of Science degree from Aalto University, Finland. He is a media artist and embodied interaction researcher based in Helsinki. His work consists of compositions, installations and performances. Often concerned with memories and the concept of interface, his pieces can incorporate animation, sound, moving objects and the augmentation of physical media using programmed hardware and software.

T. Takala is a professor in the department of Computer Science at Aalto University, Finland. His research interests include virtual reality, affective computing, embodied/enactive interfaces, and computational creativity, with applications in art and entertainment.