Survey of 3DUI Applications and Development Challenges

Tuukka M. Takala Päivi Rauhamaa Tapio Takala

Department of Media Technology, Aalto University, Finland

ABSTRACT

We present a survey of the current generation of 3D user interface (3DUI) applications, their developers, and development issues. 3DUI developing experiences from 71 developers – as well as information about 56 unique 3DUI applications – were collected and analysed.

The statistics presented in this paper give a broad view over the 3DUI field and indicate current trends in 3DUI applications, such as the most commonly used hardware and software. Popularity of Kinect and Move controllers are compared as we discuss how entertainment industry can embrace hobbyist innovation.

Our results show that reuse of high-level interaction features is rare, even though 3DUI toolkits are widely used; most developers painstakingly implement common features like navigation and object manipulation for each 3DUI application, as opposed to inheriting them from a toolkit.

We also introduce criteria for measuring 3DUI development difficulty and two ways of benchmarking 3DUI toolkits, and present example benchmarks using data from our survey.

KEYWORDS: 3D user interfaces, post-WIMP interfaces, natural interaction, user interface development.

INDEX TERMS: H.5.2 [Information Interfaces and Presentation]: User Interfaces—Prototyping, Interaction Styles

1 INTRODUCTION

Until recently there were very few people who could access and implement 3D user interfaces (3DUI), gestural interfaces, and other novel interaction styles. This is now rapidly changing with the introduction of Nintendo Wii Remote (WiiMote), Microsoft Kinect, and other affordable movement sensing devices that have been widely adopted by hacker and research communities [2][7][13].

Kinect in particular has sparked the imagination of amateur developers which has resulted in numerous homebrew 3DUI applications [6]. It is our belief that a large community of hobbyist developers can bring innovation in to the field of 3DUIs, whether that is in the form of new interaction techniques, interface metaphors, or emerging 3DUI standards. Such user-led innovations have already been demonstrated to be influential in other areas of software development, integrated circuit industry, food industry, and video games industry [1][5].

Although devices like Kinect have made it easier to get started with 3DUI development, many fundamental difficulties still remain. Virtual environment (VE) research community has been particularly active in finding and reporting these difficulties: Already in 1991 Green and Jacob [4] identified many issues regarding design and implementation of 3DUI applications. Several more recent papers focus exclusively on challenges in VE and 3DUI application development [8][9][12].

Wingrave and LaViola [12] suggested that if problems in design and implementation of VEs cannot be alleviated, this could hinder user interface advances in popular domains such as gaming, entertainment, mobile interaction, and augmented reality.

Several toolkits for developing 3DUI applications exist, which poses an additional problem for developers: How can one choose a 3DUI toolkit that suits one's skills and application requirements? Existing literature about comparison of 3DUI toolkits is limited: Figueroa et al. present a brief toolkit comparison based on programming language and coverage in input & output devices, interaction techniques, and content quality [3]. Zeitler ranks VE and multi-touch development toolkits using a score-based comparison, where a set of generic features is used to calculate a total score for each toolkit [14].

The research presented in this paper is motivated by our aspiration to make 3DUI development easier for hobbyists. We wanted to discover what the initial obstacles are in the beginning of 3DUI development. That purpose in mind we designed a questionnaire and announced it online, requesting 3DUI developers to participate.

2 SURVEY BACKGROUND

The questionnaire was organized into parts according to different topics: Personal background of the developer, developed 3DUI application, toolkits used, and development difficulties.

Many of the 3DUI development issues are related to exotic input devices. Therefore it was required from questionnaire participants that they had been developing at least one 3DUI application that used other controllers than just mouse and keyboard. Each participant was asked to choose one 3DUI application that they had been developing, and base his/her answers on experience with that application. This way the answers contained information about 3DUI applications and the development processes that created them.

Most of the questions were multiple choice questions with an option to add one's own answer. In some cases the participants were asked to rate a set of 3DUI development aspects using Likert scales. All questions were mandatory with the exception of open ended questions, and thus there were no issues with missing data. The questionnaire is still available online[1].

2.1 Participants

We sought questionnaire participants by contacting our colleagues and making announcement posts in 3DUI related online forums[2]. Each participant's answers were evaluated by us so that only valid data would be collected. Answers from 71 participants were accepted, and one participant was disqualified due to inconsistent answers and failure to reply to our confirmation email.

Table 1 presents how the participants were reached. In total there were 44 participants who had not been in contact with us previously. Ten of them we contacted to ask to participate in the questionnaire after coming across their 3DUI applications in internet.

{ tmtakala, paivi.rauhamaa, tapio.takala } @ tml.hut.fi

IEEE Symposium on 3D User Interfaces 2012
4-5 March, Orange County, CA, USA
978-1-4673-1205-9/12/$31.00 ©2012 IEEE

[1] http://goo.gl/RuZHT

[2] OpenNI Google Group, 3DUI mailing list, gamedev.net, glovepie.org., forum.openframeworks.cc, forum.processing.org, wiimoteproject.com, nuigroup.com, kinectforums.net,

Table 1. Questionnaire participants

Participant	Count
People who found the questionnaire by themselves	34
Students of a virtual reality course organized by us	14
Previously unknown people who we asked to participate	10
Colleagues within our local, Finnish network	10
Colleagues within our network abroad	3
Total	71

2.1.1 Student Participants

Participating in the questionnaire were 14 students taking a course in virtual reality, where 2-3 students formed teams that each developed a 3DUI application. Every team was using the same custom hardware and programming environment to create their 3DUI application. In order to avoid biased results, students' answers were omitted when calculating 3DUI application statistics. Answers from students were only used for developer demographics and example of difficulty-based benchmark.

The students' grades for the course were decided before they answered in the questionnaire, but study credits were registered only after everyone had answered. This was done to maximize questionnaire participation and minimize doubts about coupling of grades with questionnaire answers.

3 Results

Below are the main statistics extracted from our questionnaire.

3.1 Developer Demographics

The charts in this section are color coded so that each colour represents a different developer type: Professional, hobbyist, and student. In total there were 46 professionals (researchers or practitioners), 11 hobbyists, and 14 students.

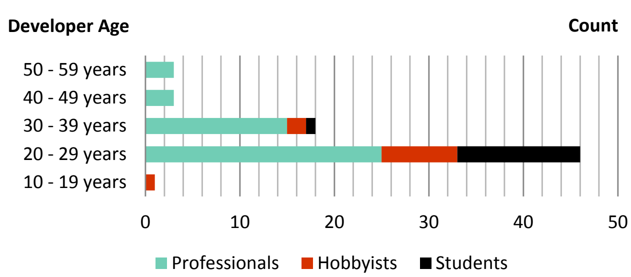

Figure 1. Age distribution of 71 questionnaire participants

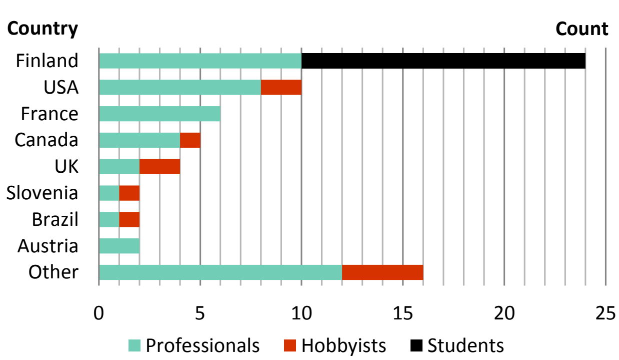

Figure 2. Countries where the developers were based in

Figure 1 shows the age distribution among the developers, most of who were in their 20s. Majority of 67 developers participating in our questionnaire were male, and 4 were female.

Countries where the developers were based in are presented in Figure 2. The 'Others' category of the figure consists of 16 developers, one each from Colombia, Egypt, India, New Zealand, South Korea, nine different European countries, as well as two developers whose countries we were unable to determine.

Figure 3 reveals that very few students and hobbyists had more than one year of experience with developing 3DUI applications, compared to the several years of most professionals.

This experience gap can also be seen from Figure 4, which compares the number of 3DUI applications that each participant had been developing to the number of those whose development they had completed. Participants were advised to consider an application to be completed, if it was developed far enough to be usable in a meaningful way. The figure is missing one data point from a professional who reported to have been involved with around 30 3DUI applications and completing development of most of them.

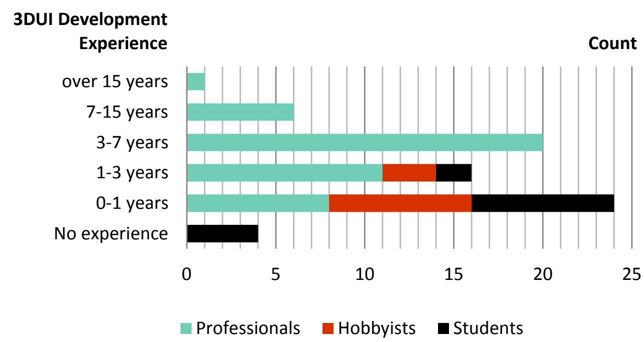

Figure 3. Distribution of 3DUI development experience in years among the 71 participants

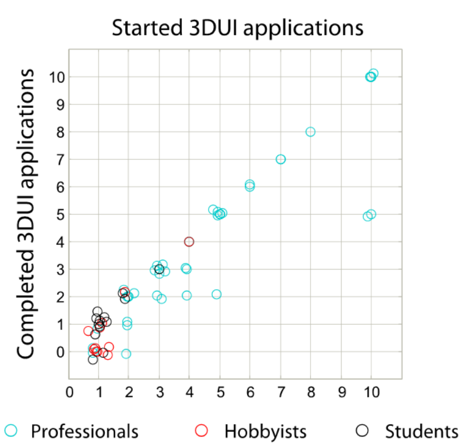

Figure 4. Number of 3DUI projects started by each participant compared to the number of projects brought to completion. Overlapping data points have some added jitter to show their multitude.

3.2 3DUI Application Statistics

The 3DUI applications reported by the questionnaire participants varied from very simple, Kinect controlled virtual keyboards to CAVE-based surgery simulators, so the collected data represents a wide range of applications and intended usages. Each participant was asked to base his answers on one 3DUI application that he had been developing, regardless of whether the development process was still ongoing, completed, or abandoned.

Two of the participants had been developing the same 3DUI application, and in that case only one developer's answers about the application were included. Results in this section were calculated from a total of 56 unique 3DUI applications.

Participants were asked to classify their 3DUI application development project as a research, hobby, or commercial project. We expected a somewhat even distribution, but were surprised to find that a vast majority of participants' applications were developed as part of research, as seen from Figure 5. A number of developers identifying themselves as hobbyists reported a 3DUI application that was made as a research project, and likewise some professionals reported a hobby project.

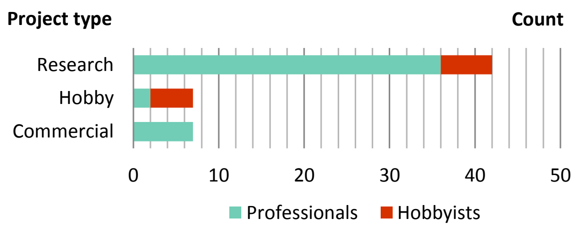

Figure 5. 3DUI application development project types as reported by professional and hobbyist developers

In the subsequent bar charts – starting from Figure 6 – the different color bars represent different types of 3DUI application development projects: Research projects are blue, commercial projects are light green, and hobby projects are dark red.

Figure 6 shows that most of the applications were released within past two years, and only 9 applications out of 56 were released prior to year 2009.

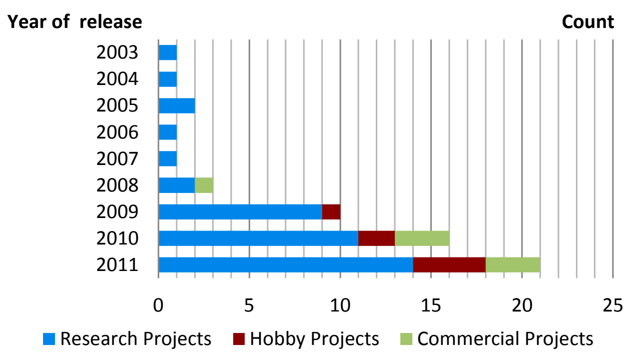

Figure 6. Year of release distribution of 56 different 3DUI applications

For some participants the question about release year might not have been applicable, if the 3DUI application was still incomplete or if its development was abandoned. Therefore the participants were asked to report the status of their 3DUI application with a multiple choice question, whose answer distribution in presented in Figure 7.

Figure 8 shows the reported number of developers working on each 3DUI application. The participants were not asked directly for any approximation of development efforts, such as person-months. This is a shortcoming that should be addressed in any future questionnaires.

The developers were also asked to estimate the number of users their application had when it was most popular. These results are in Figure 9. As one might expect, 3DUI applications made as hobby and research projects rarely become widely used. The three applications with most users were all located in a public place, provided entertainment, and they all used a projector, Kinect, OpenNI, and open source graphics toolkits.

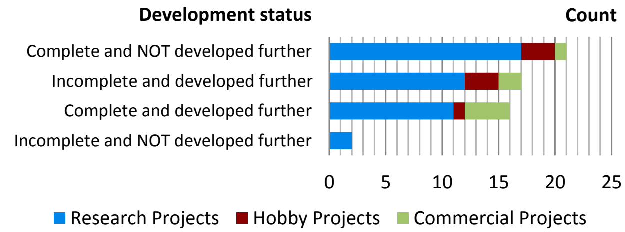

Figure 7. Development status distribution of 56 reported 3DUI applications

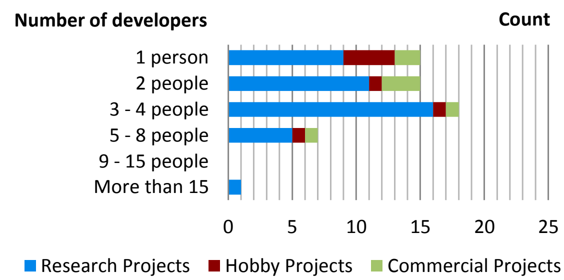

Figure 8. Developer force distribution of 56 different 3DUI applications

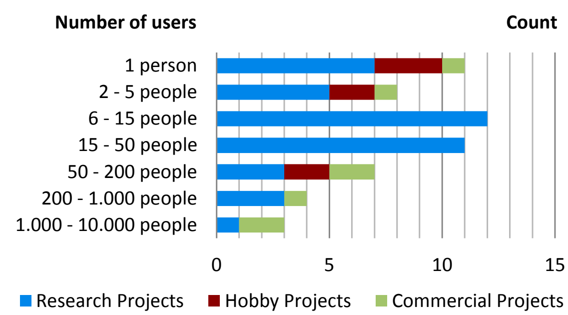

Figure 9. User population distribution of 56 different 3DUI applications

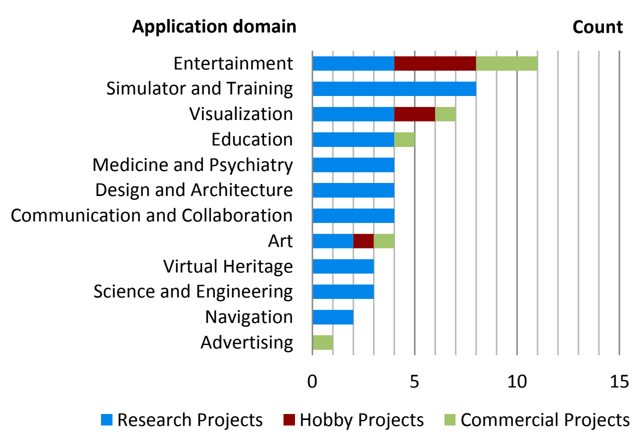

Figure 10. Most common application domains among 56 different 3DUI applications

The different application domains are presented in Figure 10, where it can be seen that entertainment is the preferred domain for commercial and hobby projects.

The most common programming language used in development was C/C++ with 27 applications, followed by C# with 11 applications and Java with 8 applications. Table 2 presents some other miscellaneous information about the 3DUI applications.

Table 2. Miscellaneous traits among 56 applications

Application trait	Count
Supports multiple operating systems	20
Input devices can be simulated with mouse & keyboard	25
Supports multiple simultaneous users	19

3.2.1 Input and Output Devices

Kinect, cameras, and WiiMote were most popular input devices among developers, as seen from Figure 11. The input device survey in our questionnaire included over a dozen of device choices as check boxes and an option to report any additional devices.

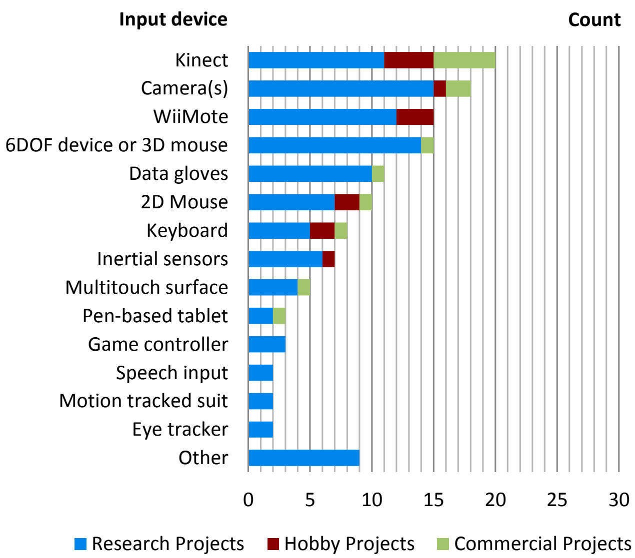

Figure 11. Most common input devices among 56 reported 3DUI applications. Grouping 6DOF device and 3D mouse together was a mistake in questionnaire design

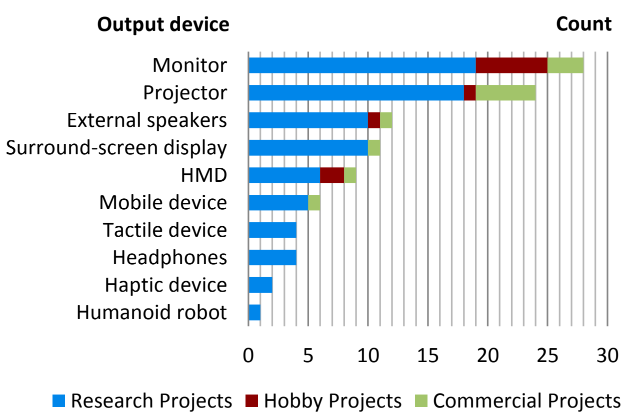

Figure 12. Most common output devices among 56 different 3DUI applications

Many 3DUI applications had several input devices, either to be used simultaneously or as an alternative input device. Therefore the total device count in Figure 11 is greater than 56, the number of applications that were used to calculate the input device statistics. Maximum number of input devices used in a single 3DUI application was nine, while the median over all the different applications was two.

The 'Other' category of Figure 11 includes nine input devices that were reported only once each: a time-of-flight based depth camera, locomotion interface, biological sensor, dance mat, WiiMote Nunchuk device, Wii Balance Board, and two haptic devices: Phantom Omni and HapticMaster.

Figure 12 presents output devices used by the 3DUI applications. Interestingly one research project had a humanoid robot that reacted to commands from Wiimote and Kinect.

3.2.2 3DUI Toolkits

Most popular 3DUI toolkits among the developers are presented in Figure 13. Other toolkits that were reported only once are: ArUco, Bespoke 3DUI XNA Framework, CAVElib, EON SDK, MiddleVR, OpenSpace3D, Optitrack Arena, Unofficial Kinect ToolKit, and Vizard.

The developers also used supplemental development libraries and frameworks, most of which are listed in Table 3. Other libraries and frameworks that received only one mention are: C4 engine, JSR184, OpenCV, QT, TactaBox, Vuzix SDK, Windows 7 Touch Screen, and vvvv.

Only three developers reported using a high-level 3DUI description language: InTml and two unspecified XML-based 3DUI description languages were mentioned.

Table 3. Supplemental development libraries

Library	Count
OpenSceneGraph	5
Ogre3D	5
Unity3D	4
XNA	2
OpenFrameworks	2

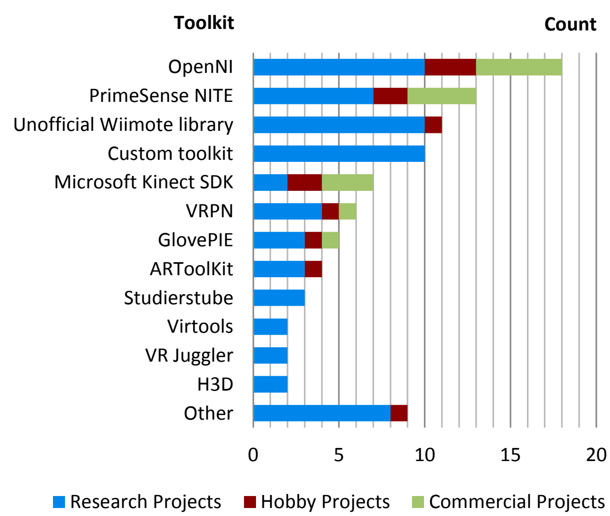

Figure 13. Most common 3DUI toolkits used in the development of 56 different applications

3.2.3 3DUI Application Features

Below is a list of common 3DUI features that can be used to roughly characterize 3DUI applications. Some of the features are inspired by Varcholik et al. [11] who compiled a set of requirements that are necessary for a 3DUI research toolkit:

F1. 3D stereographics

F2. Head tracked view rendering

F3. Full-body interaction

F4. Two-handed interaction

F5. Finger interaction

F6. Gesture recognition

F7. Navigation techniques

F8. Object manipulation techniques

F9. Physics engine

F10. 3D audio

It should be noted that the above list is not meant to be complete. There are many other important aspects for 3DUI toolkits, such as input device abstractions, programming interface, and error handling.

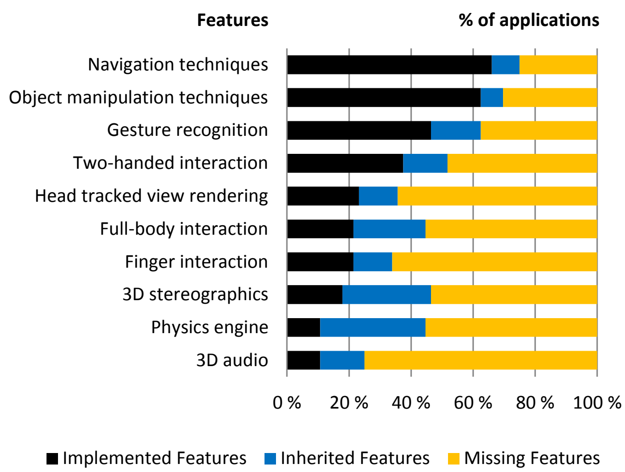

Figure 14. Presence of 3DUI features in 56 unique applications and their origin: Whether they were implemented by developers or inherited from a 3DUI toolkit

Figure 14 shows how commonly the different 3DUI features (F1-F10) were present within the 56 unique applications. It also reveals how often those features had to be implemented by the developers, or if they were inherited from 3DUI toolkit(s). For example: A full-body interaction feature (F3) could be inherited by using OpenNI because it provides relatively high-level data abstraction for tracked human bodies and major limbs. On the contrary, a developer using solely OpenNI would have to implement finger interaction feature (F5) by himself, because fingers are not present OpenNI's human body model.

A high percentage of missing features could reflect that the feature in question was not important, was difficult to implement, or was often missing from the toolkits used. The ratio between inherited and implemented feature percentages is of particular interest. By comparing the implemented and inherited bar's width in the figure, it is clear that most features are implemented rather than inherited.

3.3 Open Ended Questions

The questionnaire results obtained by calculating statistics from participant answers indicate general trends in 3DUI application development, but are unable to express some of the developer needs that arise from different circumstances. We counterbalanced this with qualitative analysis of developers' answers to open ended questions. The questions concerned the developers' problems in development, and their reasons for satisfaction or dissatisfaction with their 3DUI application. Answering these questions was voluntary.

In the questionnaire, 16 out of 71 participants described their development problems and 29 out of 71 explained their reasons for satisfaction or dissatisfaction with their application. All participants were also asked to rate how satisfied they were with their application on a Likert scale, and 75% expressed some level of positive satisfaction. Dissatisfaction was expressed more in the open-ended question, as 9 participants explained their reasons for positive satisfaction whereas 20 explained why they were not satisfied. Reasons for dissatisfaction were often related to development issues: *"There is a huge gap between great ideas and practical implementation when dealing with gesture recognition and Kinect."*

A quarter of the reported problems in development and quarter of the reasons for dissatisfaction concerned issues with testing of the application: "*Testing requires full interaction, making it hard for one person to rapidly test and tweak at the same time. Mouse and keyboard input for testing is of no real use; if you can do what you want with mouse and keyboard, then you are not exploiting the real capabilities*." Several participants expressed that development is time consuming which leads to having too little time for testing: *"Developing 3DUIs takes a lot of time, testing them is even worse"*.

Quarter of the reported problems in development and fifth of the reasons for dissatisfaction concerned hardware. Several participants mentioned issues with calibration; it was either too complicated or that it had to be done too often. Some of the hardware issues were usability related: *"[The 3DUI] is not easy to use for novices, due to novelty of the devices."*

Fifth of the reported problems in development and fifth of the reasons for dissatisfaction were related to 3DUI toolkits. Developers expressed their need to have better documentation and examples for 3DUI toolkits.

4 3DUI Toolkit Benchmarking

Surveys can be used to extract information about 3DUI toolkits and development issues. Below we examine how that information can be further processed for benchmarking 3DUI toolkits.

4.1 Measuring Development Difficulty

We present a list of 18 statements that describe different aspects of development difficulties. Some of the statements are based on our own experience; some are influenced by Wingrave and LaViola's work [12]. Statistical knowledge about development challenges can be collected by asking developers to rate their agreement with the statements while considering their experience with 3DUI application development.

The participants of our questionnaire rated each statement on a seven point Likert scale (where 1 indicated strong disagreement and 7 indicated strong agreement), and based their ratings on development experience with the 3DUI application that they had reported. Each statement was constructed in a way that a higher rating signifies that more difficulty was experienced.

4.1.1 Difficulties in Early Development Phases

The 10 statements below describe difficulties in the beginning of 3DUI application development:

A1. The input devices required by my 3DUI application were too expensive

A2. The output devices required by my 3DUI application were too expensive

A3. Getting input device drivers to work was difficult

A4. 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

A5. Device input data was too low-level for quickly getting started with my 3DUI application

A6. Lack of documentation or tutorials about the 3DUI toolkit made the development difficult

A7. The 3DUI toolkit had a steep learning curve

A8. The development was difficult because the 3DUI toolkit had a bad programming interface

A9. Programming in general was difficult

A10. Creating mathematical algorithms required by my application's 3DUI was difficult

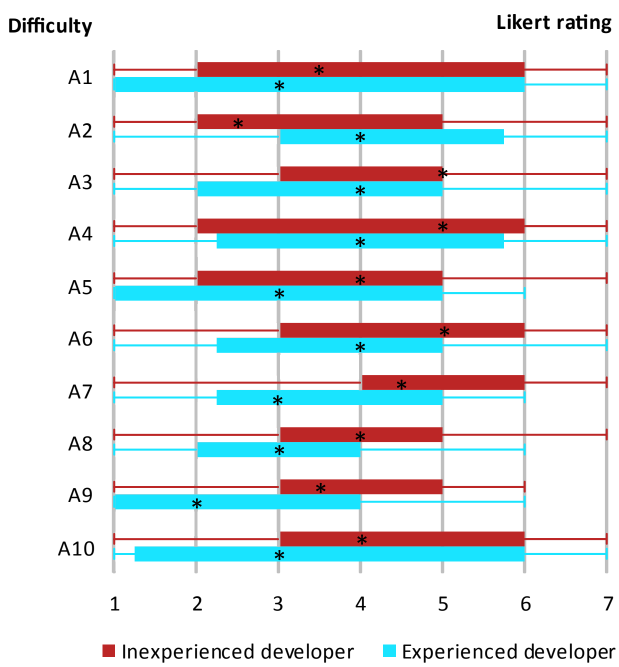

Figure 15. Severity of initial development difficulties in 3DUI application development, presented with a boxplot of Likert ratings. Edges of the color boxes represent 25% and 75% quantiles, and asterisks mark medians

Figure 15 illustrates Likert rating distributions for statements A1-A10 as rated by participants of our questionnaire, partitioned into two groups of developers: 27 experienced developers who had been developing four or more applications and 30 developers who had been developing less than four; i.e. the partitioning was done using experience threshold of four developed 3DUI applications. As can be seen from the figure, the less experienced developers tended to meet more difficulties in every category, except the case of output devices being too expensive (statement A2).

Analysis with Wilcoxon rank-sum test revealed two statistically significant differences: Inexperienced developers rated higher development difficulties with regard to statements A8 ($z = -2.5$, $p = 0.014$) and A9 ($z = -2.4$, $p = 0.018$), when compared to experienced developers. Statistically significant differences ($p < 0.05$) remained between inexperienced and experienced developer groups' ratings for statements A8 and A9, when we repeated the rank-sum test with two other developer partitions, by using experience threshold of three and five applications. We see this as an indication of inexperienced developers having more trouble with 3DUI toolkit programming interfaces (A8) and programming in general (A9).

4.1.2 Difficulties in Later Development Phases

The eight statements below describe difficulties that may occur later in 3DUI 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

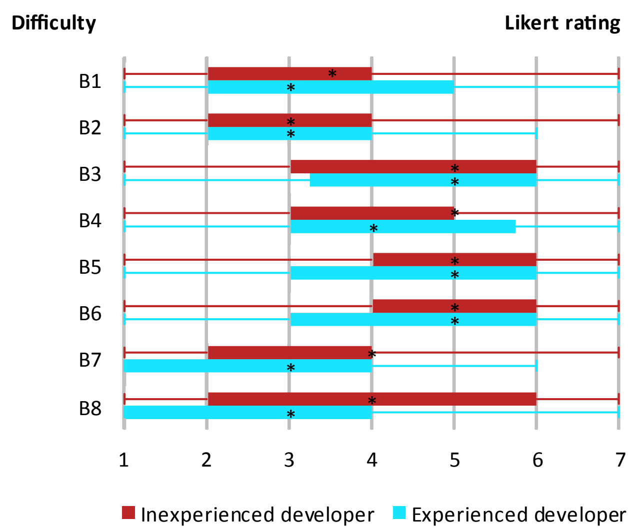

Figure 16. Severity of later development difficulties in 3DUI application development, presented with a boxplot of the Likert ratings

While statements B7 and B8 might not be applicable to some of the development projects, the other statements are relevant for all but the simplest of 3DUI applications.

Our questionnaire participants' Likert rating distributions for statements B1-B8 are presented in Figure 16: Differences in severity of later development difficulties are less distinct between experienced and inexperienced developer groups, and no statistically significant differences were found.

4.2 Difficulty-based Benchmarking

Here we introduce the concept of ranking 3DUI toolkits according to expected development difficulty that is estimated with data from a questionnaire similar to ours. This benchmark model aims to help developers, especially hobbyists, to choose a 3DUI toolkit that causes the least development trouble.

Figure 15 and Figure 16 illustrated how 3DUI development difficulties between different groups of developers can be measured by comparing their Likert rating distributions for statements A1-A10 and B1-B8. We modify this idea and divide questionnaire participants into groups according to the toolkit(s) that they have used in the 3DUI application development. These groups' Likert rating distributions for statements A3-A8 and B2-B6 can then be used to compare the toolkits. The basic principle of the comparison is that easy-to-use toolkits have lower ratings.

Difficulty-based benchmark can be calculated when each rating of a statement is associated with a development process whose toolkit(s) are known. With a sufficient number of ratings from developers, the benchmark can indicate differences between 3DUI toolkits.

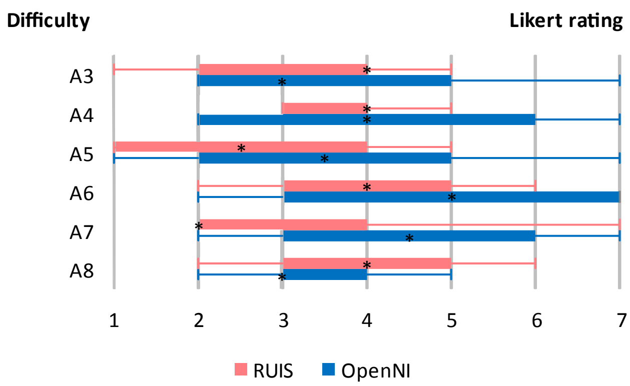

Figure 17. Severity of initial development difficulties in 3DUI application development, presented with a boxplot of Likert ratings. Edges of the color boxes represent 25% and 75% quantiles, and asterisks mark medians

In Figure 17 we present an example benchmark that was obtained by comparing Likert ratings of 18 OpenNI developers and 14 students, who were all using RUIS – a 3DUI toolkit based on Processing. In this case the only statistically significant difference ($z = -2.3$, $p = 0.021$) was found in the case of statement A7 ("The 3DUI toolkit had a steep learning curve"). We see this as a weak indication that OpenNI has a steeper learning curve than RUIS toolkit.

In the case of questionnaire participants who report using more than one 3DUI toolkit, it is not clear how much each toolkit contributes to development difficulties of statements A3-A8 and B2-B6. One way to handle this problem is to disregard ratings from those participants. Alternatively, individual ratings from participants with multiple toolkits can be included in the difficulty-based benchmark, by letting them affect the benchmark with same weight for each toolkit. Some of the toolkits would then receive unfounded ratings, but overall results from a large number of developers should still be informative.

4.3 Feature-based Benchmarking

We propose the idea of benchmarking 3DUI toolkits by comparing applications' inclusion of features (F1-F10) using data obtained with a questionnaire such as ours. This benchmark could help developers to identify the toolkit that provides most of the essential features required by their application.

The feature-based benchmarking works by ranking 3DUI toolkits according to the ratio between inherited and implemented 3DUI features among the surveyed applications. The benchmark can be performed for each 3DUI feature of interest. Number of applications where the feature is missing does not affect the benchmark, as those applications might not have needed the feature in the first place.

An example is presented in Figure 18, where we can see the presence of full-body interaction feature (F3) in applications that were created using some of the more popular toolkits. Feature-based benchmark ranks Microsoft Kinect SDK first with our survey data, because all seven 3DUI applications developed with it had inherited full-body interaction feature. This suggests that if a developer wants to include full-body interaction in a 3DUI application, using Microsoft Kinect SDK could be a slightly better choice than PrimeSense NITE, whose features were apparently not enough for some developers in our survey.

In practice both toolkits provide similar full-body interaction capabilities, and the choice between them depends on the developer and application.

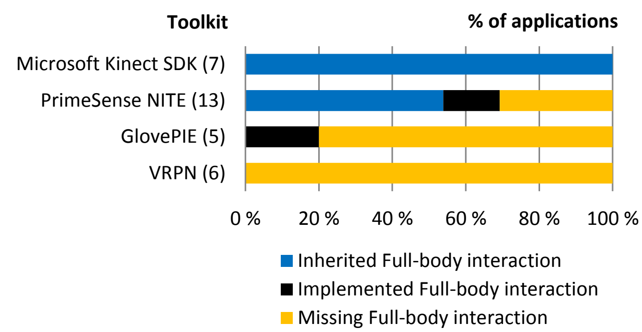

Figure 18. Full-body interaction in applications that were created using four popular 3DUI toolkits. The number of applications used to calculate the percentages are presented next to the name of toolkit involved

In our questionnaire each participant reported all toolkits that were used to create his/her 3DUI application. Participants who used multiple toolkits were not required to report the contributing toolkit for each inherited feature. When calculating the example benchmark of Figure 18, each feature marked as inherited added a point for all of the toolkits used by the developer, regardless of which toolkit actually provided the feature. Benchmark inaccuracies introduced by such a convention are avoided when benchmarking data from questionnaires where the participant has to attribute each inherited feature with one toolkit.

5 Discussion

Somewhat surprising was the small number of hobbyists (11) participating in our questionnaire, despite our many announcements in Kinect and Wiimote developer forums. It could be that many of the hobbyists are also researchers in the 3DUI or related fields. While students of 3DUI project courses can be asked to participate, it is important to consider how to get other hobbyists to participate in any future questionnaires of this nature.

It is unlikely that there will ever be an ultimate 3DUI toolkit that answers all the needs of different developers. Requirements for a toolkit vary with the available hardware and purpose of the developed application. As a consequence, many 3DUI toolkits exist, and this creates a need for benchmarking them. The two methods presented in this paper can be used to rank toolkits according to expected development difficulties and ease of feature inheritance, if a sufficient body of 3DUI application development survey data is available.

The difficulty-based 3DUI toolkit benchmark is based on statements A3-A8 and B2-B6 whose rating can be affected by other factors besides toolkit choice: input and output devices, application domain, specific requirements for the application, etc. This can lead to misleading toolkit benchmark scores if there are too few survey participants and their 3DUI applications are diverse. We argue that with a sufficiently large number of participants the influence of other factors will even out and the benchmark can give useful information about 3DUI toolkits.

5.1 Kinect and Move: Two Different Strategies

According to our questionnaire, Kinect was the most popular input device among developers with 20 out of 56 applications using it. From applications released in 2011 alone, 12 out of 21 were using Kinect. Not a single developer reported to have used PlayStation Move controller, even though it was released in autumn 2010, over a month before Kinect.

One reason for this situation is that Move is yet another 6DOF tracking device, whereas Kinect offers affordable full-body tracking and depth acquisition. Second reason is that OpenNI, a third party software development kit (SDK) for Kinect, came out already in December 2010, whereas Sony released its Move.me SDK in July 2011.

Hobbyist applications using Move are very difficult to find using online search engines even though Move.me SDK has been out for over five months at the time of writing this paper. In contrast, dozens of Kinect applications made by hobbyists started to surface already in November 2010 [6], right after Kinect became available. This is also reflected in online forum activity: Kinect forums for hobbyist developers are bustling with discussion, while new posts appear very rarely in Move forums.

We propose that this popularity gap can be partly explained with the different strategies that Sony and Microsoft have chosen: Microsoft has been more thoughtful about hobbyists by providing a free Kinect SDK that is available for everyone and allowing the existence of OpenNI consortium's open source SDK. In fact, Johnny Lee from Kinect development team secretly arranged a competition in 2010, where a money prize was given to the first individual to create PC support for Kinect by 'hacking' it [10].

A hobbyist Kinect developer only needs a PC, free software, and Kinect (~$100) to get started. Sony's Move.me SDK on the other hand is only available in North America, costs $100, and requires that the developer has a PlayStation 3 console (~$250) in addition to PC and Move.

In the current situation Kinect is the preferred input device among 3DUI hobbyist developers, who are educating themselves to become future Kinect developers, and inadvertently buying into Microsoft's vision of what the future of 3DUIs is. Despite all this, our early tests with Move controllers show their promise for 3DUI applications. We plan to use them in our future projects and 3DUI courses alongside with Kinect and other affordable hardware.

A positive feedback loop could emerge between 3DUI hobbyist community and related technology companies, where innovations spread and accumulate between commercial developers and hobbyists. If Sony wants to attract more hobbyist developers and compete with Microsoft in this arena, they should consider releasing free device drivers for PC that do not require the console itself. Any device manufacturer, that wants to tap into the stream of hobbyist innovation, should aim to remove unnecessary barriers between their device and its potential developers.

6 Conclusion

In this paper we have presented an overview over current generation of 3DUI applications, their developers, and the issues involved. We introduced two ways of benchmarking 3DUI toolkits and criteria for measuring 3DUI development difficulty. Their use was briefly demonstrated.

Results from our statistical analysis suggest that the most common features in 3DUI applications are navigation and object manipulation. In most cases developers implement these features for each 3DUI application, as opposed to inheriting them from a 3DUI toolkit. This is congruent with Wingrave and LaViola's finding that *"It Is Easier to Build than to Reuse" [12]*.

We also found indications that when compared to experienced 3DUI developers, inexperienced developers have more trouble with 3DUI toolkit programming interfaces and programming in general.

Finally we discussed how game console companies can encourage hobbyist developers to use their products, and how this could benefit them.

References

[1] Y. Aoyama, H. Izushi, User-led Innovation and the video game industry, *IRP Conference*, London, May 22-23, 2008.

[2] J. Giles, Inside the Race to Hack the Kinect, *The New Scientist*, 208(2789):22-23, 2010.

[3] P. Figueroa, W. F. Bischof, P. Boulanger, and H.J. Hoover, Efficient comparison of platform alternatives in interactive virtual reality applications, *Int. J. Hum.-Comput. Stud.* 62(1), 73-103, 2005.

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

[5] E. Hippel and R. Katz. Shifting Innovation to Users via Toolkits, *Manage. Sci.* 48(7), 821-833, 2002.

[6] KinectHacks, http://kinecthacks.net, last visited: 18th of November, 2011.

[7] J.C. Lee. Hacking the Nintendo Wii Remote, *Pervasive Computing, IEEE*, 7(3): 39-45, 2008.

[8] A. Steed. Some useful abstractions for re-usable virtual environment platform. *SEARIS Workshop at IEEE Virtual Reality,* 2008.

[9] R. M. Taylor, J. Jerald, C. VanderKnyff, J. Wendt, D. Borland, D. Marshburn, W. R. Sherman, and M. C. Whitton. Lessons about virtual environment software systems from 20 years of VE building. *Presence: Teleoper. Virtual Environ.* 19(2): 162-178, 2010.

[10] D. Terdiman, Kinect developer claims credit for hack bounty idea, *CNET News*, 21st of February, 2010, http://news.cnet.com/8301-13772_3-20034579-52.html

[11] P. D. Varcholik, J. J. LaViola, and C. Hughes. 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* (Sandbox '09), Stephen N. Spencer (Ed.). ACM, New York, NY, USA, 55-61, 2009.

[12] C. A. Wingrave and J. J. LaViola. Reflecting on the Design and Implementation Issues of Virtual Environments. *Presence: Teleoper. Virtual Environ.,* 19(2): 179–195, 2010.

[13] C. A. Wingrave, B. Williamson, P.D. Varcholik, J. Rose, A. Miller, E. Charbonneau, J. Bott, J. J. LaViola, The Wiimote and Beyond: Spatially Convenient Devices for 3D User Interfaces, *Computer Graphics and Applications, IEEE*, 30(2): 71-85, 2010.

[14] A. Zeitler, Survey and Review of Input Libraries, Frameworks, and Toolkits for Interactive Surfaces and Recommendations for the Squidy Interaction Library, University of Munich, 2009.

Acknowledgement: This work was supported by the Finnish Doctoral Program in User-Centered Information Technology (UCIT).