

# Teaching Mixed Reality Using Video Tutorials

**Gunver Majgaard**

**Embodied Systems for Robotics and Learning at University of Southern Denmark,  
Denmark**

[gum@mmmi.sdu.dk](mailto:gum@mmmi.sdu.dk)

**Abstract:** What are the learning potentials of using online video tutorials as educational tools in game programming of Mixed Reality (MR)? The paper reports on the first experiences of teaching third-semester engineering students design of MR using online step-by-step programming video tutorials. MR covers in this case both Augmented and Virtual Reality. Until recently, most of the instructional support for the software and game development came from paper tutorials. YouTube's rapid growth in popularity and easy to use programs for video production makes video tutorials a promising alternative to paper tutorials. It is often hard to find up-to-date thoroughly-worked-through textbooks on new and emerging topics such as MR. This motivated me to use existing video tutorials as teaching materials in the course titled Mixed Reality. The learning approach was inspired by the concepts of communities of practice and constructionist learning ideas. The imitating and copying of step-by-step programming video tutorials was a part of becoming a member of virtual community around game programming. In addition, the coding and problem-solving experiences were discussed in the classroom. The constructionist part was where the students made revisions and experiments by adding, combining and testing new coding elements to what they have already done in the tutorials. The students developed applications using the game engine Unity and tested the applications using android smartphones. Extending the applications based on the tutorials facilitated deeper learning for the students. For example, when they applied or recycled code components in new program contexts they developed a better and deeper understanding of the code. Surprisingly, comprehensive, polished and worked-through tutorials promoted minor changes in the developed applications. Whereas, less polished and less professional tutorials made the students become more creative and diverse in development of extended applications. The use of video tutorials in teaching of emerging technologies such as programming of MR prototypes is increasing. It is important to reflect on the learning potentials of both MR and video tutorials and to develop new appropriate teaching strategies to fit new types of learning materials.

**Keywords:** video tutorials, game programming, mixed reality (MR), learning, teaching

---

## 1. Introduction

Until recently, most of the instructional support for software and game development came from paper tutorials (van der Meij et al, 2016:332). YouTube's rapid growth in popularity and easy-to-use programs such as Camtasia Studio for video production makes video tutorials a promising alternative to paper tutorials. In addition, software and game engine companies such as Unity have already switched to video and other online materials as the primary medium for their tutorials. It is often hard to find up-to-date thoroughly-worked-through textbooks on new and emerging topics. Students tend to use video tutorials on their own initiative as supplementary tutorials for new and hard topics. This motivated me to use existing video tutorials as teaching materials in the course titled Mixed Reality (MR) for third-semester engineering students.

The MR course covered the fundamentals of game programming using the game engine Unity and the scripting language C#. The goal of the course was to develop applications for the android mobile platform for MR. MR applications included both Augmented and Virtual Reality games. Augmented Reality (AR) is a blend of the physical world seen through a camera and virtual content on top (Dunleavy, 2014). Virtual Reality (VR) is in this case applications seen through google cardboard.

The learning approach was inspired by concepts like communities of practice and constructionism (Lave & Wenger, 1991; Wenger, 1998; Papert, 1993; Majgaard, 2014; Majgaard, 2015). The imitating and copying of step-by-step programming video tutorials was a part of becoming a member of virtual community around game programming. Moreover, the coding and programming experiences were evaluated in community of the classroom. The constructionist part was where the students revised and extended the implemented tutorials. They experimented by adding, combining and testing new coding elements to what they have done in the tutorials. The learning goal was to make the students apply lessons learned in new contexts.

This raises the question, how video tutorials for game development of MR prototypes support the learning processes. Moreover, what are the key elements for successful use of video tutorials?

The questions were answered by evaluating the MR course where twelve third-semester engineering students primarily used video tutorials as education tools for learning to program MR prototypes.

Organisation of paper: First, we introduce related studies of using video tutorials in natural science and fundamental programming courses. These tutorials in the related work were primarily homemade. In this case study, the students used existing materials for programming emerging technologies e.g. developed by the manufacturers. Second, we introduce our educational case study by describing the idea of the MR course and the educational goals. Third, we present highlights from the execution of the course under two themes: 1) Fundamentals of game programming; 2) MR. Each theme ends with evaluations and reflections. The article ends with a summary and conclusion.

## **2. Related work**

Long et al (2016) used video lectures in an undergraduate course in “Water and Civilization” as part of a flipped classroom model. The flipped classroom is an instructional model in which the students are initially exposed to subject matter outside classroom through instructor-provided video lectures or other pre-class learning materials, and utilize classroom time for active learning processes, such as problem solving and group work” (Long et al, 2016: 46). The students found the videos motivating and valuable but according to Long et al (2016) they should not be more than 20 minutes.

Well et al (2012) developed video tutorials to improve outcomes of a basic programming course at university level. Programming is a difficult task to learn and requires a lot of practice in front of the computer. In addition, programming requires many skill sets besides the content matter such as problem solving and basic mathematics. Video tutorials were developed in Camtasia Studio (Techsmith Camtasia, 2016) which recorded screen images, mouse movements, keystrokes, and menu selections, as well as providing video audio of the person using the software. They created 12 tutorials each focused on the weekly content and consisted of the following steps: Presentation of the programming theme/problem; step-by-step coding; debugging and program testing. The duration was 10-15 minutes per video. The videos allowed the students to follow their own pace while watching and copying the code and other program manoeuvres. The videos were simple to create in the first place and extra videos were created to solve specific programming issues during the semester.

Van der Mei et al (2016) investigate effects of video tutorial for software training such as improved user motivation, retention and task performance. In software training, the emphasis lies in getting to know the software interface and learning the action-reaction pattern. Van der Mei et al (2016) describe programming skills as procedural knowledge in action. The knowledge in action is a series of coding steps, which leads to the intended program behaviour. In a tutorial, the learners learn programming skills by copying the coding steps. In addition, they get a feeling of reward when the program reacts as expected. Regarding structure of a successful tutorial Mei et al (2016) proposed a summary of the results in the beginning of the video tutorial and recap in the end for improving the retention of the new-learned skills. Retention in this case means the ability to remember the learned skills. In our case study, we additionally aim for the students to use the learned skill in other contexts.

## **3. Method and case study**

The method in this case study is inspired of Action Research and Design-based Research. Experiments and critical reflections are the core of Action Research, allowing learning from and through practice (Lewin, 1946). Lewin describes the Action Research process as being like a spiral staircase in which each cycle is composed of: planning, action, and fact-finding about the result of the action (Lewin, 1946:38). Design-based research is an approach with the intent of producing new theories, artefacts, and practices that account for and potentially impact learning and teaching in naturalistic settings. To understand and support learning processes, the research involves the development of technological tools, curriculum, and especially theory (Barab & Squire, 2004; Majgaard et al, 2011). In the case study presented in this paper, we developed new course curriculum, which we conducted and evaluated in two rounds. The first round covered the theme of game programming in Unity and the second round covered MR. In both rounds, the used learning material were mostly video tutorials. The evaluations are reflections over the student products and student feedback associated with theoretical concepts. The link between concepts and experience provides a language for a better understanding of the learning potentials and pitfalls of using video tutorials in programming.

The feedback from the students were collected in several rounds. We had short informal feedback rounds when the students presented their productions: They were asked to explain how their program worked including some coding elements; How they extended the tutorial; What part of the coding process was most difficult and where they spend most time on problem-solving. They were also asked to comment on their experience with the given tutorial. The final day of the course, the students were asked to reflect on each of the tutorials and project work in retrospective. They were asked about the length, pace and quality of the tutorials. They were to explain why they liked or disliked a given tutorial. Finally, they were asked to reflect on how we could improve the course and the coursework.

#### **4. About the course mixed reality (MR)**

The course title was MR and the goal was to teach 3<sup>rd</sup>-semester engineering students fundamental Unity Programming and development of VR and AR applications for a mobile android platform. The students have had programming courses in both 1st and 2nd semester so they knew fundamentals of variables, data structures, object orientation, methods and so on. They had previously developed their own extended version of the beginner tutorial Roll-a-ball in Unity (Unity Roll-a-ball, 2016). In order to make Virtual and AR work on an android platform, third-party software was used such as Vuforia (2016) for the image-based AR support. This added complexity to the installation and adjustments of settings in Unity. In image-based AR, a virtual content is enabled when the camera detects a specific image in the real world.

#### **5. About using video tutorials**

Both the development of MR applications and the use of video tutorials as educational tools were new themes in teaching programming (van der Mei et al, 2016). In addition, it was very difficult to find good up-to-date paper tutorials for MR. Moreover, the chosen programming platform, Unity, was changing and developing new features more than once a year.

In this case study, we used online video tutorials from Unity's web page for fundamentals of game programming and tutorials found on YouTube for MR.

Requirements for suitable video tutorials summarised based on own experiences and the related work (Well et al, 2012; Long et al, 2016; and Mei et al, 2016):

- Content goals should fit curriculum;
- Recap and programming goals in the beginning of the video;
- Step-by-step programming;
- Testing and debugging;
- Recap in the end of the video;
- Short – no longer than 30 minutes;
- Moderate in pace;
- Good and precise in language;
- Zooming in on important steps;
- Not skipping important steps;
- Up to date with current version of Unity;
- Additional written materials of scripts and environment settings make re-finding of difficult elements easier.

Most of the used tutorials satisfied these requirements.

#### **6. Preparing for teaching**

The lesson plan consisted of twelve four-hour lessons taught over twelve weeks and divided into two phases:

- Fundamentals of game programming in Unity for three weeks: The students implemented 10 Fundamental C# scripts in Unity (Unity Scripting, 2016); a 2D-side-scroller (Unity Basic Platformer Game, 2016) and a 3D two-player (one keyboard) shooter game (Unity Tank, 2016).

- MR the last nine weeks: The students were introduced to VR by exploring existing applications and design guidelines for reducing cyber sickness. Then they implemented a Mars tutorial (Rabidgremlin VR, 2016) and a VR Character Controller (from Unity's asset store).

Finally, in the last three weeks they implemented an AR maze and their own AR application (WireBeings AR, 2016).

The students were asked to implement the above-mentioned tutorials and additionally they were required to make their own extended versions. The reason for this was that the changes and additions to the code would force them to better understand the application they had just developed.

Organisation of the lessons: The students were to complete the tutorials at home as preparation to the lessons. During the lessons, they were to present the programming results and explain the code in smaller groups. Bug fixing and problem solving was done in class as well. In the end of each lesson, they were introduced to next week's tutorial.

Students' active participation was part of the constructionist learning approach and key for learning to program (Papert, 1993; Majgaard, 2015). Plans for making the students active:

- They were to do oral presentations and reflections of the produced code in the classroom mostly in small groups of 3-4 students;
- The students should make extended and revised versions of the tutorials in order to understand and master the new code elements;
- The students should combine tutorials in order to become creative problem solvers and to apply "old" knowledge in new contexts.

These didactical ideas should improve the students' ability to become creative and knowledgeable programmers of MR.

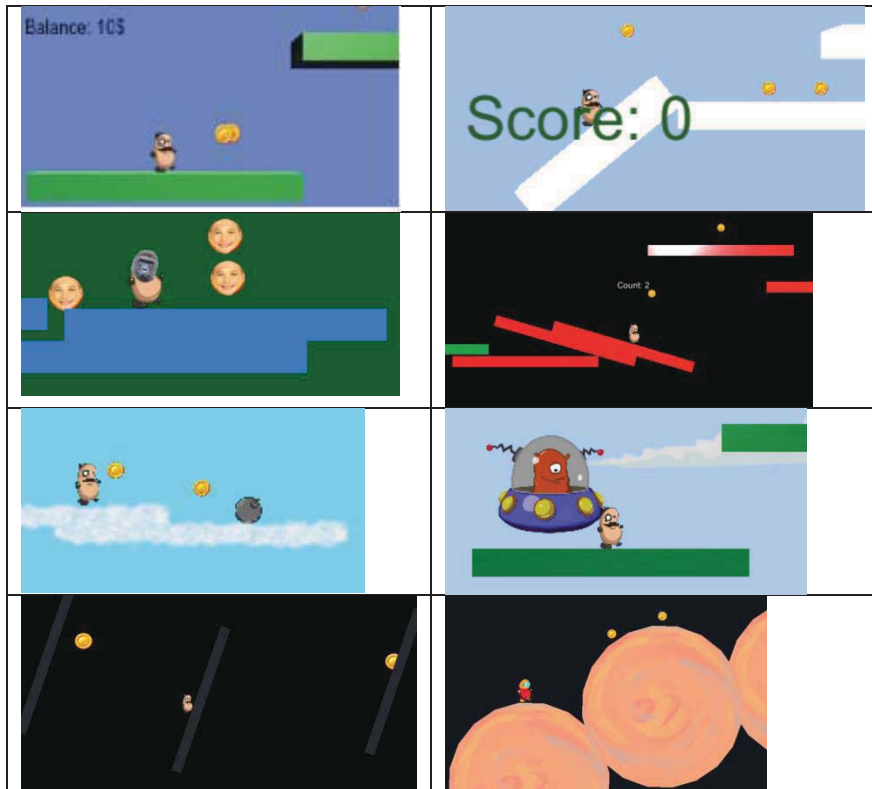
## **7. Highlights from phase 1: Fundamentals of programming games**

In the first three weeks, the students worked their way through fundamental game programming in Unity. Initially, they implemented C# scripts about getting input from keyboard, if statements, loops, update system etc. (Unity Scripting, 2016). Then they implemented a simple side-scrolling game in their own extended version, see the figure below (Unity 2D Platformer Game, 2016). A side-scrolling game is a 2D game viewed from a side view angle, and the character moves from the left side of the screen to the right (Fullerton, 2008).

The tutorial described the step-by-step programming of the side-scroller using assets for the hero-character and the collectable coins from Unity's free asset store. The students were to add points e.g. inspired by the roll-a-ball tutorial (Unity Roll-a-ball, 2016). The side-scroller should restart automatically when the hero-character fell of a ground element. The tutorial didn't support an up-to-date description of the restart function, so a different tutorial was used for that purpose. Most students added points (Table 1. first row). One student changed the sprites of the hero-character and coin-element (Table 1. second row). Several students also added background music. Two students also added a bomb and if the hero touched the bomb the game restarted (Table 1. an example third row on the left). One of the students also integrated a spaceship, which automatically followed the hero-character and upon contact the game restarted (Table 1. third row to the right). Three students made the ground rotate in various ways (Table 1. examples in the fourth row).

The third assignment was a hot-seat multiplayer game and the tutorial from start to end took over four hours. The tutorial was chosen because it was thorough and had supporting written material. The goals covered most aspects of third-person shooter including player control, health, camera zoom and audio. The figure above shows the game, Tank, in a student-extended version with an additional cross where the tanks could regain health points. Since the tutorial took over four hours, the students were not initially required to extend the tutorial but some of them extended the tutorial anyway e.g. by adding a health-spawn point where players could regain their health or change the way the tanks were shooting.

**Table 1:** Six extended versions of side-scroller.



**Figure 1:** Tank hot-seat multiplayer extended version with health-spawn point.

### 7.1 Reflections on deep learning by adjusting and extending the program

In their first semester, the students had already implemented a tutorial containing a scoring system (Unity Roll-a-ball, 2016). The students were told to use their knowledge about the scoring system in a new context in the side-scroller. In order to add a scoring system to the side-scroller, the students had to understand where and how to declare the score variable and where to increment it. Additionally, they had to know how to show the score on the canvas. The process of using existing knowledge in a new context constitutes deep way of learning. As a teacher, you always hope the students are able to use their newly learned skills and knowledge in other situations. In learning theory, the process of using existing knowledge in a new context is referred to as second-order learning (Bateson, 2000). When the students implemented the scoring system in roll-a-ball for the first time everything about programming games in Unity was new to them and the process of implementing the scoring system for the first time was a more rudimentary type of learning. In learning theory the process of doing something new e.g. programming the first simple game in Unity constitutes first-order learning according to Bateson (2000). As a teacher, I knew they got the scoring system to work but I didn't know if they actually understood what they were doing. After evaluating their extended side-scroller code and discussing their solutions in class I felt confident that they had got a deeper and better understanding of both the scoring system

and the code from the side-scroller. Furthermore, they had to locate where to implement the scoring system in the side-scroller. In order to do this, they had to understand important parts of the side-scroller code.

## **7.2 Reflections on how more specific requirements for the extended versions might improve creativity**

The students' feedback on the side-scroller circled around more specific and detailed requirements to the extended version e.g. objects falling from the sky, levels, bombs or UFO-AI. Some of the students already implemented some of these elements. They didn't quite know when they met the expected requirement for the extended version and what we expected of them.

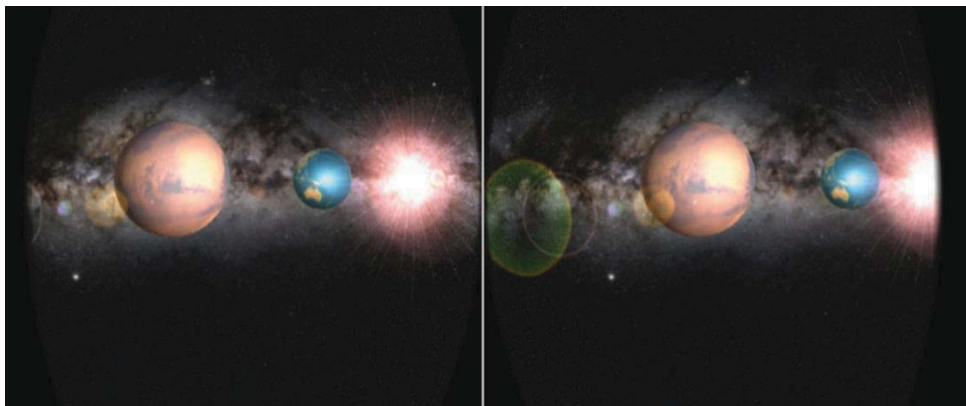
In the 1990s, a group of Danish film directors wrote up ten rules to reinvent their film making (Danmarkshistorien Dogme 95, 2016). These rules or specific requirements forced the directors to become creative within a narrow frame, e.g. hand-held cameras and shooting on location. In this perspective, more specific requirements for extending the games might improve the students' creativity within a particular frame. In the next iteration of the course, we will introduce requirements that are more specific as a result of the feedback received from our students and the inspiration gained from the Dogme-95 group.

## **7.3 Reflections on fundamentals of programming games: Length and pace of tutorials**

The side-scroller tutorial was about thirty minutes long and the Tank tutorial was about four and a half hour. The Tank tutorial was divided into eight video lengths of between ten minutes and one hour. Four and a half hour is a long time to follow step-by-step coding. It would take even longer to implement if the students had to revisit parts of the tutorial as part of problem solving. On the other hand, they might also skip some part of the oral coding explanations. Contrary to expectations, the students didn't mind the length of the Tank tutorial because it covered a lot of core-game-programming concepts in Unity such as character control in 3D, cameras, health, audio, prefabs, and assets. The students said for example: "I got much out of the tutorial" and "I didn't need to pause the video". The students often mentioned pace when they evaluated the tutorials and they often found them too fast. In the Tank tutorial most of the students liked the pace. The video was shot in front of a live audience who simultaneously followed and implemented each step of the instructions – this may have encouraged the instructors to keep a steady pace. The Tank tutorial also provided additional material such as power point introducing each theme and step-by-step settings in the game engine environment supporting the video. All in all our students really appreciated this thorough tutorial. Especially the AR tutorial was too fast in pace and criticised by most students even though they got the AR technicalities to work, see the next section.

## **8. Phase 2 mixed reality (MR)**

The next nine weeks the students worked with MR. First, various VR platforms were discussed and the students tried out existing VR apps for Google Cardboard. Google Cardboard was chosen as a platform because it was mobile and less expensive than e.g. Oculus Rift. The students then implemented their first VR application for google Cardboard, the figure below shows an example of an extended version of their first VR tutorial.



**Figure 2:** Student project: Part of the solar system, the globes circled around themselves and the sun, and the moon circles around the earth

The first VR tutorial was the development of a Mars simulation in VR for Google Cardboard. The Mars tutorial took about 30 minutes. In order to develop a Google-Cardboard-VR application, the students needed to download and import the software development kit Google VR for Unity. The package provided the stereoscopic view for left and right eye, see the figure above. In order to run the application from a mobile phone, the students needed to switch platform and build the application directly on an android mobile wired to the computer or a Dropbox virtually linked to the android mobile. The students extended the tutorial with other globes from the solar system e.g. Earth and the Moon spinning around the Earth. This was possible without additional coding in C#; they just needed to duplicate the Mars object and change attributes such as fabric texture, size, and which globe to circle around. This made the students explore the parent/child relation between objects. The students found the export of the program to an android a bit difficult and it wasn't covered in detail in the tutorial.

The second part covered a development based on a customised walker in a maze asset and since we didn't find a good tutorial we introduced it ourselves. An asset is an extension to the game engine, in this case a first person walker who walks around in a maze. The walker turns left or right dependent of the player's head position and the walker stops when the player looks down.



**Figure 3:** A student assignment based on the Walker-asset. On the left is the first-person perspective of the walker. On the right is an overview of the maze with many trees

The last three weeks the theme was AR and we found a short tutorial, which introduced image-based AR. See the figure below. The tutorial was about 30 minutes long and covered the development of an AR maze. The students were to develop their own extended version or own idea.



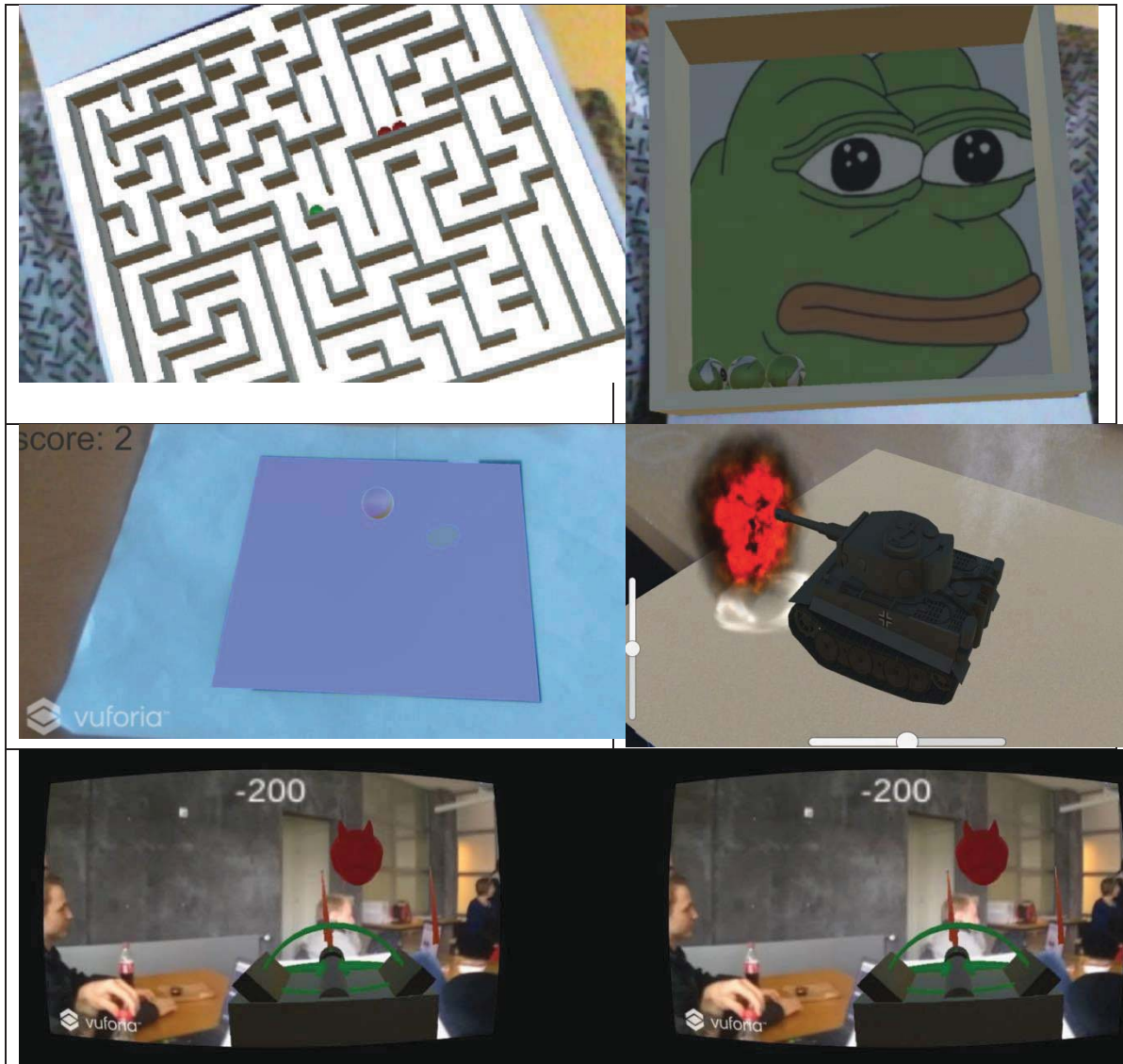
**Figure 4:** Image-based AR. The black and white image enables the virtual 3D Earth

The virtual content is enabled when the camera in the computer or mobile recognises a specific image, see the figure above. The extension Vuforia provides a computer vision service to recognise self-chosen images (Vuforia, 2016). In the figure above an image of a black and white maze enables the virtual content – the Earth. Below is examples of the students' own AR applications, see table 2 below.

The maze in the top-left corner of table 2 is similar to that of the tutorial – except from a transparent lid to keep the balls in the maze and two extra red balls. The player moves the paper and/or camera in order for the green ball to escape maze without colliding with the red balls. The box in the top-right corner of table 2 shows three

balls rolling dependent of how the player moves the camera and/or paper. In the middle to the left of table 2, a ball bounces on a piece of paper and the player uses the paper as a table tennis bat. To the right, a tank with a rotating gun barrel is shooting. At the bottom is a combined VR and AR shooter. The game is initiated by a Vuforia image and the shooter earns points by aiming at the brown trolls.

**Table 2:** Screen dumps of the students' AR prototypes



**8.1 Reflection on VR and AR: Diverse student solutions based on the less elaborated tutorials**

For both VR and AR it was hard to find good, thorough and up-to-date tutorials. The Mars tutorial was simple and introduced well to VR. The AR tutorial was up to date but the pace was too fast (according to the students). Additionally, setting up the maze required a lot of micro-management.

The students managed to improve the maze by putting an invisible lit on top in order for the balls to stay in the maze.

The students' AR prototypes were very diverse – but all assignments covered the basics of using the camera and enabling the virtual content using a Vuforia image. The tutorial and the requirement for the delivery didn't tie the students to develop uniform prototypes. Moreover, they were encouraged to develop their own versions and most students succeeded. The last assignment showed that some students used a new and broader range of programming elements, see figure 6. The diverse solutions also revealed that the students mastered the technology. They did more than copying code – they used the code to support their own ideas.



Astonishing, the less professional AR tutorial set the students free to become more creative. Perhaps, it was more difficult to become creative when using the more polished and elaborated tutorials such as the Tank tutorial.

## **9. Summery and conclusion**

A challenge of using step-by-step programming video tutorials in software development education is to make the students (re)use essential pieces of code in new contexts and understand what they just implemented.

Extending the tutorials promoted deeper learning for the students. The (re)use of coding elements in new contexts facilitated the students' understanding of the coding elements in both old and new contexts. Applying the newly learned knowledge in the form of coding elements in new contexts was an example of second order learning. Second order learning is a deep and lasting kind of learning (Bateson, 2000). Moreover, second order learning provided the students with the skill of actively applying newly learned knowledge. The repetitive part also improved the retention.

Based on the case study we propose the following student activities to improve the students' retention, learning, and understanding:

- Use key concepts from previous tutorials in new programming projects to promote second order learning
- Extend the code from the tutorial programs e.g. by adding points, objects or new behaviour
- Explicit requirement for minimum extensions of the tutorial programs e.g. propose the students to implement 4 out of 10 requirements
- Explain and discuss the code in class
- Problem solving in class

The students requested a minimum number of explicit requirements for extensions of applications, which will be easy to implement next year (also based on the students' productions from this year).

It was easy find good and professional tutorials for game programming in general. For MR it was a somewhat hard task to find professional and thorough tutorials. Astonishingly, these less professional videos made students use the tutorials more as a lever for developing their own ideas. Additionally, some students fixed the bugs in the tutorial and made an improved version.

According to experiences from the course, student feedback and related work (Well et al, 2012; Long et al, 2016; Mei et al, 2016) a successful and valuable step-by-step video tutorial is to be measured by the following elements:

- Content goals should fit curriculum
- Clearly structured video beginning with a summery
- Step-by-step programming, test and debugging
- Recap of lessons learned in the end
- Thorough explanations
- Moderate pace
- Not too long. If the video is too long it should be divided into chapters as in the Tank tutorial
- Up to date (Unity makes several changes each year)
- Good and precise in language
- Zooming in on important steps
- Not skipping important steps
- Additional written materials of links to third-party software, scripts and environment settings make re-finding of difficult elements easier.

Surprisingly, the students didn't mind the length of the video if the other tutorial elements were well covered.

## References

- Bateson, Gregory, (2000 (1972)) *Steps to an Ecology of Mind: Collected Essays in Anthropology, Psychiatry, Evolution, and Epistemology*, Chicago Press, ISBN 0-226-03906-4
- Danmarkshistorien Dogme 95 (2016) Dogma 94, <http://danmarkshistorien.dk/leksikon-og-kilder/vis/materiale/dogmereglerne-dogme-95-1995/> retrieved 14-12-2016
- Dunleavy, M. (2014) "Design principles for augmented reality learning." *TechTrends*, 58(1), pp. 28-34
- Fullerton, T. (2008) *Game Design Workshop*, Morgan Kaufmann.
- Lave, J., & Wenger, E. (1991) *Situated Learning: Legitimate Peripheral Participation*, Cambridge: Cambridge University Press, ISBN 978-0-521-42374-8.
- Long T., Logan J., Cummins J., Waugh M. (2016) "Students' and Instructor's Attitudes and Receptions of the Viability of Using a Flipped Classroom Instructional Model in a Technology-Enabled Active Learning (TEAL) Classroom: A Preliminary Study", *Journal of Teaching and Learning with Technology*, Vol. 5, No. 1, July 2016, pp. 46-58. doi: 10.14434/jotlt.v5n1.18879
- Majgaard, G. (2015) "Humanoid Robots in the Classroom", *IADIS International Journal on WWW/Internet*, vol 13, nr. 1, s. 72-86.
- Majgaard, G. (2014) "Teaching Design of Emerging Embodied Technologies", In P. Isaias, & K. Blashki (red.), *Human-Computer Interfaces and Interactivity: Emergent Research and Applications*, pp. 188-206. IGI global. DOI: 10.4018/978-1-4666-6228-5.ch011
- Papert, S. (1993) *Mindstorm – Children, Computers, and Powerful Ideas*, Basic Books
- Rabidgremlin (2016) Creating a Google Cardboard VR app in Unity. <https://www.youtube.com/watch?v=AFGOhZJwoFw> last retrieved 01-12-2016
- Techsmith Camtasia (2016) <http://discover.techsmith.com/camtasia-brand-desktop/?gclid=Cj0KEQIA9ZXBBRC29cPdu7yuvrQBEiQAhyQZ9P1bLLyLLuoUFDxe3-ckDr4Udxvdw27BHAFukYcKCHMaAnX28P8HAQ> last retrieved 01-12-2016
- Unity Scripting (2016) <https://unity3d.com/learn/tutorials/topics/scripting> retrieved 01-12-2016
- Unity 2D Platformer Game (2016) <https://unity3d.com/learn/tutorials/topics/2d-game-creation/creating-basic-platformer-game?playlist=17219> last retrieved 01-12-2016
- Unity Tank (2016). <https://unity3d.com/learn/tutorials/projects/tanks-tutorial> last retrieved 01-12-2016
- Unity Roll-a-ball (2016) <https://unity3d.com/learn/tutorials/projects/roll-ball-tutorial> last retrieved 01-12-2016
- van der Meij, H. & van der Meij, J. (2016) "The effects of reviews in video tutorials", *Journal of Computer Assisted Learning* (2016), 32, 332–344. John Wiley & Sons Ltd.
- Wells J., Barry R. M., and Spence A., (2012) "Using Video Tutorials as a Carrot-and-Stick Approach to Learning" *IEEE Transactions on Education*, Vol. 55, No. 4, November 2012 453
- Wenger, Etienne (1998) *Communities of Practice: Learning, Meaning, and Identity*, Cambridge University Press. ISBN 978-0-521-66363-2.
- Wiki Side-scroller (2016) [https://en.wikipedia.org/wiki/Side-scrolling\\_video\\_game](https://en.wikipedia.org/wiki/Side-scrolling_video_game) last retrieved 01-12-2016
- WireBeings (2016) How To: Augmented Reality App Tutorial for Beginners with Vuforia and Unity 3D. <https://www.youtube.com/watch?v=uXNjNcgW4kY> last retrieved 01-12-2016
- Vuforia (2016) <https://library.vuforia.com/articles/Solution/Installing-the-Unity-Extension>