Gaming Equipment for the Enrichment of Computer Science Education

Heinz-Josef Eikerling[1], Michael Uelschen[2]

Faculty of Engineering and Computer Science
University of Applied Sciences Osnabrück
Osnabrück, Germany
[1] h.eikerling@hs-osnabrueck.de, [2] m.uelschen@hs-osnabrueck.de

Abstract

Within several courses we have examined the use of gaming equipment to enrich computer science courses, as gaming was, currently is and is supposed to still be a catalyst for computer engineering and software development. In this paper we particularly address the use of equipment in educational units taking Microsoft's Kinect as an example. We sketch different course layouts and report experiences made with respect to those. Finally, we will draw some conclusions on the likely evolution of this approach and give recommendations concerning the adaptation of this concept to instructors.

Keywords: Gaming, education, computer science, programming, user interfaces, gesture recognition

1. Background & Motivation

High motivation is an extremely good prerequisite for sustainable learning success. Thus means to increase students' motivation should be continuously checked for and examined for their applicability especially with respect to technical disciplines. In recent years gaming equipment technology-wise progressed significantly. For instance General Purpose Computing on Graphics Processing Units (GPGPU) and the use of game engines to implement sophisticated augmented reality applications used in a business context have recently gained interest.

Within the last 2 years we have defined various course layouts targeting different types of students which make use of advanced gaming equipment. Through this we intend to motivate students to explore the functions and engineering aspects of the involved gadgets more deeply. Moreover, the students should apply creative solution finding with respect to programming assignments they have to deal with as part of the computer science syllabus.

According to our experiences the Kinect platform issued by Microsoft some years ago [5] is quite well suited for this purpose. The use of such device within a system comprising hardware and software applications in order to do sensor data processing is highly appreciated by the students. We noticed that students easily take up with such technology though the use of it requires some preparation and the integration isn't always done quite easily. Next, we will briefly review the primary technology, afterwards describe some educational units held in which this technology was key. Finally we give some suggestions for setting up own learning units based on our experiences made.

2. Gaming Device: Microsoft's Kinect

During the last few years the interaction between the player and the gaming device (i.e., game console) changed. Instead of using a simple stationary controller the interaction now involves the full body of the player. Basically two approaches are prevalent in the market: (i) wire-less controllers with integrated sensors (e.g. Nintendo's Wii), and (ii) contact-free body-tracking systems (e.g. Microsoft's Kinect).

The Kinect-system consists of a multi-array microphone, a RGB-camera with VGA resolution (640x480), and an infrared-based depth sensor. The combination of camera picture and depth information facilitates to identify and to track body parts. Up to 6 individuals can be recognized by the Kinect system. The algorithms were developed by Microsoft as part of the *Natal* project. Details on the algorithm that identifies body parts only on a single frame can be found in a seminal paper [6].

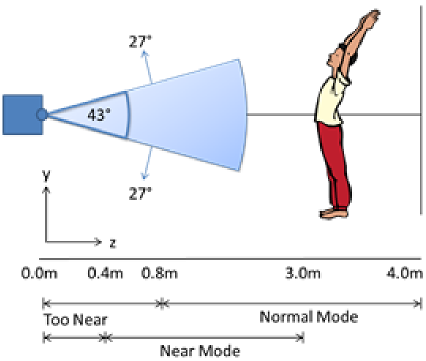

Fig. 1: Dimensions of Kinect's sensed area

Along with the hardware comes a software development kit (SDK) by Microsoft. Aside from games, the SDK permits to develop serious applications making use of tracked data. It offers a high-level programming interface which simplifies further processing of the identified body parts, like joints, arms, and legs.

Figure 1 shows some technical details of Kinect's depth sensor. The system can be configured in two modes. The normal mode enables the identification of persons in a range between 0.8 and 4.0 meters in front of the sensor. The near mode has a reduced range but it has the capability to identify hand gestures or other body parts up to 0.4m. The Kinect system is equipped with a motor in order to adjust the view of sight up to 27° in two directions. The angle of view is 43° for the vertical plane (y-z) and 57° for the horizontal plane (x-y).

3. Usage in Computer Science Syllabus

Computer science curricula exist in various editions. Due to aspects of globalization and the widespread proliferation of standards in the academic and the business world during the last decade educational programs became more harmonized. At our institution, the University of Applied Sciences Osnabrück, we offer bachelor (6 semesters / 3 years) and master (4 semesters / 2 years) programs in computer science. The general agenda for the bachelor degree is guided by the goals

- to convey a general, extensible knowledge of the subject which can be applied in a professional setting,
- to enable students to solve problems in the domain,
- thus skilling students with respect to core analytic competences and formal methods,
- to teach and practice the required communication and social skills.

These goals are rather in line with the commonly expected capabilities of computer science graduates described in the ACM / IEEE Computer Science Curriculum 2013 [3]. According to the 18 knowledge areas (KAs) described in [3], the above goals are associated with the KA 'Social and Professional Issues (SP)'.

In summary, we can address at least 50% of the 18 KAs (e.g., software engineering, hci, platform-based development, programming languages,…) listed in [3] by employing the Kinect in our educational units. We will now describe, how this done.

4. Courses

4.1. One week undergraduate course

Within the faculty of electrical engineering and computer science regularly a one-week course is offered to 3rd semester students. The course is organized as a project, similar to course layouts described in [1]. According topics are offered by the advisors and address a variety of topics. The project work includes a presentation of the results which in turn is reviewed by a jury composed of faculty staff members and students. Under this label we conducted two variations of the project in which the Kinect was featured.

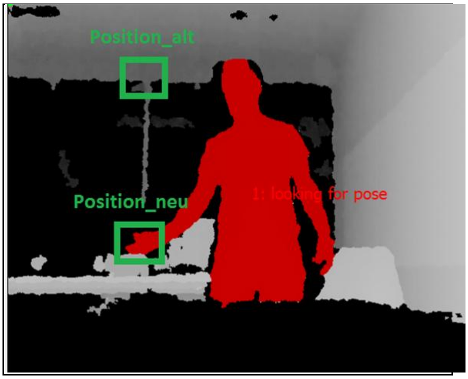

Fig. 1: Recognition of dynamic gestures.

Both courses were offered to students of computer science, electrical / mechanical engineering and mechatronics. Additional to technical competences our intention was also to skill the students regarding non-technical competences (i.e., soft skills). Though the available time frame was rather limited, during the second edition of the course we were able to address more complex topics like the algorithmic aspects of dynamic gesture recognition (see Figure 1). In general, tasks (UI design, driver implementation, system integration) resulting from the assignment (design and implementation of a Kinect controllable information display) are distributed among the members of a team comprising 5 – 6 students. Since we did not assume any experiences or prior knowledge in regard to the Kinect platform, the students received a full-fledged version of the IDE including a sample project.

4.2. Engineering project ‘Carolo Cup’

The design of this course was inspired by a national student initiative called ‘Carolo Cup’. The challenge is to drive a model car autonomously through a predefined road network mockup. In former revisions of the course, the vehicle including the required control software was build.

Fig. 2: The Kinect’s depth sensor is used to identify objects while driving.

Within this revision we followed a new, innovative approach on utilization of Microsoft’s Kinect. Instead of having the Kinect immovable on a desk, and a person acting in front of the system, we used the Kinect in a mobile environment. The Kinect is part of a movable car and observes the static surroundings (see Figure 2).

The project is accompanied by a lecture on software engineering for final year undergraduate students of computer science. This project shall be conducted within a semester timeframe by a group of 4-8 students.

The project addresses the software engineering and development knowledge areas mainly. Occurring technical and design issues related to the platform had to be resolved independently through the competences of the involved students. During the execution of this project it turned out that the performance requirements of an initial revision of the object recognition did not match the capabilities of the Kinect System. Hence, optimization with respect to the concept and the implementation had to be done.

4.3. Targeted challenge project

This project type is intended to last for 2 months. The objective of a concrete project was to control an autonomous robot through gestures. This means that a movement of a person should be copied by the robot. For instance, if the controlling person raises his left arm, the robot should also raise its left arm. The assignment was actually handled by an exchange student.

Fig. 3: The robot follows the movements of a user tracked by the Kinect sensor

The environment is rather complex and features 2 distinct systems: (i) for the interaction using the Kinect system, and (ii) a system for supervision of the humanoid. We are using the NAO robot from *Aldebaran Robotics* which can be programmed in a mixture of C++, python, and a graphical language. This project is intended to train the following competences:

- Project and time planning based on the use-cases and scenarios given to the student at project start
- Software architecture of a distributed (client / server) system, where the Kinect-system is the client and the robot acts as server
- Definition of an appropriate interface between both components
- System and network programming using sockets as communication channel

Figure 3 sketches the application scenario. As a result, the student was able to control the humanoid by movements of the upper parts of his body. Since the movements of a human are usually much faster than the actors (i.e., motors) attached to the various joints of the robot we faced the problem that the motion of the robot appeared delayed.

5. Findings & Recommendations

5.1. Coverage of Knowledge Areas

As can be seen, by featuring the Kinect technology in the according courses an extensive coverage of the computer science knowledge areas can be achieved. A straight-forward usage of the Kinect technology is in the Human-Computer Interaction area. Software engineering and software development fundamentals are less specific to Kinect projects and courses; instead the arising challenges are a result of letting students work in teams.

This partly applies to the social and communication skills required in such projects and courses. Demonstrators being developed and exhibited by students may easily attract spectators if this was purposefully planned for. Distributed and net-centric computing is not an inherent demand for Kinect projects and courses, but a result of our projects and course layouts. As mentioned before, this is a frequently occurring architectural style in sensor-based systems and thus should be featured in respective student projects.

5.2. Team vs. Single Student Project

Our Kinect projects indicate that the technology is suited for team and individual student projects. Applications featuring Kinect inherently address various required competences relevant to computer science students which can be quite well handled by student teams. The steep learning curve helps to achieve quick successes and hence maintain team sanity.

Individual projects should be rather focused. The Kinect platform gives ample space for developing concepts with respect to certain issues in sensor based applications. Pertaining to the project described in 4.3 machine learning and speculative execution to predict gestures could be explored in order to reduce the inevitable effect of latency.

6. Limitations and Outlook

Two areas turned out to be difficult to address in educational units featuring Kinect: algorithms and platform-based design. The latter to our mind requires a more extended time frame and a higher skill level concerning software engineering, programming and software design fundamentals.

Concerning algorithmic aspects there is a certain trade-off: on the one hand, a variety of rather different algorithms are known [2], on the other hand they are rather specific, and typically not addressed in the basic algorithm and data structure courses. Here some space to optimize the curriculum coverage of Kinect-based courses and projects exists.

7. References

[1] H. Li. "*Education in Electrical Engineering through A Design Project*", Canadian Conference on Electrical and Computer Engineering, 2009. CCECE '09, pp.342-345, 3-6 May 2009.

[2] S. Mitra, T. Acharya. "*Gesture Recognition: A Survey*", IEEE Transactions on systems, Man and Cybernetics, Part C: Applications and reviews, Vol. 37, No. 3, pp 311- 324.

[3] M. Sahami, S. Roach, E. Cuadros-Vargas, and D. Reed, "*Computer science curriculum 2013: reviewing the strawman report from the ACM/IEEE-CS task force*", Proceedings of the 43rd ACM technical symposium on Computer Science Education, ser. SIGCSE '12, New York, NY, USA: ACM, 2012, pp. 3–4.

[4] A. Tucker (ed.). "*A model curriculum for K-12 computer science. In: final report of the ACM K-12 task force curriculum committee*", ACM, New York (2003).

[5] Jarrett Webb, James Ashley. "*Beginning Kinect Programming with the Microsoft Kinect SDK*", Apress, 2012.

[6] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, A. Blake. "*A Real-Time Human Pose Recognition in Parts from a Single Depth Image*". 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1297 – 1304.