Problem-Based Learning

in Software Engineering

586297-EPP-1-2017-1-EL-EPPKA2-CBHE-JP

Co-funded by the Erasmus+ Programme of the European Union

Document Info	
Project reference	586297-EPP-1-2017-1-EL-EPPKA2-CBHE-JP
Deliverable / Task	T4.1
Dissemination level	Public
Date	13.04.2021
Document version	1.0
Status	Final
Authors	Raja Jamillah Raja Yusof, University of Malaya Aishah Abu Bakar, University of Malaya Hazleen Haris, Universiti Tenaga Nasional Zailani Ibrahim, National University of Future and Emerging Sciences Irum Inayat, National University of Future and Emerging Sciences Shree Raj Shakya, Tribhuvan University
Reviewer	Raja Jamillah Raja Yusof, University of Malaya
Approved by	Steering Committee

Co-funded by the Erasmus+ Programme of the European Union

Contributors

Dhiraj Shrestha, Kathmandu University

Kamran Khowaja, ISRA University

586297-EPP-1-2017-1-EL-EPPKA2-CBHE-JP

Co-funded by the Erasmus+ Programme of the European Union

Contents

Co-funded by the Erasmus+ Programme of the European Union

1. Introduction

Active learning is a process in which students participate in activities that encourage higher-order learning skills, including analysis, synthesis, and evaluation. According to Bonwell and Eison, this is the definition of strategies that stimulate active learning: "instructional activities involving students in doing things and thinking about what they are doing" (Bonwell and Eison, 1991). These strategies allow students to engage in the learning activities and at the same time construct their own knowledge and understanding – making meaning in what they do in the learning activities. Active learning approaches focus on developing students' skills rather than transmitting information. Active learning strategy enables students to solve real-life problems; identify suitable resources for problem solving; use effective self-directed and self-motivated learning skills; continuous monitoring and evaluation of the sufficiency of their own knowledge and problem-solving skills; teamwork, which helps students develop communication and leadership skills, social and ethical skills.

There is a wide range of active learning strategies: it can be as simple as pausing during a lecture so that students have the opportunity to clarify and organize their ideas; or it can be as complex as using real cases in the real-world environment. In active learning, learners are always engaged in activities that involve the use of their higher-order thinking in groups.

There are many active learning strategies available. They include Problem-based Learning, Case-based Learning, Challenge-based Learning, Studio-based Learning, Experiential Learning, Gamification and Simulation and Scenario/Story-based Learning. Challenge-based Learning (CBL) is where real world issues are posed as challenges broadly. This will allow for a variety of solutions for students working in teams to tackle. CBL emphasizes on exploring topics and solving the challenge from multiple perspectives. This will allow students to appreciate the connections between these subject areas. Reflection, documentation and

586297-EPP-1-2017-1-EL-EPPKA2-CBHE-JP

formative assessment are parts of the learning process. In contrast to CBL, Game-based Learning Gamification (GBL) and simulation approaches are meant to motivate students to learn by using video game design and game elements in learning environments. The goal is to maximize enjoyment and engagement through capturing the interest of learners and inspiring them to continue learning. Games and simulations enable students to solve real-world problems in a safe environment and enjoy themselves while doing so. This pedagogical approach is also seen as a means to train students to develop perseverance attributes via 'fail often and early' in a safe environment. Scenario/Story-based Learning (SBL) involves students working through a storyline based around an ill-structured or complex problem (where the scenario changes over time) which they are to solve. Students have to apply subject knowledge, critical thinking and problem solving skills. Decisions made will affect and/or alter subsequent event leading to new events like in real life. Mistakes are part of students learning to prompt them to make better choices in the future. In studio-based learning students work like apprentices in a common space under the tutelage of a mentor. Students undergo periodic critiques (crits) of their designs, projects or products. Crits are done to gain knowledge about their work and involve student-to-mentor first before evolving into self-learning crits between peers. Final works or products can be presented publicly.

All active learning strategies employed by lecturers need to be supported with appropriate teaching methods to ensure that the strategies are effective. Amongst teaching methods available for lecturers to consider are Flipped Classroom, Cooperative Team Learning, Jigsaw Teaching, Fishbowl Class Discussion, Think Pair Share, Cooperative Note Taking, Debate, Gallery walk and many others.

The integration of teaching methods and active learning strategies in designing meaningful active learning curriculum is shown in the Figure 1 below.

Co-funded by the Erasmus+ Programme of the European Union

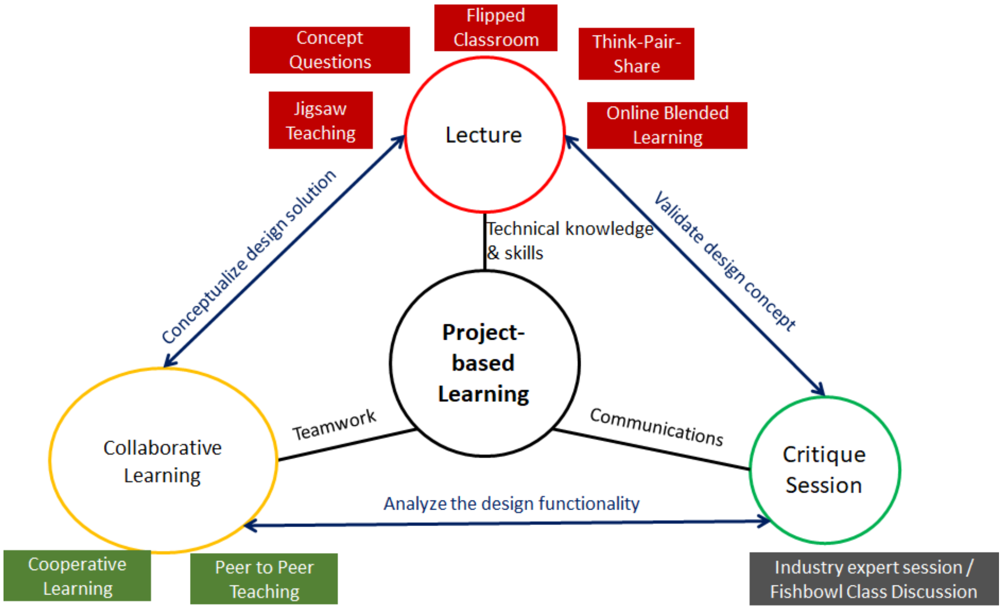

Figure 1. Integration of teaching methods to Active Learning approach (Project-based Learning - Final Year Capstone Project).

Co-funded by the Erasmus+ Programme of the European Union

The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

2. The different active learning strategies in Software Engineering

2.1 The rational

The aim of active learning strategies in Software Engineering is to provide a conducive environment for the development of knowledge, understanding and skills in the Software Engineering field. In the perspective of constructivist learning, the practical oriented environment supported with technology, social structure, learning by doing activities leads to a pedagogical praxis.

The challenge of developing active learning activities that leads to a pedagogical praxis should address all the different problems related to Software Engineering education can be explained through two different perspectives; the soft skills and the technical skills. The soft skill problems include instilling knowledge and intuition, experience, communications, holistic approach dealing with various human attitudes. While the technical skills need to enhance the problem solving ability in terms of changing requirements, compatibility, portability, flexibility, operability, complexity, multi disciplinary issues and others.

586297-EPP-1-2017-1-EL-EPPKA2-CBHE-JP

Co-funded by the Erasmus+ Programme of the European Union

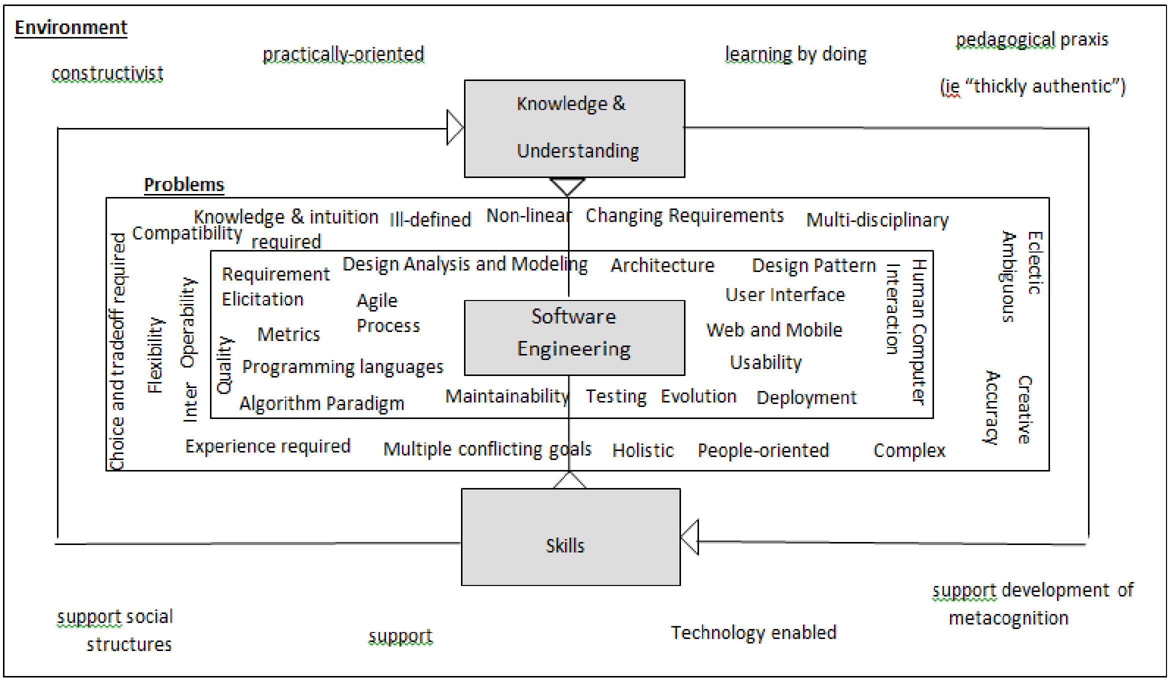

Figure 2: Environment for teaching Software Engineering and problems that need to be addressed (Adapted from Hainey (2010)).

The knowledge and skills in the Software Engineering field should cover all aspects of software development life cycle. These include requirement elicitation, design and modeling, implementation, architecture, human computer interactions, deployment, quality and others. Software Engineering students will confront the challenges of software development in real world environments. Grasping the knowledge and developing skills sets related to current advancement of technology on cloud computing, Internet of Things (IoT) and AI predictive systems requires teaching and learning strategies that are effective. Active learning is therefore seen to be an appropriate teaching and learning strategy to be chosen to train future software engineers. Figure 2 shows the summary of the ideal environment for active learning in teaching Software Engineering.

2.2 Active learning strategies and teaching methods in Software Engineering (Raja-Yusof, Abu-Bakar & Salim, 2018)

The common strategies and teaching methods of active learning in the Software Engineering courses were found through a literature search. We searched through Web of Science (WoS) database and Google Scholar Search Engine. The keywords used were “Active Learning” and “Software Engineering”. After the filtering process, we short-listed 49 papers presented in conferences or published in journals between the years 2005 - 2018. Table 1 shows the result of the search.

It was discovered that many strategies can be employed to conduct active learning in the Software Engineering subjects. The highly popular strategy is team and collaborative learning through projects or assignments; researchers who advocated or supported this strategy include Sibona, Pourreza, & Hill (2018), Tiwari, Saini, Singh, & Sureka (2018) and Marcos-Abed (2018). We noted that this approach was reported in more than 20 different articles. A moderately popular strategy is project which was reported by Díaz Redondo, Fernández Vilas, Pazos Arias, & Gil Solla, (2014); another moderately popular strategy is game-based learning which was reported by (Caulfield & Veal, 2011). These strategies were reported between 10 and 20 times in various articles. Here are some less popular active learning strategies: discussion (Tiwari, Saini, Singh, & Sureka, 2018), problem-based (Fonseca & Gomez, 2017), case-based (Hainey, Connolly, Stansfield, & Boyle, 2011), role play (Damian & Borici, 2012), peer to peer (Semushin, Tsyganova, Ugarov, & Afanasova, 2018) and others. These strategies were reported less than 10 times in various articles.

ACTIVE LEARNING Total Search Document Used = 49

Co-funded by the Erasmus+ Programme of the European Union

Year of Paper Publication: 2005 - 2018			
POPULARITY	SE SUBJECT /TOPICS	Strategies and teaching methods	FREQUENCY
High	Implementation/ programming/ development Software Engineering	Team/ Collaborative	more than 20
Moderate	Project Management Requirement	Project Based Game-Based	more and equals 10
Less	Agile SE methodology Design Architecture / Modeling Evaluation/Testing Configuration	Discussion Problem-Based Learning Case-based learning Role play Peer-to-peer Pitching/ Presentation Learning Outside class Experiential Simulation Open learning (students sharing source code)	less than 10

Co-funded by the Erasmus+ Programme of the European Union

The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

		Studio-based learning Laptop exercises Learning by doing Real Mentorship Design-based learning Brainstorming Review	

Table 1. Active Learning strategies and teaching methods in Software Engineering Subjects/ Topics.

Based on the search results, some of the most popular subjects that apply active learning are Implementation / Development/ Programming (Fonseca & Gomez, 2017) and Software Engineering (Sedelmaier & Landes, 2015). Moderately popular subjects that use active learning are Project Management (Claypool & Claypool, 2005) and Requirement (Damian & Borici, 2012). The less popular subjects that utilize active learning are Design (Claypool & Claypool, 2005), Architecture and Modelling (Sedelmaier & Landes, 2015), Evaluation, Testing (Holmes, Allen, & Craig, 2018) and Configuration (Krusche, Reichart, Tolstoi, & Bruegge, 2016).

2.3 Serious games for Software Engineering

Software Engineering is a diverse discipline that encompasses other domains of computer science as well. Being theoretical in nature it is a challenging task to teach the core Software Engineering concepts into games. However, over the years a surge is seen in the growth of games related to Software Engineering. Narrowing down to Software Engineering, literature shows only a handful of examples where GBL is used for teaching and learning SE. SE is taught as a series of lectures in most of the educational institutes with a semester project to develop as the main source to get hands-on experience of practical software development. However, the present system somehow fails to adequately teach the software process to the learners. Therefore, there is a need to provide the learners with an opportunity to practice the concepts as close to practical environment as possible.

In this section, we discuss some of the available games used for Software Engineering education (summarized in table 2).

Ref	Game name	Game type	Teaching program
(Bora & Rad, 2018)	Bubble Sort Level	Based on Unity 3D	Data Structures
(Tillmann et. al, 2014)	Code Hunt	Web-based game	Advanced Programming
(Ahmadi & Jazayeri, 2014)	AgentWeb IDE	Web-based game	Software Engineering
(Letra et al., 2015)	SimSE	Video game	Software Engineering Management
(Gednandes & Sousa, 2010)	Play Scrum	Board/card game	Software Engineering
Bezerra & Coutinho, 2013)	iTest Learning	Video game	Software Testing
(Gresse von	XMED	Video game	Software

Wangenheim, et. al, 2009)			Requirements
(Ribeiro & Paiva, 2015)	iLearnTest	Video game	Software Testing
(Long, 2007)	RoboCode	Desktop game	Coding
(Miljanovic & Bradbury, 2017)	RoboBug	Desktop game	Debugging
(Nogueira, 2018)	SE RPG	Video game	Software Engineering Models
(Shaw & Dermoudy, 2005)	Sim Java	Video game	Software Engineering
(Xia et al., 2012)	SimSoft	Video game	Software Engineering

Table 2. Summary of games developed for Software Engineering education.

There are games developed to teach the basic concepts of core courses of Software engineering like data structures, coding, requirements engineering, and testing. For instance, in Bubble sort level game the authors taught five sorting algorithms to the students using the game, followed by a post-game survey to analyze the game-based learning results. The results were then compared with the traditional learning group (Borna & Rad, 2018). For coding there are games like Sim Java SP (Shaw & Dermoudy, 2005), Robo Code (Long, 2007), Robo bug (Miljanovic & Bradbury, 2017), among others. Robo Code is an open-source learning desktop game started by Matthew Nelson and provided by IBM. The game is designed to help people learn to program in Java while enjoying the experience. Robo code is a serious game that helps the student to teach coding (Summary 2007). Robo bug is a Game in which a player assumes the role of a futuristic programmer trying to find the bug in the piece of code with the help of hints and help (Miljanovic & Bradbury, 2017). Likewise, for testing and debugging games play a very important role. Games like iTest Learning is used for teaching software testing courses through the serious game. It is a single-player game where the player develops a plan on how to test a specific project based on its specifications provided at the beginning of the game (Bezerra & Coutinho, 2013). iLearn Test is a single player game (Ribeiro & Paiva, 2015) focused on teaching Software Testing. It has different lessons or mini-games where each game covers a separate concept.

There are games introduced to help the students learn the basic concepts of Software development lifecycle management. Some of the games are specifically designed for requirements engineering while others focus on project management concepts. Play scrum is a board game that requires 2 - 5 players. Each player plays the role of Scrum Master. Play scrum is used to teach Software Requirements, Software Engineering Models and Methods, especially scrum (Fernandes & Sousa, 2010).

From the literature review, it can be seen that there is a considerable number of serious games focused on Software Engineering topics, especially on software project management (e.g., SimSE , SE RPG (Nogueira, 2018), PlayScrum (Fernandes & Sousa, 2010), SESAM (Drappa & Ludewig, 2000), AMEISE (Bollin et al., 2012), Sim JavaSP (Shaw & Dermoudy, 2005), SimSoft (Xia et al., 2012) , XMED (Gresse von Wangenheim et al., 2009), Problems and Programmers (Baker et al., 2003), DELIVER (von Wangenheim et al., 2012), ProDec (Calderón et al., 2017)) and software testing (e.g., U-Test (Silva, n.d., 2010), iTest Learning (Bezerra & Coutinho, 2013), iLearnTest (Ribeiro & Paiva, 2015)).

XMED is a video game that allows the player to follow the flow of a project from beginning to delivery. XMED game teaches the student the software requirement course and practices students to gather different requirements and analyze the requirement of the software through the game (Gresse von Wangenheim et al., 2009). SE RPG is a video game that simulates a Software Development environment in which the player needs to interact with different characters to advance the development of his project. SE RPG is used to teach students different Software Engineering courses like Software Requirements, Software Construction, Software Testing, Software Engineering Models, and Methods Software Quality (Nogueira, 2018). SimSoft is a Serious video Game aimed at teaching assessing Risks Management to players. The game starts with a small survey to understand how proficient the player is in this area. After this first phase, the project's description and details are provided to him. The Software Engineering courses like Software Maintenance, Software Engineering Models and Methods, and Engineering Foundations are learned through this course (Xia et al., 2012).

Literature shows that serious games are being used to teach Software Engineering concepts. However, a recent study shows that serious games have been most used to cover "Software Process", "Project Management" and "Software Requirements" knowledge area, while "Software Architecture", "Software Maintenance" and "Software Modelling" are the least covered topics (Campolina et al., 2018). The main reasons for not using these approaches are related to lack of knowledge, lack of information about relevant games for teaching Software Engineering, and the lack of time to plan and include these approaches in the classroom (Campolina et al., 2018). However, use of serious games in teaching Software Engineering subjects has shown considerable improvement in learning outcomes of the students (e.g. (Ghazali et al., 2018), (Inayat et al., 2016)).

3. Case studies of active learning in Software Engineering

3.1 Case study at the University of Malaya for Agile Software Development

The Agile Software Development course examines agile methods, including Extreme Programming (XP), Scrum, Lean, Kanban, Crystal, Dynamic Systems Development Method, Test-Driven Development and Feature-Driven Development to understand how rapid realization of software occurs most effectively. The ability of agile development teams to rapidly develop high quality, customer-valued software is examined and contrasted with teams following more traditional methodologies that emphasize planning and documentation. Students will learn agile development principles and techniques covering the entire software development process from problem conception through development, testing and deployment. Several issues of adopting agile methods are also discussed.

3.1.1 Description of participants

The course is an elective course for Masters of Software Engineering students in the Faculty of Computer Science & Information Technology of the University of Malaya. 14 students were enrolled in the course in Semester 1, 2019/2020 session. Half of the students are Malaysian while the other half are from Bangladesh and Afghanistan. Participants have been exposed to jigsaw active learning teaching method, however our focus was on Project-based Active learning strategy.

586297-EPP-1-2017-1-EL-EPPKA2-CBHE-JP

Co-funded by the Erasmus+ Programme of the European Union

The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

3.1.2 Description of active learning activities deploying ALIEN methodologies and tools, equipment, software, and educational material to be used

Each group of students are seated in an island. They use the workstation to display the discussion ideas. At the same time most of them will also use their own laptops for doing their individual work.

- The writable surface is also used to sketch their ideas.

- The drone was part of the project deliverables that fits within the context below:

a. Apply Agile - Scrum methodology for project development

b. Select the applicable gesture for selected area of development

c. Design a prototype for a gesture recognition system using drone

d. Identify the platform to output the response from the drone

e. Display the outcome on selected social media platform (Twitter)

Table 3, Figure 1, Figure 2, and Figure 3 show the list of equipment used and the pictures of the equipment.

Equipment	Features			Brief Description
1.Workstation	8th Generation Intel® Core™ i5-8400	8GB DDR4 2666MHz 16GB Intel® Optane™ memory	Premier Wireless Keyboard and Mouse	One workstation per island

	Processor Windows 10 Home 64bit	accelerated 1TB 7200 RPM HDD		
2. 32" LED Monitor Display	with HDMI Cable Portable Stand for LED TV Display Moveable stand with Adjustable shelf height for storage space	Integrated cable management system Heavy gauge columns constructed Resolution 1366 x 768	HDMI/USB/Component In (Y/Pb/Pr)/Composite In (AV) Connectivity Slim Type LED Type Wide Colour Enhance Slim edge Mold Design	One monitor per island

586297-EPP-1-2017-1-EL-EPPKA2-CBHE-JP

Co-funded by the Erasmus+ Programme of the European Union

The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

3. Heavy-duty casters	Max load capacity: 46 kg Power Extension Socket Tower Type-2-Tier	With reinforced design support the equipment weight while allowing for swift manoeuvrability	
4. Computer Peripherals	Professional HDMI Cable 2M Asus RT-AC58U AC1300 Dual-Band Gigabit Wireless Router		
5. Drones	Programmable Drones with accessories		To support programmable hardware projects which can relate to Software Engineering and the Internet of Things (IOT), Machine Learning and Cloud

			Computing.
10. Samsung Galaxy Tab A	A with S-Pen come with Miracast device		For teachers to control lecture materials in an active learning environment
11. Writable surface	Materials: Acrylic Glass - Background: Avery White	-Size & Thickness: 90 cm x 60 cm x 5 mm	To support brainstorming and discussion sessions with sketches and diagrams to facilitate active learning

586297-EPP-1-2017-1-EL-EPPKA2-CBHE-JP

Table 3. Equipment in TEALS.

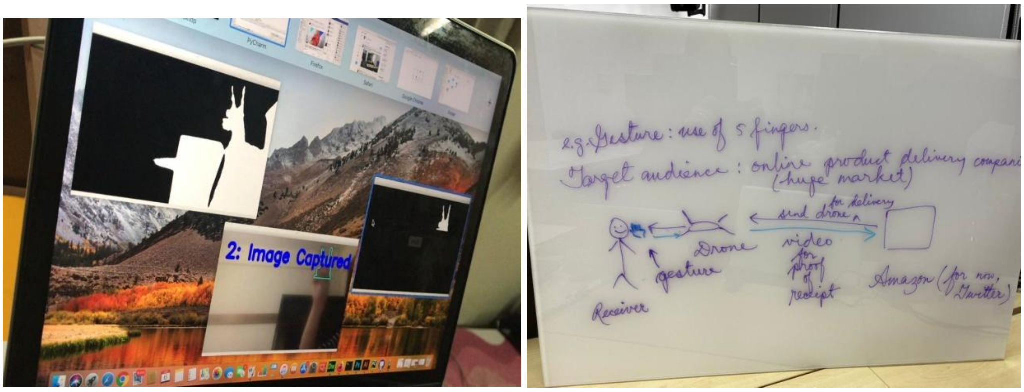

Figure 2. Writable surface.

Figure 3. Drones.

3.1.3 Software

The following software are being used and they are downloadable from the Internet

Pycharm: IDE for Python language

Python Interpreter

Related Python packages to access social media, gesture recognition libraries

3.1.4 Educational material (books, scenarios, etc. and sources)

Books:

Project Management Institute. (2017). Agile Practice Guide. Project Management Institute.

Kent Beck and Cynthia Andres. (2004). Extreme Programming Explained: Embrace Change (2nd Edition). Addison-Wesley Professional.

Ohno, Taiichi (1988). Toyota Production System. Productivity Press, Cambridge, MA.

Jeff Sutherland (2010). Scrum Handbook. Scrum Training Institute Press, MA.

David J. Anderson and Janice Linden-Reed (2010). Getting Started with Kanban for Software Development. DZone Refcardz. http://www.dzone.com

Mary Poppendieck, Tom Poppendieck. (2003). Lean Software Development: An Agile Toolkit. Addison Wesley. Upper Saddle River.

Alistair Cockburn. (2004). Crystal clear a human-powered methodology for small teams, retrieved at https://www.researchgate.net/publication/234820806_Crystal_clear_a_human-powered_methodology_for_small_teams

Tello Drone Resources, retrieved at

https://dl-cdn.ryzerobotics.com/downloads/Tello/Tello%20SDK%202.0%20User%20Guide.pdf

Others: See Project Brief

3.1.5 Description of the active learning session conducted in WOC7010 (Agile Software Development)

The active learning session conducted was a Project-Based Active Learning strategy. The objective of the project are as follows:

To show Agile development practices in a software project using Scrum methodology

Co-funded by the Erasmus+ Programme of the European Union

- o Practical – weekly monitoring
- o Report – at the end of the project

To design, implement and test an Agile software product specifically a gesture recognition system involving the usage of programmable drones and social media accounts

To present the software product to be evaluated

A description of the Project Brief was presented to the student in Week 9 of the course. They had to do the project within 4 weeks and the 5th week was the presentation. Every week students were evaluated as a group of their progress.

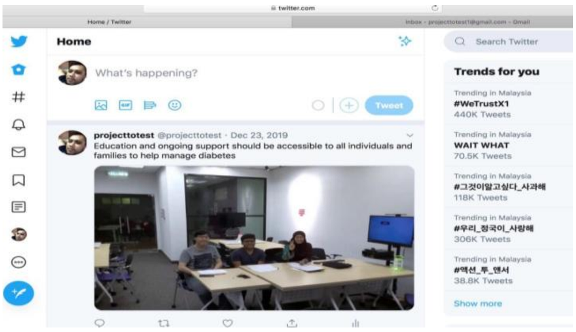

Figure 4.: Drones taking picture and sent it to twitter.

3.1.6 Assessment

The assessment was conducted using the following tools:

The FILA form

The Group Contract

The Reflection

The Report

The Presentation

The Post Test and the Pre-Test

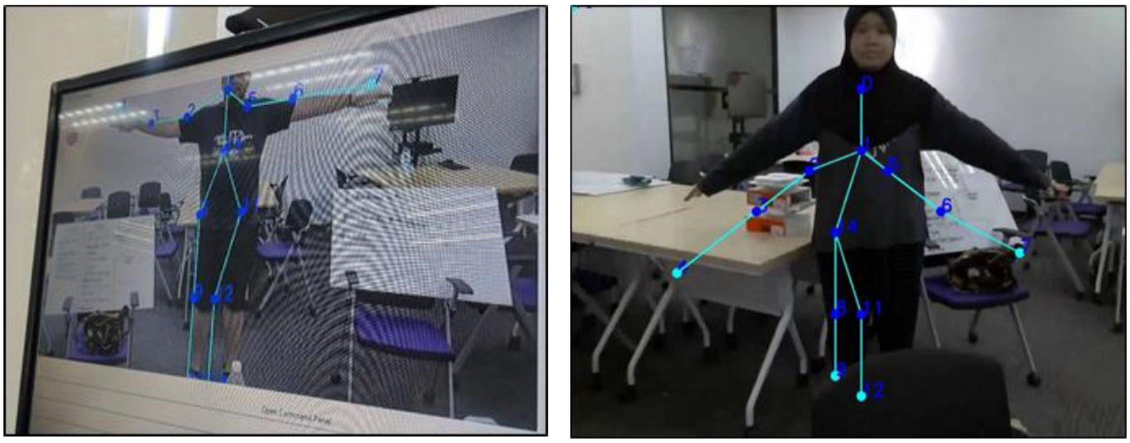

Figure 5. Recognizing gesture from the drone.

Co-funded by the Erasmus+ Programme of the European Union

Figure 6. Other picture sent to twitter.

During the execution of class, and photos of the group work (ask students to snap photos of their work). See examples of photos from RJRY class, last semester. https://drive.google.com/file/d/1DqoscHXtWMtpN4qN5BPnPyWafu9eX4OV/view?usp=sharing

Figures 3-5 show the feature of the implemented gesture recognition from the project based learning in TEALS.

3.2 Case studies in NUCES, Pakistan: teaching and learning Object-oriented Analysis and Design with 3D Game

The advent of information and communication technology (ICT) in education and learning has hugely improved the conventional teaching and learning methods. Game-based learning (GBL) is also a new facet of learning as compared to the traditional learning mechanisms. The concept of game-based learning emerged in early 80's as “serious games” (Navarro & van der Hoek, 2001). Game-based learning enhances problem solving skills in students and increases ease of learning (Vogel et al., 2006). Playing game is automatically connected with learning as mentioned in (Young et al., 2012) “you cannot play a game unless you learn it” (p.3.). Games engage player's interest which helps to achieve positive learning outcome. However, an interesting and attractive game not only constitutes of pedagogical principles but also comprises of interactive-ness, feedback, and challenges (Sehaba et al., 2005). These features make the learning experience interesting for the learners. GBL also tends to invoke teamwork and interactive skills among the learners which makes it a suitable format for continuing education (Ahmad et al., 2010).

Co-funded by the Erasmus+ Programme of the European Union

Researchers studied effects of game-based learning on students' performance, learning outcome, motivation level, problem solving skills, and critical thinking development in variable domains. Based on learners' interest and engagement, GBL outperforms traditional learning methods. For this the games need to be engaging, interesting, challenging and interactive. Several guidelines are available to design effective learning games that maintain a balance between enjoyment and education (Sehaba et al., 2005).

Literature shows that GBL is not only an effective mean of learning for per-school and primary level students, but it is equally effective for higher and tertiary level education (e.g. (Nunes et al., 2012)). Nevertheless, a handful of research studies are conducted on the use of GBL for computer science (CS) students. Therefore, this paper is aimed to conduct an empirical investigation on how GBL helps learning OOAD in university level students.

In summary, we aim to study the effects of GBL on studying OOAD course by answering these research questions as: RQ1-How does the use of OOAD-Game help the students to improve knowledge of the subject, RQ2-How is the use of OOAD-Game helpful in project completion for students, and RQ3-How does OOAD-Game improves learning outcome of the students. This study is unique as we have studied the effects of GBL in a theoretical course which includes no programming unlike the available studies on programming and mathematical skills development. Moreover, we have conducted the study on a real class setting with an ongoing course. This study furthers the understanding on the use of GBL in computer science and contributes to the knowledge on the topic by providing empirical results.

3.2.1 Experimental design

The research design followed in this study is experimental in order to gain insights on how the learners utilize the proposed game in learning OOAD course.

Co-funded by the Erasmus+ Programme of the European Union

We recruited 36 undergraduate students of Software Engineering discipline for this experiment. The students were divided into two groups (18 members each): control group (CG) and experimental group (EG). The distribution of students in both of the groups was kept random to avoid bias in the study. The experiment was designed to fit in the class standard duration of the semester (16 weeks), class (45 minutes) and lab (60 minutes).

Both groups attended classes to gain the basic knowledge of Software Engineering and lab sessions were conducted separately. In the start of the semester, EG was demonstrated about the game and its usage followed by an interactive 50 minutes session in which the EG members clarified their confusions about the game. The other group carried on with their projects in the conventional way. Both groups were divided in 6 teams (6 members each), 3 in each group to carry out their projects. Three projects were assigned to both of the groups, one for each team. Similar projects were chosen for both groups to make the assessment justified.

3.2.2 Materials and instruments for 3D-game for learning OOAD

An online 3D game is designed to help the learners with their practical comprehension of OOAD concepts. We have stored our helping material in terms of lectures, books and solved cases studies in the database for self-learning purpose). This game consists of several backgrounds or situations for the user to select e.g. meeting room, development hall, recreational area etc. (as shown in Figure 7). First the user selects the situation of her choice by clicking on the particular picture of the available situations. By logging into the game the user can see her online project members.. The user can start conversation by using the inbuilt chat application. It also provides offline communication support through emails and social media. The users can have conversations with each other (chat), can share pictures and documents, and also archive the conversations. All of these details are recorded in

MyProject. A screen shot is shown in Figure 8 which shows conversation of several actors taking place in leisure room environment.

Figure 7. Available situations.

This game enables the user to interact with others to collect requirements, define project goals, and set their future milestones. The user can check project progress by clicking on the respective tab. In each lab session the students can log in, open their projects, and interact with other group members in different roles, view previous conversations and records, check their progress, and also consult helping material.

586297-EPP-1-2017-1-EL-EPPKA2-CBHE-JP

Co-funded by the Erasmus+ Programme of the European Union

Figure 8. Screen shot of OOAD-game.

3.2.3 Evaluation and results

In this study, a game for learning object oriented analysis and design is proposed. The effectiveness of the proposed game is examined through an experiment. The experiment is performed on a class of 36 students divided into two groups of equal size as control and experimental group. Control group is taught the course in the usual way using lab based project development. However, the experimental group used the proposed 3D game for OOAD project development. The results are obtained through a questionnaire from both groups after the experiment.

3.3 Case study at the University Tenaga Nasional (UNITEN) for Requirements Engineering

Traditional method of unidirectional, face-to-face lecture, though still relevant, is no longer seen as the only means to impart knowledge to the students. Since it relies, to a great extent, on the lecturers' interpersonal skills, it can be dull and boring at times, resulting in little knowledge being imparted to the students, despite the use of teaching aids such as presentation slides. To improve the situation, more active involvement of the students in the learning process is needed. Requirements Engineering and Software Engineering subjects are no exception. Some topics in the subjects such as requirements elicitation and analysis, are difficult to comprehend without immediately applying the knowledge obtained to solve real software development scenarios. To do this, more practices are needed and more time is required, which is not feasible due to the limited lecture hours and the need to move on to cover other topics. Therefore, additional coaching on the topic beyond class hours is needed in order to help improve students' understanding. However, this has to be done in an interactive and exciting manner to avoid boredom in learning the topics outside class hours. As such, to make it more interesting and to increase students' engagement in the learning activities, digital game-based learning (DGBL) strategy approach is a possible solution to address the problem.

Therefore, in our study, a digital serious game is proposed with the aim to help students improve their understanding on the requirements elicitation and analysis topic by applying the knowledge gained using several scenarios in the game. The requirements elicitation and analysis tasks to be performed by the students are presented in the form of missions to be completed in the game. By playing the game, students are able to practice requirements elicitation and analysis skills outside class hours using the scenarios provided, be more engaged and interested in the learning process, which hopefully lead to the improved understanding of the topic. With improved understanding, the students can be better prepared to face the real problems in the real working environment later. The game, which is expected to be used as part of the overall learning solution, is designed and developed using the ADDIE model for instructional system design (ADDIE, 2017). Pilot test performed at the end of the development phase confirms the ability of the game to help students

Co-funded by the Erasmus+ Programme of the European Union

attain the learning objectives. Some issues with regards to the usability were also discovered, which will be addressed in future work.

3.3.1 Game design and development

The game is developed as a three-dimensional (3D) serious game. For the moment, it is a single player game. However, there is a plan to extend the game to allow multiplayers and enable online collaborative play. In this game, the player who assumes the role as a requirement analyst is given a mission to gather the requirements of a system to be developed (Figure 9). The player will gather the requirements from the non-player characters (NPCs). There is a total of seven NPCs in this game and they are categorized into two according to their roles. Firstly, there is the quest giver who gives the quest to the player. The quest is like a 'pass' for the player to start gathering the requirements. Timer will start once the quest is accepted. The second category of NPCs is the stakeholders who are the sources of the requirements to be gathered. Depending on the scenarios, they can be the clerks, administrative assistants, executives, project leaders, cashiers, managers and directors. This category of NPCs can be further divided into two, actual and dummy. Actual ones are the correct NPCs who can potentially provide correct and dummy requirements while the dummy NPCs are the characters that are not relevant to the system to be developed and hence only provide unrelated requirements.

Co-funded by the Erasmus+ Programme of the European Union

Figure 9. The player character at the start of the game.

There are at least three different scenarios of three levels of increasing difficulty. In the first level (level 1), the player has to gather the requirements for a research assistant application system. The second level is a supermarket scenario where the player needs to collect the requirements for a supermarket inventory system. The third level is a car service centre scenario where the requirements for an appointment booking system have to be gathered. The higher levels are harder in the sense that the scenarios have bigger scope with higher number of requirements to be gathered and analyzed. To meet the learning objectives of requirements elicitation and analysis topic, the requirements given by all NPCs are categorized based on their completeness, consistency, relevance and practicality, and the player will have to label the categories correctly. In level 1, the player is asked for the completeness and relevance of the requirements. Consistency is added in level 2 onwards and in level 3, the player is also asked to evaluate the practicality of the requirements. The game starts after the player chooses the 'Play' option from the main menu. The player is then given an instruction to go to the quest giver, who briefs the player about the mission. The timer starts counting once the quest is accepted and the player is given a maximum of

600 seconds to complete the mission. Since there are many NPCs in the respective scene, such as administrative staff, executive, manager et cetera, each of the NPC's roles and job descriptions will be displayed in the form of a dialog box whenever the player bumps into them. This allows the player to decide on the NPCs to talk to. This is part of the game design to allow the player to practice in determining the right persons to obtain the requirements from. Dummy NPCs are purposely included in the game for this reason. As each requirement is given by the NPCs in the form of a dialog box, the player will be given the options to accept the requirement provided by the NPC, to reject it (irrelevant), to mark it as inconsistent or impractical, depending on the game level.

The learning objective that serves as the basis for the design of the learning solution is the learning objective of the requirements elicitation and analysis topic stated below, which is taken from (Sheldon, 2018).

"To describe the activities in the requirements engineering process which are concerned with discovering requirements and analyzing requirements for incompleteness, inconsistency, relevance and practicality, as well as negotiating the final requirements for the system."

To be able to design the learning activities that will address the learning objective, the latter is broken down further into individual objectives as shown in Table 4.

Learning objective	Learning activity	Measure
To discover requirements (LO1)	To identify the right NPCs in the game to collect the requirements from	The number of NPCs, documents and observations correctly

Learning objective	Learning activity	Measure
	To identify the right documents in the game to collect the requirements from To identify the right scenes in the game to collect the requirements from	identified as sources of requirements
To analyse requirements for completeness (LO2)	To determine additional requirements not obtained from the game	Percentage of the number of correctly identified requirements from the total number of correct requirements The number of additional requirements included that match the real requirements
To analyse requirements for relevance (LO3)	To determine requirements that are correct	The number of requirements correctly included and excluded in the final list of requirements

Learning objective	Learning activity	Measure
To analyse requirements for consistency (LO4)	To determine requirements that are inconsistent	The number of requirements correctly identified as inconsistent
To analyse requirements for practicality (LO5)	To determine requirements that are impractical	The number of requirements correctly identified as impractical
To negotiate final system requirements (LO6)	Collaborative discussion between the students in groups on the outcomes of the game	The number of negotiated requirements that are acceptable

Figure 10. Learning objectives defined for the subject.

3.3.3 Evaluation and results

A pilot test is performed at the end of the development phase to obtain feedback on the usability of the game and to evaluate students' achievement with regards to the specified learning objectives. Figure 10 show an ALIEN PBL lab setting for the use of the game in the classroom. For the pilot test, only the first level of the game is covered. Participation in the pilot test is on voluntary basis. However, participants are required to have completed at least one of the following subjects; System Analysis and Design (SAD), Fundamental of Software Engineering (FSE) and Requirements Engineering (RE). This is to ensure that they have sufficient knowledge on requirements elicitation and analysis needed to play the game. A briefing is given to the participants before the pilot test to explain about the purpose and procedures of the test. They are then asked to play the game. The time taken to complete the game is recorded using a mechanism embedded in the game. As backup, the screen recorder is also activated throughout the play. After completing the game, the participants are asked to answer an online questionnaire comprising five sections on demographics, game usability, specific game details, usefulness and effectiveness and player's perception. The questions were adapted from (Vidani, Chittaro and Carchietti, 2010). There are a total of 21 questions in the questionnaire with 18 close ended and three open ended questions. Close ended questions are answered using a 5-point Likert scale with 1 being "strongly disagree" and 5 being "strongly agree".

With regards to usability, the focus of discussion is on the section of the questionnaire that evaluates game usability, which comprises four questions. Reliability test performed yielded Cronbach's Alpha values of 0.9330 for the overall consistency of the questionnaire and 0.7765 for the consistency of the questions on game usability. Table 4 shows the questions in the game usability section and their respective median values. From the results, it can be seen that the controls used in the game are deemed easy to remember by the participants and that they can concentrate on the game without being disturbed by the need to remember the commands and controls. However, only moderate scores are recorded for the game difficulty and navigability, which indicate rooms for improvement. With respect to this, the following comments and suggestions were given by the participants to improve usability, which were mostly on game controls.

To use keyboard keys only in controlling the movement of the player instead of using the combination of mouse and keyboard. One of the respondents mentioned that it was hard to use both devices to move around and more time was taken to get used to coordinate them.

To increase the sensitivity of the controls so that the movement of the player in the game is smoother.

To address the above, the use of mouse and keyboard will be modified during the actual implementation and evaluation phases. The keyboard will be set to control the player's movement entirely and the mouse will be used to select the requirements. Professional game developers' expertise will also be sought to improve the game controls. A more complete results including the learning outcomes attainment can be found in Ibrahim, Soo, Soo and Aris (2019).

Figure 11. The ALIEN PBL lab and the use of the game in the classroom session.

3.4 Case studies in IOE,TU, Nepal: application of centralized visualization system (CVS) lab for PBL and AL

In case of Institute of Engineering (IOE), Tribhuvan University (TU), Problem Based Learning (PBL) and Active Learning (AL) have been introduced recently at Pulchowk Campus for masters and bachelors level courses. PBL/AL was first initiated in selected courses offered under the Department of Electronics and Computer Engineering.

Figure 12. Applications of PBL for conduction of courses and robotic competitions at IOE, TU.

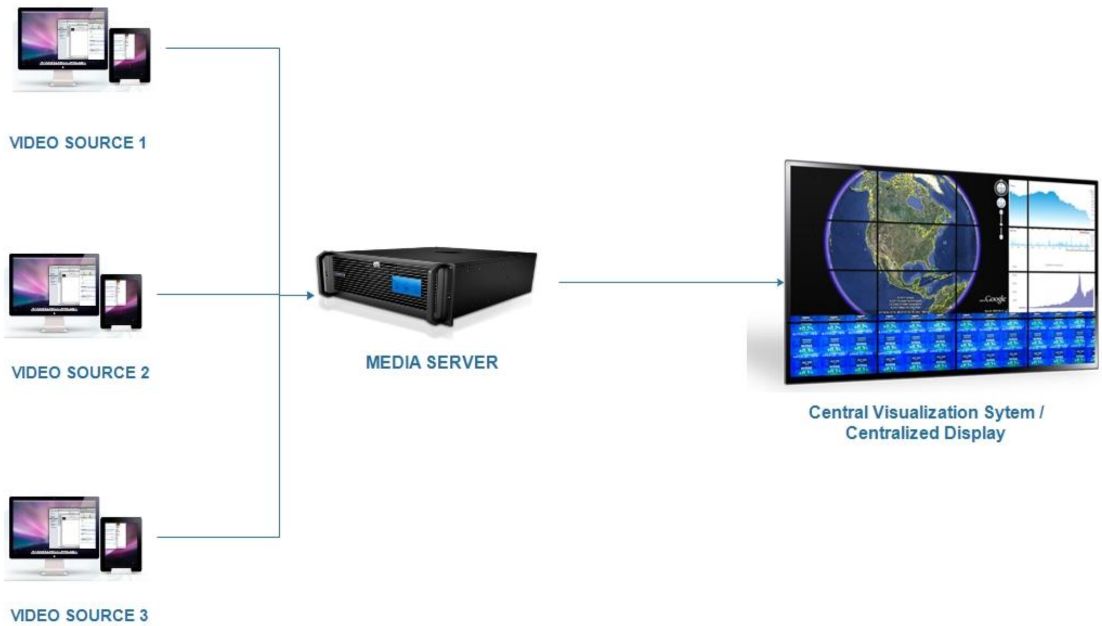

Figure 13. Architecture of CVS lab at IOE, TU.

Co-funded by the Erasmus+ Programme of the European Union

However, initiative has been taken to used in other disciplines too (Operation Research and Management Science, Multi-criteria Design Analysis etc. under Department of Mechanical and Aerospace Engineering). PBL methods were also applied in the Robotics Club at Pulchowk Campus, IOE. The club involves multidisciplinary teams of students and faculty from departments of Mechanical and Aerospace Engineering, Electronics and Computer Engineering and Electrical Engineering. PBL methods and active learning was used for design and development of robots and automation in the club. The club annually participates in the ROBOCON International competition and National Competitions for robotics as seen in Figure 12.

A Centralized Visualization System (CVS) Lab for PBL / Active Learning has been developed and implemented at IOE, TU. The architecture of CVS lab is given in Figure 13.

Implementation of the CVS lab was started with pilot testing in Msc. program in Computer Systems and Knowledge Engineering, offered at the Department of Electronics and Computer Engineering. The CVS lab was used in project works of the course "Knowledge Engineering" offered in the first semester of the program. 20 students are typically enrolled in the course each year. The objectives of this course are

- To familiarize the students with the basic concepts of knowledge engineering
- To teach the basics of knowledge acquisition methods, IR, NLP and machine learning techniques
- To teach students about knowledge representation, logic and reasoning
- To introduce the students into the field of Semantic Web and ontology engineering

The details of the implementation of the CVS lab for PBL/AL in the Course on Knowledge Engineering consists of following:

- The students were divided in three or four groups.

Each group of students is assigned a mini-project to do a case study and develop a simple knowledge-based system that would be useful for the institution or society.

Each group of students collects and review materials on the web and campus intranet. They have meetings/interactions with experts and contact persons of the concerned knowledge domain. They also collect necessary data and documents from the person and/or the internet.

They explore existing solutions from the web and propose a knowledge-based system for the project.

Each group would make group presentation followed by feedback provided by other groups and the instructor and domain experts.

Each group developed a solution prototype and demonstrated in the final presentation to all other groups, instructor and the concerned domain experts.

The groups were evaluated based on these presentations and the solution prototypes developed.

Few selected project titles demonstrated and presented at the CVS lab are given below.

- Online Form Photo Validation System
- Exam Results and Entrance Score Analytics
- Online Assessment Marks Entry System
- Exam Papers Package Handling System

Implementation of the CVS lab was also started with pilot testing in the elective course “Big-data Application and Analytics” offered in Msc. program, at the Department of Electronics and Computer Engineering. Minimum of 6 students are enrolled in the course when offered. The objectives of this course are

- To give overview of Big data and latest Trend in Big Data Analytics

586297-EPP-1-2017-1-EL-EPPKA2-CBHE-JP

Co-funded by the Erasmus+ Programme of the European Union

- To introduce the technologies for Handling Big Data
- To perform basic exploration of large, complex datasets and understand scalable big data analysis
- To apply big data tool for advanced analytics disciplines such as predictive analytics, data mining, text analytics and statistical analysis.

The CVS lab was used in laboratory works of this course. In the beginning of the course, the lab was used to demonstrate the installation procedure of the different big data tools. The students used the workstations and lab connectivity to collect required documents for the tools setup and they tried to follow the installation procedures and shared the issues encountered during installation to the entire class. Either one of the students or instructor shows the way to solve the issues. Once the installation is completed, in the next lab session, they were asked to work in a group to

- Define a real world problem to be solved using big data stack.
- Specify along with the sources from where the data will be collected. Identify the important features that will be taken into consideration while designing the system.
- Determine how data will be stored. Choose the components of big data ecosystem to design the system. Illustrate in detail with a block diagram.
- Develop the algorithms that will be used to process the collected data. Determine the way to visualize the results.

The students searched different articles and explore existing solutions from the web to identify the problems that can be solved through big data. They propose a big data project with the appropriate use of big data tools to solve the identified problem. There were group presentations and feedback provided to each group by other groups and the instructor. In the last session, each group developed a solution prototype and demonstrated in the final presentation to all other groups, instructor. The students were evaluated through the final presentation of the complete system that has been developed.

Implementation of the CVS lab is also planned for Bachelor level Course "Computer Graphics", offered at the Department of Electronics and Computer Engineering. The CVS lab can be used for laboratory work and project works of the course ""Computer Graphics". 96 students each are typically enrolled in the course each year. The objectives of this course is to familiarize with graphics hardware, line and curve drawing techniques, techniques for representing and manipulating geometric objects, illumination and lighting models.

The students can be taken to Lab to demonstrate the techniques for representing and manipulating geometric objects, illumination and lighting models in groups and in different sessions. They can be asked to implement the techniques demonstrated and the solutions and issues they encountered can be shared to the others in the big screens.

The students can also use this lab to demonstrate the computer graphics project to other students and faculties. Similarly, implementations are also ongoing for courses on Image Processing, Operation Research and Management Science.

TU is still working on integrating the CVS lab application in more existing and new courses in the bachelors and masters programs. It is evaluating feedback/suggestions of students/teachers on effective use of the CVS, preparing to conduct capacity building activities for the faculty and staff.

586297-EPP-1-2017-1-EL-EPPKA2-CBHE-JP

Co-funded by the Erasmus+ Programme of the European Union

4. Conclusion

Active learning has been identified as an appropriate and effective learning strategies in Software Engineering field for the development of knowledge for both the soft skills and the technical skills. Active learning also can be deployed into teaching and learning approaches in all aspects in software development life cycle – requirement elicitation, design, implementation, architecture, HCI, deployment, quality and etc. Among the popular active learning strategies used in Software Engineering are team and collaborative learning, project-based and game-based learning, on top of many more strategies. Several case studies of the implementation of active learning in Software Engineering done by partner universities are presented in this document. The University of Malaya has implemented project-based active learning in the Agile Software Development course for Masters of Software Engineering. The project aims to show agile development practices in a software project using Scrum methodology, to design, implement and test an agile software product and to showcase the software product to be evaluated. The assessment of students learning was conducted using several tools such as FILA form, report, pre- and post-tests, etc. At NUCES, Pakistan, game-based active learning has been implemented in the Object-oriented Analysis and Design course, one of the theoretical courses in Software Engineering. An online 3D game is designed to help the students with their practical comprehension of OOAD concepts. The game allows collaboration and interaction among team members in different roles to define project goals, to collect requirements, to set future milestones and to check project progress. Effectiveness of the game was examined via an experiment that comprises of control group and experimental group, and results were obtained through a questionnaire.

Co-funded by the Erasmus+ Programme of the European Union

At the University Tenaga Nasional (UNITEN), Malaysia, digital game-based learning via a 3D serious game has been implemented for requirement elicitation and analysis topics. This approach is believed to be able to increase students' engagement in the learning activities because of the gamification elements such as interactivity and the excitement. The game allows the player i.e. the requirement analyst to complete a mission to gather the requirements of a system to be developed from the non-player characters. Several scenarios are available in the game in which each scenario has a different difficulty level and different set of learning objectives to achieve. The requirements given by all the NPCs are categorized based on their completeness, consistency, relevance and practicality. A pilot test has been performed at the end of the development phase to obtain feedback on the usability of the game and to assess student's ability in achieving the specified learning objectives.

At the Institute of Engineering, Tribhuvan University, Nepal, problem-based learning method was adopted in several courses such as Knowledge Engineering and Big-data Application and Analytics for MSc. program offered by the Department of Electronics and Computer Engineering. Centralized Visualization System (CVS) Lab has been developed and used for project and/or laboratory works of the courses. Students are divided into groups and assigned a mini-project to develop a knowledge-based system which involve several collaboration and interaction activities, as well as installation procedure implementation of big data tools for big data project. The CVS lab is also planned to be used for Computer Graphics, Image Processing, Operation Research and Management Science courses for bachelor level.

References

1. The ADDIE Model Infographic, e-Learning Infographics, (2017). [Online]. Available: https://elearninginfographics.com/the-addie-model-infographic/. [Accessed: 31-Jul-2019].
2. Ahmad, W. F. W., Akhir, E. A. P., & Azmee, S. (2010). Game-based learning courseware for children with learning disabilities. 2010 International Symposium on Information Technology, 1, 1–4. https://doi.org/10.1109/ITSIM.2010.5561303
3. Ahmadi, N., & Jazayeri, M. (2014). Analyzing the Learning Process in Online Educational Game Design: A Case Study. 2014 23rd Australian Software Engineering Conference, 84–93. https://doi.org/10.1109/ASWEC.2014.34
4. Baker, A., Navarro, E. O., & van der Hoek, A. (2003). An experimental card game for teaching Software Engineering. Proceedings 16th Conference on Software Engineering Education and Training, 2003. (CSEE T 2003)., 216–223. https://doi.org/10.1109/CSEE.2003.1191379
5. Bezerra, C. I. M., & Coutinho, E. (2013, July 23). Avaliação do Jogo iTestLearning: Um Jogo para o Ensino de Planejamento de Testes de Software.
6. Bollin, A., Hochmuller, E., & Samuelis, L. (2012). Teaching Software Project Management using Simulations - The AMEISE Environment: From Concepts to Class Room Experience. 2012 IEEE 25th Conference on Software Engineering Education and Training, 85–86. https://doi.org/10.1109/CSEET.2012.33
7. Bonwell, C. C., & Eison, J. A. (1991). Association for the Study of Higher Education.; ERIC Clearinghouse on Higher Education. Retrieved from https://files.eric.ed.gov/fulltext/ED336049.pdf
8. Borna, K., & Rad, H. M. (2018). Serious Games in Computer Science Learning Goals. 2018 2nd National and 1st International Digital Games Research Conference: Trends,

Technologies, and Applications (DGRC), 161–166. https://doi.org/10.1109/DGRC.2018.8712030

9. Calderón, A., Ruiz, M., & Orta, E. (2017). Integrating Serious Games as Learning Resources in a Software Project Management Course: The Case of ProDec. 2017 IEEE/ACM 1st International Workshop on Software Engineering Curricula for Millennials (SECM), 21–27. https://doi.org/10.1109/SECM.2017.3
10. Campolina, P., Souza, M., & Figueiredo, E. (2018, October 1). Games and Gamification in Software Engineering Education: A Survey with Educators. https://doi.org/10.1109/FIE.2018.8658524
11. Caulfield, C., & Veal, D. (2011). Artigo-A systematic survey of games used for Software Engineering education. http://doi.org/10.5539/mas.v5n6p00
12. Claypool, K., & Claypool, M. (2005). Teaching Software Engineering through game design. Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education - ITiCSE ’05, 123. http://doi.org/10.1145/1067445.1067482
13. Damian, D., & Borici, A. (2012). Teamwork, Coordination and Customer Relationship Management Skills: As Important as Technical Skills in Preparing Our SE Graduates. In Proceedings of the First International Workshop on Software Engineering Education Based on Real-World Experiences (EduRex ’12).
14. Díaz Redondo, R. P., Fernández Vilas, A., Pazos Arias, J. J., & Gil Solla, A. (2014). Collaborative and role-play strategies in Software Engineering learning with web 2.0 tools. Computer Applications in Engineering Education, 22(4), 658–668. http://doi.org/10.1002/cae.21557

15. Drappa, A., & Ludewig, J. (2000). Simulation in Software Engineering training. Proceedings of the 22nd International Conference on Software Engineering, 199–208. https://doi.org/10.1145/337180.337203 England, 2006), p. 24.
16. Fernandes, J. M., & Sousa, S. M. (2010). PlayScrum—A Card Game to Learn the Scrum Agile Method. 2010 Second International Conference on Games and Virtual Worlds for Serious Applications, 52–59. https://doi.org/10.1109/VS-GAMES.2010.24
17. Fonseca, V. M. F., & Gomez, J. (2017). Applying Active Methodologies for Teaching Software Engineering in Computer Engineering. IEEE Revista Iberoamericana de Tecnologias Del Aprendizaje, 12(3), 147–155. http://doi.org/10.1109/RITA.2017.2738178
18. Ghazali, S. N. H., Salim, S. S., Inayat, I., & Ab Hamid, S. H. (2018). A risk poker based testing model for scrum. Computer Systems Science & Engineering, 33(3), 169–185.
19. Gresse von Wangenheim, C., Thiry, M., & Kochanski, D. (2009). Empirical evaluation of an educational game on software measurement. Empirical Software Engineering, 14(4), 418–452. https://doi.org/10.1007/s10664-008-9092-6
20. Holmes, R., Allen, M., & Craig, M. (2018). Dimensions of Experientialism for Software Engineering Education. Proceedings of the 40th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET ’18), 9. http://doi.org/10.1145/3183377.3183380
21. Ibrahim, Z., Soo, M. T., Soo, M. C. and Aris, H. (2019). The Development of a Serious Game for Requirements Elicitation and Analysis, IEEE Conference on Teaching, Assessment and Learning for Engineeringrning (TALE), Yogjakarta, Indonesia, pp. 46-51, doi: 10.1109/IC3e.2018.8632650.

22. Inayat, I., Inayat, Z., & Amin, R. U. (2016). Teaching and Learning Object-Oriented Analysis and Design with 3D Game. 2016 International Conference on Frontiers of Information Technology (FIT), 46–51. https://doi.org/10.1109/FIT.2016.017
23. Information Systems Committee (JISC), Designing Space for Effective Learning: A Guide to 21st Century Learning Space Design (Bristol, U.K.: Higher Education Funding Council for
24. Joint Information Systems Committee (JISC) (2006), Designing Space for Effective Learning: A Guideto 21st Century Learning Space Design, p.30.
25. Krusche, S., Reichart, B., Tolstoi, P., & Bruegge, B. (2016). Experiences from an Experiential Learning Course on Games Development. Proceedings of the 47th ACM Technical Symposium on Computing Science Education (SIGCSE '16). http://doi.org/10.1145/2839509.2844599
26. Les Watson, pro vice-chancellor, Glasgow Caledonian University, quoted in Joint
27. Letra, P., Paiva, A. C. R., & Flores, N. (2015). Game Design Techniques for Software Engineering Management Education. 2015 IEEE 18th International Conference on Computational Science and Engineering, 192–199. https://doi.org/10.1109/CSE.2015.42
28. Long, J. (2007). Just For Fun: Using Programming Games in Software Programming Training and Education - A Field Study of IBM Robocode Community. J. Inf. Technol. Educ. https://doi.org/10.28945/216
29. Marcos-Abed, J. (2018). Poster: A Matrix for Analyzing Projects in Software Engineering Courses. In Proceedings of the 40th International Conference on Software Engineering (ICSE '18).
30. Miljanovic, M., & Bradbury, J. (2017). RoboBUG: A Serious Game for Learning Debugging Techniques. 93–100. https://doi.org/10.1145/3105726.3106173

Co-funded by the Erasmus+ Programme of the European Union

31. Navarro, E., & van der Hoek, A. (2001). Adapting Game Technology to Support Individual and Organizational Learning. 347–354.
32. Nogueira, J. M. F. E. (2018). Game Design Patterns in Serious Games for Software Engineering Education.
33. Nunes, T., Barros, R., Evans, D., & Burman, D. (2012). A Game-Based Working Memory Intervention for Deaf Children. In S. De Wannemacker, S. Vandercruysse, & G. Clarebout (Eds.), Serious Games: The Challenge (pp. 31–39). Springer. https://doi.org/10.1007/978-3-642-33814-4_4
34. Raja-Yusof, Raja-Jamilah, Abu-Bakar, Aishah, Salim, Siti Salwah, (2018). Designing spaces for active learning in teaching Software Engineering courses, International Conference on Sciences, Technology, and Social Sciences 2018 (ICSTSS2018).
35. Ribeiro, T., & Paiva, A. (2015). ILearnTest: Educational game for learning software testing. https://doi.org/10.1109/CISTI.2015.7170608
36. Sedelmaier, Y., & Landes, D. (2015). Active and Inductive Learning in Software Engineering Education. In 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering (Vol. 5, pp.418–427). IEEE. http://doi.org/10.1109/ICSE.2015.174
37. Sehaba, K., Estraillier, P., & Lambert, D. (2005). Interactive Educational Games for Autistic Children with Agent-Based System. 3711, 422–432. https://doi.org/10.1007/11558651_41
38. Semushin, I. V., Tsyganova, J. V., Ugarov, V. V., & Afanasova, A. I. (2018). The WHATs and HOWs of maturing computational and Software Engineering skills in Russian higher education institutions. European Journal of Engineering Education, 43(3), 446–472. http://doi.org/10.1080/03043797.2017.1385594

Co-funded by the Erasmus+ Programme of the European Union

39. Sheldon, Frederick T., (2018) "Requirements Elicitation and Analysis." From Requirements Engineering Processes and Techniques by G. Kotonya and I. Sommerville 1998, Wiley. [Online] : https://www.csm.ornl.gov/~sheldon/cs531/ch3.pdf
40. Shaw, K., & Dermoudy, J. (2005). Engendering an Empathy for Software Engineering. 42, 135–144.
41. Sibona, C., Pourreza, S., & Hill, S. (2018). Origami: An Active Learning Exercise for Scrum Project Management.: EBSCOhost. Journal of Information Systems Education, 29(2), 105–116.
42. Silva, A. C. (n.d.). JOGO EDUCACIONAL PARA APOIAR O ENSINO DE TÉCNICAS PARA ELABORAÇÃO DE TESTES DE UNIDADE. 195.
43. Tillmann, N., Halleux, J., Xie, T., & Bishop, J. (2014). Code hunt: Gamifying teaching and learning of computer science at scale. 221–222. https://doi.org/10.1145/2556325.2567870
44. Tiwari, S., Saini, V., Singh, P., & Sureka, A. (2018). A Case Study on the Application of Case-Based Learning in Sooware Testing. Proceedings of the 11th Innovations in Software Engineering Conference (ISEC '18). http://doi.org/10.1145/3172871.3172881
45. Vidani, A. C., L. Chittaro, and Carchietti, E. (2010). Assessing Nurses' Acceptance of a Serious Game for Emergency Medical Services, in 2010 Second International Conference on Games and Virtual Worlds for Serious Applications, Braga, Portugal, 2010, pp. 101–108.
46. Vogel, J., Vogel, D., Cannon-Bowers, J., Bowers, C., Muse, K., & Wright, M. (2006). Computer gaming and interactive simulations for learning: A meta-analysis. Journal of Educational Computing Research, 34, 229–243. https://doi.org/10.2190/FLHV-K4WA-WPVQ-H0YM

586297-EPP-1-2017-1-EL-EPPKA2-CBHE-JP

Co-funded by the Erasmus+ Programme of the European Union

47. von Wangenheim, C. G., Savi, R., & Borgatto, A. F. (2012). DELIVER! – An educational game for teaching Earned Value Management in computing courses. Information and Software Technology, 54(3), 286–298. https://doi.org/10.1016/j.infsof.2011.10.005

48. Xia, J. (Cecilia), Caulfield, C., Baccarini, D., & Yeo, S. (2012). Simsoft: A game for teaching project risk management. Creating an Inclusive Learning Environment: Engagement, Equity, and Retention. Proceedings of the 21st Annual Teaching Learning Forum. Teaching and Learning Forum 2012. https://espace.curtin.edu.au/handle/20.500.11937/8519

49. Young, M. F., Slota, S., Cutter, A. B., Jalette, G., Mullin, G., Lai, B., Simeoni, Z., Tran, M., & Yukhymenko, M. (2012). Our Princess Is in Another Castle: A Review of Trends in Serious Gaming for Education. Review of Educational Research, 82(1), 61–89. https://doi.org/10.3102/0034654312436980

586297-EPP-1-2017-1-EL-EPPKA2-CBHE-JP

Co-funded by the Erasmus+ Programme of the European Union