LEARNING THROUGH PLAYING: DEVELOPMENT OF SOFTWARE ARCHITECT'S SKILLS WITH BUILDING BLOCKS

Sabrina Ahmad, Emaliana Kasmuri, Noor Azilah Muda

Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka (MALAYSIA)

Abstract

Software Architecture and Design is a course introduced in the curriculum of Computer Science Bachelor Degree. It is a part of the software engineering body of knowledge to instill good practice in software development. The teaching and learning method of delivering competence in a high level abstraction is challenging in way to obtain knowledge appreciation among students. The experience of deploying learning through playing in a tutorial session encourages students' engagement, focus and appreciation of the teaching and learning process. The knowledge is delivered, experienced and actively discussed among students to discover more. This paper presents the design and implementation of the learning through playing building blocks for the purpose of understanding software architect's roles and responsibilities. The analysis based on the observation of the conduct is discussed and lessons learnt are elaborated.

Keywords: skills development, learning through playing.

1 INTRODUCTION

As part of the software engineering discipline, software architecture and design course cannot be taught exclusively through lecture format in the classroom. This is due to the nature of software engineering discipline which emphasized on competence and not just a body of knowledge. Any knowledge delivery of principles or experiences that is not backed up by active and regular participation by students is sure to miss the essence of what the students need to learn. This is supported by Denning [1] who said that computer science and engineering degrees should be based at least in part on demonstrations of accomplishments and competencies.

Professions from architecture to psychoanalysis usually exhibit reflection-in action. Expertise is the interplay of two competencies; core competencies that permit the practitioner to act respond effectively in familiar problem situations, and reflective skills that let the practitioner reasons about his or her skills and knowledge when the most immediate course of action seems likely to be unsuccessful. Translated into software engineering discipline, Schon [2] distinction is between the type of competence that a designer uses when making design decisions and the type of competence that leads to reason about the design method itself. Skills of the first type can be taught through lecture and by applying through small exercises. On the other hand, skills of the second type are extremely difficult to teach by instruction, because their effective deployment depends on the practitioner being sensitive to a wide range of contextual effects; some of them are not within the field of engineering at all. The skills to communicate the architecture design which involved maintenance consideration and quality attributes prioritization need more than mere explanation. It involves the ability to handle argument and the reasoning competence.

This paper discusses on the effort to introduce and to experience software architecture skills through playing building blocks. Following introduction, Section 2 explains the art of learning through playing. Next is the methodology and design of the class activity to embed playing in the learning process. This is followed by the implementation procedures. Then, discussion and lessons learnt are elaborated. This paper is concluded with conclusion section.

2 LEARNING THROUGH PLAYING

Playing is usually known as activity done for its own sake, characterized by the process rather than the end results, rich in flexibility and offer positive affect. Learning through playing is popular for early childhood development [3]. Children invest time and energy in play and there are opportunities for learning when they do play. This is generally true with young mammals even though other mammals show much less variety of play forms than human children. These findings suggest that play has development benefits [4, 5]. Benefits of playing are not only for early childhood development, it benefits a wide range of ages in many ways for many different purposes. As an example, playing games gained escalating popularity to engage students in higher education [6, 7]. This effort support the deployment of active learning in which students are fully engaged in the learning process. Active learning happens to be among seven principles for good practice in undergraduate education [8].

2.1 Learning Through Play in Software Architecture

Learning through play is an activity mainly used for pre-school children. Play engaged the children with the environment and giving them the new experience. Play developed the children social skills, self-confidence and content knowledge. The effectiveness of learning through play is implemented in the software architecture to expose the students with the required skills to design and produce business solution using software technology.

In the play, the students are given a set of problems, which they have to solve and deliver it within specific time. The play is design to inculcate cognitive and psychomotor skills, which are related to software architecture. There are challenges that the students need to address in completing their task as a software architect in the play. The challenges will develop the skills required as software architect as dictated in the foundation of software architecture book of knowledge.

Learning software architecture is not only about the IT technology that is able to solve the business problem. The dimension of software architecture is more than that. It covers design of the solution, human dynamics, quality attributes, IT environment and business technology strategy. Human dynamic deals with conveying design solution to stakeholders, managing the culture of the organization, interactions with team members, negotiating with the stakeholder and presenting report using oral and writing skills. The solution and design strategy must consider the IT environment that covers the functional and procedural aspects of IT organization.

2.2 Characteristics of Software Architect

Software architect is a person who is able to use art and science to deliver valuable software technology for the business. The person has performed various IT roles such as developer, system analyst, project manager, network or server engineer etc. Software architect is an educable profession rather than role that has to be carried in the software development team. Software architecture is a specialization embodied in the IT architecture book of knowledge.

A software architect carries certain responsibilities to deliver the business solution using software technology. Mainly, a software architect focuses on the strategy to deliver and develop software technology solution and implementation to the business. The strategy is to design the software component, utilizing existing component and made decision between buy or build decision for the component to be used. Other than that, the software architect needs to leverage the capability of the chosen component with the optimization of the software quality attributes. The decision is made from series of interaction and discussion with stakeholders, project manager, development team and business users.

2.3 Relation of Learning through Playing with Software Architecture

A software architect is answerable to IT architecture challenges as shown in Figure 1. These challenges define the skills sets that need to be inculcated to the software architects. These skills can be molded through a small play activity designed by the instructor for the lab activity. For this course unit (Software Architecture and Design), the students need to build a structure from a set of building blocks.

The students need to understand the requirements given by the instructor, design solutions for the requirements, consider the quality attributes to be imparted into the design, communicate and implement the design, connect the project with the organization or environment and deliver the solution within specific time. The requirements are given by the instructor and to be completed within specific time. The conformance to the requirements and delivering the solutions within the specific time is exercising the skills of understanding the requirements and compressed business processing time. The students need to discuss the solution structures, design concepts, magnitude of the structures and possibilities to be integrated with the current environment. The ability to justify the design of the structure is exercising the challenges of communicating the design, conceptualizing object, dealing with scalability, integrating entities, isolated islands of IT project and implementing quality design.

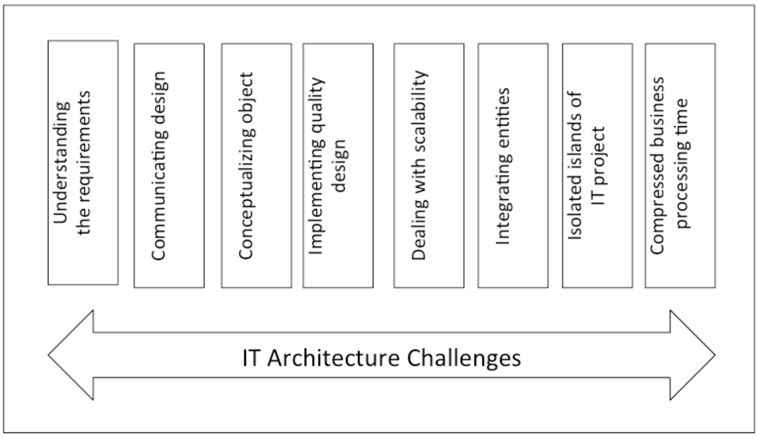
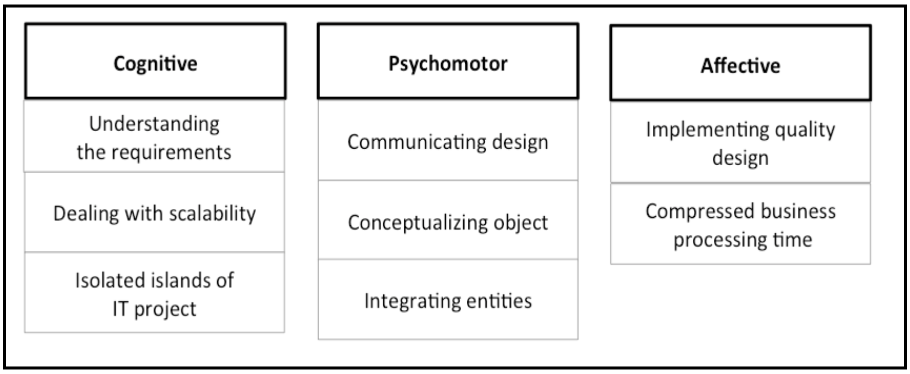

Understanding the requirements
Communicating design
Conceptualizing object
Implementing quality design
Dealing with scalability
Integrating entities
Isolated islands of IT project
Compressed business processing time
IT Architecture Challenges

Figure 1: IT Architecture Challenges.

The activity develops the student social skills, cognitive skills, emotion maturity and self-confidence when they engaged in the new environment and experience. These skills fulfill the pillars in the Bloom taxonomy used in Software Architecture and Design as shown in Figure 2. Cognitive is associated with the knowledge posses by the software architect. Therefore, to understand the client's requirement, the software architect needs to know about the business domain. The solution must be able to scale up as the business grows and the software architect needs to anticipate this using the contributing factors. IT solutions are not operating in silos anymore. Isolated island project need to be integrated. Integrating these solutions is exercising the student's psychomotor. Apart from that, the design produced by the software architect must be communicated using the medium understood by the team and the stakeholders. Affective skill is inculcated through appreciation of quality design and associating the value of the project with time and costs.



Cognitive	Psychomotor	Affective
Understanding the requirements	Communicating design	Implementing quality design
Dealing with scalability	Conceptualizing object	Compressed business processing time
Isolated islands of IT project	Integrating entities	

Figure 2: Mapping of IT Architecture Challenges to the Pillars of Bloom Taxonomy.

3 THE METHODOLOGY AND DESIGN

3.1 The Playing Object, Subject and Environment

In an article on Learning through Play [3], there are many types of play exist to assist in early childhood development as stated in Table I. It gives an overview of types of play exist for the purpose of development and education.

Table 1: Play Type.

Play Type	Explanation
Locomotor play	Involves large body activity and is generally thought to support physical training of muscles, for strength, endurance and skill. e.g.: running, climbing
Social play	Playful interactions between children or with parents. It can be physical and incorporate objects or language. e.g. fighting and chasing
Object play	Playful use of objects which allows children to try out new combinations of actions, free of external constraint and may help develop problem solving skills. e.g. building blocks, jigsaw puzzles
Language play	Use language humorously with repetition and sometimes laughter. e.g. sociodramatic play
Pretend play	Involves pretending an object or an action is something else than it really is. e.g. role taking

Based on the five types of play presented in Table 1, this paper presents the implementation of object play by using building blocks for undergraduate education. Building blocks are chosen to encourage creativity and to develop problem solving skills among the students. The course unit in which object play is deployed is a high level subject matter in the software engineering discipline. The Software Architecture and Design syllabus includes the theory and practical of architecting software system in which mould the direction and deployment of high level and detail software design. It also discusses the skills required for the software architect and designer. The skills are complex to be taught in a lecture format as plain statements and explanation will not give impact to the students. In addition, it is competence that the students need to experience rather than receive one way knowledge delivery through formal lecture. Hence, the playing approach presents in this paper allows the students to experience the skills required to become a software architect and designer.

The subject for the implementation of object play here is the third year undergraduate computer science students from Universiti Teknikal Malaysia Melaka. These students have strong background knowledge both in theory and practical within software engineering discipline. They have gone through two academic years in computer science which specialized in software development. Therefore, the students have sufficient knowledge to conduct effective communication among group members in which the area is familiar to all of them. In addition, the level of maturity of the third year students is sufficient enough to handle healthy reasoning and argumentation during the process. Hence, the deployment of playing method for the purpose of learning serious subject matters will certainly provides rewarding outcomes. This will be further discussed in the discussion of Section 5.

The environment for the learning through playing presents here is a modified classroom setting. The classroom is divided into small clusters and occupied by groups of students. Each group has tables and enough chairs to form an island during the process. They have ample space to work on the building blocks and handle the discussion among group members without interference from others. During the presentation and defend session, the original classroom setting is used.

3.2 The Method and Design

The learning through playing session is set up in a tutorial session of Software Architecture and Design course unit. The students are given two lecture sessions before the tutorial session took place.

Basically, the students have theoretical knowledge background on the software architecture and design.

The tutorial session is simply divided into four stages as stated in Figure 3. In the first stage, instruction and briefing are given to the students. Besides, the expectations are set and explained in order to guide the students to produce likely design from the building blocks. In the second stage, the learning through playing session take place in which the students work in a group to develop one item each group from the given building blocks. In the third stage, each group needs to come forward to present the product they produced and to explain the art of the design to the members of the class. Finally, the instructor concludes the session and provides comments to all the presentations. In addition, the value and learning outcome for the learning through playing session is explained to the students.

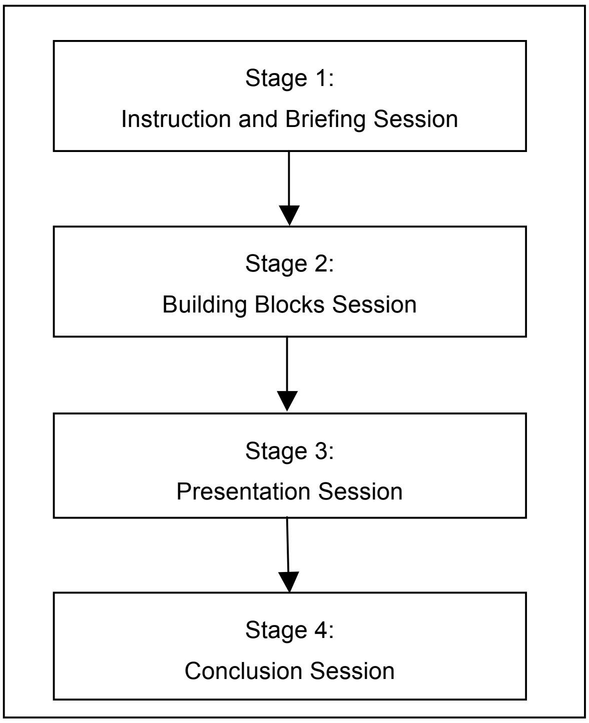


Figure 3: The four stages of implementation design.

3.3 Limitations

The limitation is in the execution as only one course unit which is the Software Architecture and Design took part and reported in this paper. The decision to deploy learning through playing in this course unit is based on the mapping of the curriculum and suitability of the students' knowledge and competence level. In the future, more course unit will be identified to adapt learning through playing in the tutorial session.

4 THE IMPLEMENTATION

The tutorial session took place in the Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka. Three classrooms were used for the tutorial session and each one of them was conducted separately on different time frame. This is due to the scheduled timetable for the respective group of students. The total of ninety students was involved and they were divided into three separate tutorial sessions. In every session, the students were grouped in a team of four or five. Approximately, each session had seven teams. Each team occupied a space in the classroom and was separated from other teams to have ample space for building blocks and discussion. Figure 4 shows the sample products produced during the tutorial session.

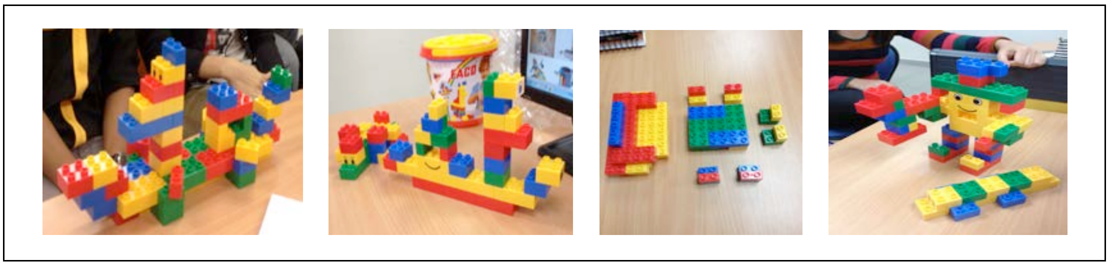


Figure 4: Building blocks sample product.

4.1 The Instruction and Briefing Session

The session started when the instructor gave verbal instructions to the students. The instructor stated that each team was given half an hour to produce a product from the building blocks. The product must have specific functions and produced for a particular purpose. The instructor made it clear that it is important to develop something beneficial to the community or may have a contribution to the body of knowledge.

In addition, the students need to present good design in the product developed by the building blocks. It should exhibit quality features and be the highlight during the presentation. Besides, during the development, the students need to consider design constraints as each team has a limit of 53 pieces blocks with different sizes. They have to use their creativity to optimize the usage of every single block. On top of that, they need to take into account the future maintenance consideration during the development. All the instructions given were in line with the role and responsibility of a software architect and designer as they have to be responsible of the product produced based on the decision they made during the architecting and designing process. The nature of software development was also portrayed as every software project needed to deal with resource constraints and yet still manage to produce functional and meaningful software product.

4.2 Building Blocks Session

After the instruction was delivered, each team was given a tub of 53 building blocks to work with for the next half an hour. When the time starts, most of the team discussed and brainstormed the product they wanted to produce. In order to trigger good idea, it was observed that several teams used the computer to browse the Internet for any possible solution. On the other hand, there were several teams quietly tried to stack and join the blocks, break them and build again several times. It was observed that they tried to reason on every single move they made in a way to build something meaningful. There were also teams that argued with the team members before being able to achieve consensus. During the process, there were a few cases where the team changed their mind and decided to disassemble the blocks towards the end of the time given. They convinced themselves to develop something else instead. It was observed that these teams made such decision due to the uncertainty to justify the product design for the presentation. In conclusion, no matter how they handled the situation, eventually, all teams managed to produce something for the presentation in the time frame given. Figure 5 shows the building blocks activities.

Figure 5: Building Block Session.

4.3 Presentation Session

During the presentation session, surprisingly, the students were cheerful and even volunteered to break the ice. In many years of the instructor’s teaching experience, the situation was odd. Usually Malaysian students are shy and are not comfortable to talk in front of their peers. When the first presenter came up to introduce his product, he was cheered by the entire class. The cheerful environment was created perhaps by the playing session which was different from their common learning process.

The presentation went smoothly for all teams where they have started with description of the product developed, justified the purpose of the design and elaborated the usefulness of the product which contributed to the community as a whole. The presentations also covered the opportunity to reason and argue among the students. The audiences basically enjoyed the question and answer session so much to disagree with, criticize or adding positive information to the presenter. The presenter principally defended their product with the ability of reasoning and positive communication. Figure 6 shows the presentation session.

Figure 6: Presentation Session.

4.4 Conclusion Session

During the conclusion session, the instructor revealed the learning outcomes from the learning through playing effort. Through the process, the students will be able to develop something based on design decisions made by collaborative team members. The students were also trained to be responsible for the design as every single part of the products they develop must have a purpose and hold some quality features. On top of that, they actually developed the ability to make other people see the logic of the design decision and the usefulness of the product through presentation. This leads to the effective two way communication as at the end of the presentation session, the floor was open for debate (question and answer session) to allow healthy argument among the students. The session revealed the maturity among the students to convince others to believe on the products they developed.

Moreover, the instructor provides comments on the playing, development and presentation made by the students. The way of each team doing things and the differences between them were also shared. Besides, the products developed were discussed towards the improvement that could be made in the future. This is important to demonstrate that a software development product comes with maintenance responsibility. That was the reason of maintenance consideration made during the design. In addition, this was important to instill the realization of there are always room of improvement no matter how good the products that have been developed.

5 DISCUSSION AND LESSONS LEARNT

5.1 Discussion

Learning through playing is not new in the early childhood development [5]. As children invest time and energy in play, there are opportunities for development benefits in which the benefits might be immediate, long term or both [3]. Playing method assists in childhood development in variety of ways such as physical coordination, language development, objects manipulation, and emotion recognition.

However, in the delivery of knowledge for undergraduate program in universities, playing method is rather new. There are several efforts to utilize playing method in universities to attract students' attention and at the same time increase the students' involvement in the teaching and learning process. The playing method discussed for the undergraduate students are usually referring to video games which focusing on the role playing [6, 7].

Adapting object play which originally meant for toddlers to recognize and manipulate objects, the same play type is used to develop meaningful product with careful design decision for undergraduate studies. At this level of maturity, the play need to be planned and structured accordingly referring to the teaching plan in order to fulfill the learning outcome for a particular course unit. Based on the deployment of learning through playing done in Software Architecture and Design course unit for Computer Science undergraduate students, it is advantageous to achieve the learning outcome and at the same time improved the learning environment. The students clearly enjoyed the session and the two hours allocation for the tutorial session was cheerfully occupied. The playing method also encourages active discussion among the team members in order to come out with a meaningful product which they need to present and defend to their peers. While having fun, the students still put their priority right as all the teams manage to pull out quality features to be embedded into their design. The purpose of the design and the contribution to the community were also elaborated thoroughly during the presentation. In addition, whenever a question was asked or a criticism was given to the presenter, they manage to handle the situation calmly. In another word, the students manage to practice healthy argumentation and carried out appropriate reasoning. These are important skills for a successful software architect in order to convince organization management to agree upon proposed architecture for a software system.

There is no better architecture or design model than the others because it all depends on the quality features one software project needs to have. Every software architecture model encompasses unique quality features and therefore software architect need to have the ability to recognize the need of the software, prioritize the quality features and design the architecture of the system accordingly. Besides, they also need to have effective communication skills to ensure other people to understand the theory behind the design decision. The skills needed and important for a software architect and designer are all embedded into the design of the tutorial session with playing method. As explained in the previous paragraph, the scenario of software architect who needed to design a product, convinced others and being responsible of the design were portrayed through all the stages of the tutorial sessions. The software product was replaced by building blocks to reduce the pressure of the process but yet, the students still manage to gain the experiences and obtained the learning outcome of the course unit.

5.2 Lessons learnt

This sub section discusses lessons learnt from the tutorial session with playing method. It is divided into four which are encourage active learning, instill cooperative learning, alleviate teaching and learning process and lighten the class environment.

5.2.1 Encourage active learning

In active learning, the responsibility of learning is on the learners. It is in contrast with the conventional method in which the instructor does most of the thing to deliver the knowledge to the students in a one way communication. The playing method evidently promotes active learning as the learners play a vital role of the learning process. Referring to the implementation design in Figure 3, the instructor only responsible to provide the instruction and conclude the session while the rest of the tutorial session was dominated by the students as learners. Playing method using colorful building blocks did catch the students' attention to be involved in the process. This is a deviation from a normal learning process they experienced and clearly enjoyed themselves very much. At the end of the session they were grateful to notice that they actually went through a learning process and obtained the learning outcomes for that particular tutorial session. Therefore, learning through playing really does encourage active learning.

5.2.2 Instill cooperative learning

Cooperative learning is different from a group work in a way that a classroom session is organized into academic and social learning experience. Students learned to capitalize their resources and skills. Meanwhile, the instructor's role changed from giving information to facilitating students' learning. The cooperative learning elements are evidently exist in the playing method presented in this paper as the students were divided into small teams to experience the architect and designer roles and responsibilities. Throughout the process the students need to cooperate with the team members, learn from each other and achieve a consensus to produce something from the building blocks. They also need to articulate the reason of the design decision they made. They engaged themselves with the building blocks to produce meaningful product and being responsible with the design by introducing the product through presentation. This is in line with cooperative learning which facilitated mutual understanding, contribute to team building and develop the ability to learn with others in a safe and engaging environment. The instructor only facilitates the students throughout the process. Therefore, learning through playing really instills cooperative learning.

5.2.3 Alleviate teaching and learning process

Conventional teaching and learning process is usually one way in which the educator (or instructor or lecturer) delivers the knowledge while the learners (students) received it. The conventional way is still effective for information feeding but when it involves a high level of abstraction and a competence based knowledge, one way teaching method is challenging. The students may end up memorize the information given without being able to appreciate the usefulness or even the existence of the knowledge. The playing method which encourage active learning alleviate the teaching and learning process by getting the students directly involved in the process. The students immerse themselves in an activity they enjoyed, experience the knowledge and share with each other while achieving the learning outcome for the course unit. Therefore, learning through playing does alleviate teaching and learning process.

5.2.4 Lighten the class environment

It is a great challenge to maintain high motivation for both educators and learners in two hours tutorial session. This is especially when it takes place in the afternoon. Therefore, learning through playing is a great opportunity to lighten up the class environment. Rather than having a serious session throughout the semester, a high level course unit like Software Architecture and Design can enjoy a playing session while learning. The students were thrilled by only looking at the building blocks and even more excited when they can put their hands on it. They talk to each other, try and error, build and break with high motivation. Unlike usual, even the presentation session look ease on them as they proudly introduce and explain their product. The tutorial sessions were lightened with laugh and high spirit from the beginning until the end of the session. Therefore, learning through playing does lighten the class environment.

6 CONCLUSION

This paper has described an example of learning through playing method using building blocks for the Software Architecture and Design course unit with the purpose of exposing the students to the software architect’s roles and responsibilities. The students are divided into groups and received a set of problems to solve. With team members, every student discusses and analyses the requirements received from the instructor where they need to design solutions for the problem, to consider important factors for the design, to communicate and cooperate with team members to implement the design, to project the solution to the organization or environment and to present the solution designed to other groups in the class. The students need to complete the process within specific time. The students use building blocks as an instrument to demonstrate the method of learning through playing where the blocks are used to design the solutions to the problems. The challenges are not exercising the development process only, but the ability to justify the solution designed, delivering the idea to the audiences and finally to convince them that the solution designed is the best solution and viable to be accepted.

At the end of the learning through playing session, it can be concluded that the students are passionately involved in the active learning environment and enjoying themselves during the process. The cooperation and communication between team members went well since all teams have achieved consensus on the solutions designed and each of them contributed to finalize the development and the presentations. The students are also seemed so involved and enjoyed working in teams, experienced the knowledge and shared ideas while at the same time attaining the learning outcomes of the course unit. The learning through playing method using building blocks can be a good example to develop the students’ social skills, cognitive skills, emotion maturity and the most important is their self confidence.

ACKNOWLEDGEMENT

This paper has been funded by Fundamental Research Grant Scheme, Malaysia with reference code FRGS/2/2013/ICT01/UTEM/02/1.

REFERENCES

[1] Denning, P.J., *Computer Science: The Discipline*, in *Encyclopedia of Computer Science*. 1999, Anthony Ralston & David Hemmindinger (eds.) 2000.

[2] Schon, D.A., Educating the Reflective Practitioner: Toward a New Design for Teaching and Learning in the Professions. 1987, San Francisco, California: Jossey-Bass Inc.

[3] Smith, P. and A. Pellegrini, *Learning through play.* Encyclopedia on Early Childhood Development, 2008.

[4] Hughes, F.P., *Children, Play and Development, 4th Edition*. 2010, USA: SAGE Publications, Inc.

[5] Casey, B.M., et al., The Development of Spatial Skills Through Interventions Involving Block Building Activities. Cognition and Instruction, 2008. **26**(3): p. 269-309.

[6] de Freitas, S., et al., *Hot Issues in Game Enhanced Learning: The GEL Viewpoint.* Procedia Computer Science, 2012. **15**(0): p. 25-31.

[7] Whitton, N., Learning with Digital Games: A Practical Guide to engaging students in higher education. 2010, New York Routledge.

[8] Chickering, A.W. and Z.F. Gamson, *Seven principles for good practice in undergraduate education.* Biochemical Education, 1989. **17**(3): p. 140-141.