

Beginners Learned Competencies in Scratch 2.0 Programming

Jennifer Bagulaya-Abogaa

Associate Professor, College of Computer Studies
Eastern Samar State University, Borongan City

ABSTRACT

The study seeks to identify and describe the learned competencies on scratch 2.0 programming, hence the revision of the course competency into advance skills in scratch programming and the recommendation of implications for the least learned competencies, if there is, be given attention.

Materials used are the learning competency with the corresponding learning outcomes or achieved objectives; the TOS or Table of Specification, a requirement in the formulation of achievement test; and the achievement test. Samples are the two sections of BSBA2 students in the course of Fundamentals of Programming using Scratch 2.0. Using Slovin's formula, a ninety four or 94 samples from the population of BSBA2 were selected randomly. Percentage Frequency Distribution was used in determining the learned competency and least learned competency. The ESSU or Eastern Samar State University is using its own scale of grading system for the passing rate with each adjectival rating.

Based on the result, competency on the stage control presentation of script using scratch, the rating was Good, the least rating among the competencies. The rest of the competencies are Very Good, Excellent, and Outstanding. All-in-all, the average Mean Percentage Score or MPS is ninety one percent or 91%, rated as Excellent.

General Terms

Competency, Hands-on, Programming, Scale, Scratch 2.0 Programming, Slovin, TOS or Table of Specification.

Keywords

Adjectival Rating, Green Flag, Script, Sprite.

1. INTRODUCTION

The know-how in the Philippine Education in secondary and in degree program are itemized from fundamental to advance called competency. Each competency has a corresponding learning outcomes or achieved objectives, then objectives are classified; in cognitive, in psychomotor, and in affective domain. Thus, the former and the latter are the significant entity in the formulation of TOS or Table of Specification, a requirement in the formulation of achievement test. Competency and domain or level of thinking associated by the placement of items, formulates the so-called item analysis.

Learned competencies are achieved from the result of a given achievement test. Programming courses often times difficult to determined the achieved outcomes from a certain competency, because most of the test is in hands-on type.

Results of the test are interpreted on school-based scale, scales are agreed and studied by the faculty members and individual scholars.

From Scale of Competency Level of Quebec; The Scales provides reference points that teachers can use guidelines in

making their end-of-cycle assessment of the levels of competency development attained by the students. While the prime responsibility for this assessment lies with the teacher who has followed and evaluated the student's learning. The scales of competency levels for the competency "Seeks answers or solutions to scientific or technological problems was devised in accordance with the main components of technology program. Each competency level consists of observable indicators that provide a description of the behaviors associated with the level of development achieved by the student at the end of the cycle. From the study of technology, the designed process was favored because it is a rich and effective way of exploring abstract concepts in concrete, hands-on manner. In general the development of this competency requires the students' active involvement. They must ask themselves questions and determine the answers through observations, hands-on activities, measurements, construction or experimentation, be it in a lab, in a workshop or in real world, Scales of Competency Level(Quebec, 2005).

Scratch is an interactive programming language designed to teach children mathematical and computational ideas through creative thinking. Developed at the Massachusetts Institute of Technology (MIT) Media Lab, Scratch was conceived out of the intent to reverse conventional assumptions about the rigid nature of computation in its interactive, visual design. Scratch tool is intended for all level in secondary and undergraduate learners with the goal of learning mathematical and computational ideas through artistic, open-ended play. In Scratch it does not require a high-level coding skill or discourage users who don't have a natural like-mindedness for the sciences in programming. The platform itself functions as blank canvas for creation, which is inviting to all ages, genders, self-perceived skill, and natural talents. With Scratch, conveying realistic life from art no longer denotes understanding of condensed programming language, it allows artistic person to explore and work in the pure art form (Adleberg, 2013).

"Scratch is a visual, block based programming language designed to facilitate media manipulation for novice programmers."

(www.media.mit.edu/~mres/papers/sigcse-08.pdf)

"A programming language that makes it easy to create your own interactive stories, animations, games, music, and art -- and share your creations on the web"
(http://info.Scratch.mit.edu/About_Scratch)

The name was derived from the scratching technique used by hip-hop disc jockeys that spin vinyl records back and forth with their hands to mix music clips together in creative ways. This is similar to what can be achieved with Scratch, mixing media in different ways to create something new. The

program was launched in 2007 and Scratch is designed to develop programming skills and concepts for those with little or no computer programming experience so they could create their own interactive games, stories animations and simulations as well as share them with other Scratch users. Scratch was created by the Massachusetts Institute of Technology (MIT) Media Laboratory, (Kendal, 2014).

2. OBJECTIVES

The study seeks to identify and describe the learned competencies on scratch 2.0 programming, hence the revision of the course competency into advance skills in scratch programming and the recommendation of implications for the least learned competencies, if there is, be given attention.

3. MATERIALS AND METHODS

3.1 Achievement Test

Name: _____

Student Number _____

Direction. Construct a block of scripts on “Catching Cherries”.

Title: Catching Cherries: A Game Project in Scratch (100 points in 30 minutes time duration).

Cherries appear at random horizontal positions at the top of the Stage at random times and fall to the ground. The player has to move the big bowl to catch the cherries before they touch the ground, and each cherry is 1 point score. Set the repeat to 30, meaning there are 30 cherries to win this game.

3.2 Score Sheets

Table 1. Score Sheets of the Hands-on Test

Learning Outcomes/Competency	Number of Items	Score	Percentage Earned (Num Items/ Score) x 100
1. Identify the function of common tools in scratch user interface. Scratch programming environment. Three panes: stage, sprite list, and scripts tab.	5		
2. Function of different block tabs which divided into 10 palettes: motion, looks, sounds, pen, data, sensing, events, control, operators, and more blocks.	5		
3. On the stage controls: (1) Sprite moves, interact, and draw; (2) Show the status of the project; (2) Presentation mode (hide the tools); (3) The green flag to start, and red circle to stop; (4) Add background image called a backdrop.	10		
4. Dragging blocks from the block tabs and snapping them together. Disassemble stack of blocks to test individual block. Detach a block in the middle of the stack, grab and drag.	20		
5. Play sounds, like missile hits or misses the target. Add sounds from MP3 and from other media. Edit the effect of the sounds like fade in or fade out sounds.	10		
6. On paint editor: Create or edit the background or backdrop; create artistic and stylish costume, and background, like bleacher in concert hall.	15		
7. Sprite’s cloning and rotation style, and the power of repeat blocks.	20		
8. Procedure using message broadcast and receiving message, and using variables and game scoring.	15		
Total	100		

3.3 Table of Specification

Table 2. Table of Specification in Introduction to Programming Using Scratch 2.0

Competency	Item Placement (Level of Thinking)			TOTAL Total
	Knowledge	Application	Evaluation	
1 Identify the function of common tools in scratch user interface. Scratch programming environment. Three panes: stage, sprite list, and scripts tab.	5			5
2 Function of different block tabs which divided into 10 palettes: motion, looks, sounds, pen, data, sensing, events, control, operators, and more blocks.	5			5
3 On the stage controls: (1) Sprite moves, interact, and draw; (2) Show the status of the project; (2) Presentation mode (hide the tools); (3) The green		5	5	10

	flag to start, and red circle to stop; (4) Add background image called a backdrop.				
4	Dragging blocks from the block tabs and snapping them together. Disassemble stack of blocks to test individual block. Detach a block in the middle of the stack, grab and drag.		5	15	20
5	Play sounds, like missile hits or misses the target. Add sounds from MP3 and from other media. Edit the effect of the sounds like fade in or fade out sounds.		5	5	10
6	On paint editor: Create or edit the background or backdrop; create artistic and stylish costume, and background, like bleacher in concert hall.	5	5	5	15
7	Sprite's cloning and rotation style, and the power of repeat blocks.		10	10	20
8	Procedure using message broadcast and receiving message, and using variables and game scoring.		5	10	15
Total Items					100

3.4 Sampling Design

Two sections of BSBA2 students in the course of Fundamentals in Introduction to Programming using Scratch 2.0. Using Slovin's formula, a ninety four or 94 samples from the population of BSBA2 were selected randomly. They are beginners in programming.

3.5 Statistical Design

Percentage Frequency Distribution was used in determining the learned competency and least learned competency. ESSU or Eastern Samar State University is using the following scale

of Grading System for the passing rate; with the corresponding adjectival rating, hence, the researcher adapted the grading system.

Actual Rating		Adjectival Rating	
100%	-	95%	Outstanding
94%	-	90%	Excellent
89%	-	85%	Very Good
84%	-	80%	Good
79%	-	75%	Fair
74%	-	below	Failed

4. RESULT

Learning Outcomes/Competency	Total Score=Samples * Number of Items	MPS	Adjectival Rating
1. Identify the function of common tools in scratch user interface. Scratch programming environment. Three panes: stage, sprite list, and scripts tab.	94 x 5 = 470	97	Outstanding
2. Function of different block tabs which divided into 10 palettes: motion, looks, sounds, pen, data, sensing, events, control, operators, and more blocks.	94 x 5 = 470	85	Very Good
3. On the stage controls: (1) Sprite moves, interact, and draw; (2) Show the status of the project; (2) Presentation mode (hide the tools); (3) The green flag to start, and red circle to stop; (4) Add background image called a backdrop.	94 x 10 = 940	84	Good
4. Dragging blocks from the block tabs and snapping them together. Disassemble stack of blocks to test individual block. Detach a block in the middle of the stack, grab and drag.	94 x 20 = 1880	98	Outstanding
5. Play sounds, like missile hits or misses the target. Add sounds from MP3 and from other media. Edit the effect of the sounds like fade in or fade out sounds.	94 x 10 = 940	87	Very Good
6. On paint editor: Create or edit the background or backdrop; create artistic and stylish costume, and background, like bleacher in concert hall.	94 x 15 = 1410	92	Excellent
7. Sprite's cloning and rotation style, and the power of repeat blocks.	94 x 20 = 1880	89	Very Good
8. Procedure using message broadcast and receiving message, and using variables and game scoring.	94 x 15 = 1410	95	Outstanding
Total		91	EXCELLENT

Competency number 3 which is on the stage control presentation of script using scratch, the rating was Good, the least rating among the competencies. The other competency are Very Good, Excellent, and Outstanding. All-in-all, the average Mean Percentage Score or MPS is ninety one percent or 91%, rated as Excellent.

5. CONCLUSION

From the result; BSBA2 students are performing excellent in the beginners' competencies in scratch 2.0 programming, and only one least learned competency. Programming is not highly intense to learn as the generation goes on, because computer application are available in any platform in free and in minimal subscription, so the new generation can easily grab and explore by their own. Higher learning objectives and advance competency on scratch programming must be the subject of improvement and revision of the course syllabi in the programming course to BSBA2 students. The course title must also be revise in accordance to advance learning competence of the students. However, the least leaned competency must have an add-on enrichment.

Developing course competencies in some courses, especially in the discipline of computing, advance skills must be consider due to the advance learning and interest of the learners. Realistic programming competencies should be introduce to the students in the undergraduate or degree programs.

6. FIGURE

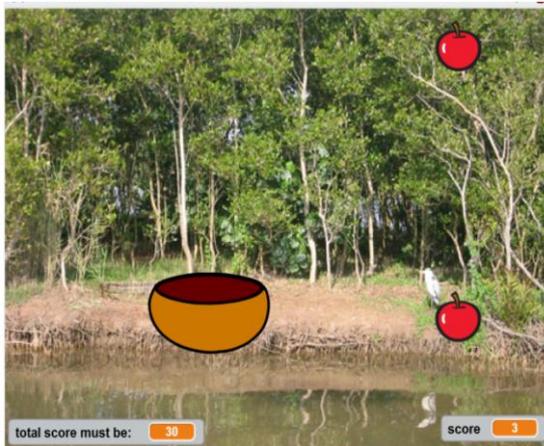


Fig.1 Screen Interface of the Game

7. REFERENCES

- [1] Abogaa, Jennifer B. & Capada, Rowena P. (2016), "Development of Computer-Based Examination as LMS or Learning Management System", ISSN 2454-1632, Volume3, Issue1, <http://www.onlinejournal.in/IJIR>.
- [2] Abogaa, Jennifer B. (2016), "Performance of Freshmen Students in Interactive Multimedia Courseware Using Toolbook Instructor 9.5: ICT Courseware for Specific Purpose", ISSN 2454-1632, Volume3, Issue1, <http://www.onlinejournal.in/IJIR>.
- [3] Abogaa, Jennifer B. (2016), "Customization of an Online Course Examination as LMS or Learning Management System", Volume 7, Issue 12, December-2016, ISSN 2229-5518. <http://www.ijser.org>.
- [4] Adleberg, Barrie M. (2013). Scratch Programming and Remix Culture: Gender Differences in Interaction and Motivation for Pre-Adolescents. Georgetown University, Washington, D.C.
- [5] Kafai, Y. B. (2006). Playing and making games for learning: Instructionist and constructionist perspectives for game studies. *Games and Culture*, 1(1), 34–40.
- [6] Malan, D., & Leitner, H. (2007). Scratch for budding computer scientists. In 38th sigse technical symposium on computer science education (Vol. 391, pp. 223–227).
- [7] Maloney, J., Peppler, K., Kafai, Y., Resnick, M., & Rusk, N. (2008). Programming by Choice: Urban Youth Learning Programming with Scratch. SIGCSE conference, Portland, March 2008.
- [8] Monroy-Hernandez, A., & Resnick, M. (2008). Empowering Kids to Create and Share Programmable Media. *Interactions*, vol. 15, no. 2, pp. 50-53.
- [9] Scales of Competency Levels, Secondary School Education, Cycle One. Quebec, 2005.
- [10] Resnick, M. (2012). Mother's Day, Warrior Cats, and Digital Fluency: Stories from the Scratch Online Community. Proceedings of the Constructionism 2012 conference. Athens, Greece.
- [11] Zuckerman, O, Blau, I, and Monroy-Hernández, A. Children's Participation Patterns in Online Communities: An Analysis of Israeli Learners in the Scratch Online Community. *Interdisciplinary Journal of E-Learning and Learning*. Objects 5,263-274.