Yang, J. C. et al. (Eds.) (2018). Proceedings of the 26th International Conference on Computers in Education. Philippines: Asia-Pacific Society for Computers in Education

Design of a Game-Based Intelligent Learning Environment for Elementary Fractions

Candy Joyce ESPULGAR, Jarl Brent OBEDOZA, Jazmine Veneth SACE, Ryan Samuel DIMAUNAHAN & Raymund SISON*
College of Computer Studies, De La Salle University, Philippines
**raymund.sison@delasalle.ph*

Abstract: A game-based intelligent learning environment (GILE) is a system that combines intelligent tutoring system elements with game elements to provide an adaptive as well as engaging learning experience. A suitable application of GILEs is in promoting number sense, which is the ability to interact with numbers flexibly and conceptually. By having number sense, students come to understand that numbers are meaningful and outcomes are sensible and expected. In this study, the level of engagement of Grade 5 students with a GILE for fraction numbers sense was investigated with the use of a modified GameFlow model. The results can be used to improve the GILE to better promote conceptual understanding of fractions.

Keywords: Game-based intelligent learning environment, game-based learning, educational games for fractions

1. Introduction

A game-based intelligent learning environment (GILE) is a system that combines intelligent tutoring system elements with game elements to provide an adaptive as well as engaging learning experience. Like intelligent interactive tutors (Woolf, 2008), a GILE has a learner modeling component and a pedagogical component. The former transforms values of game variables such as player score into indicators of a player's knowledge of concepts or procedures. The latter adjusts game difficulty levels based on the learner model. A GILE's game component can be viewed as the system's communication component in which the learner and the system interact.

A suitable application of GILEs is mathematics, being one of those subjects with an essential need for multiple pathways and individualized learning (Devlin, 2011, p. 29). One of the best stages to build a solid mathematical conceptual understanding is in Grade 4. This is because Grade 4 is the stage when children start to build the foundation for more complex math. At this stage, children start to apply topics that were taught in Grades 1-3 (such as addition, subtraction, multiplication, and division). They start to multiply and divide larger numbers, add and subtract fractions, and express fractions in reduced forms.

In the rest of this paper, we describe a GILE for fraction number sense, in the process discussing related game mechanics, after which we present some preliminary results.

2. Game Design

Discord is a GILE that aims to promote conceptual understanding of Grade 4-level fraction number sense. The GILE was developed following an outcome-based methodology (Sison, et al., 2018), in which special game mechanics are designed for each learning outcome. The GILE covers the following learning outcomes: (1) Add and subtract similar fractions, (2) add and subtract equivalent fractions, and (3) add and subtract dissimilar fractions.

The system is composed of a pedagogical component that applies a problem-based learning approach, a game component that utilizes game mechanics based on sound pedagogies, and a learner modeling component that interprets user actions and updates the learner model accordingly. The learner model is represented as a Bayesian network. The system is able to accept multiple ways to solve a given problem, record the player's progress and learning state, present challenges within the range of the player's skills, and engage the player to continue playing.

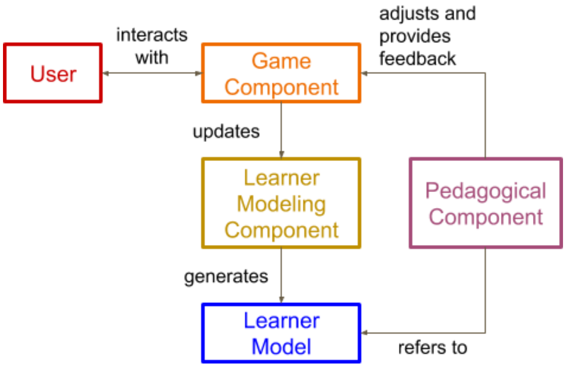

Figure 1. Relationship of GILE Components

Figure 1 shows the overall architectural design. The game component is what users directly interact with. Certain actions in the game component are tallied to get the user's score. These scores are used by the learner modeling component to generate or update the user's learner model. The challenges encountered in the game component are adjusted by the pedagogical component which uses the learner model as a basis on how difficult the challenges should be.

2.1 Game Component

Discord is a 2D puzzle platformer. It provides a series of locked rooms that the player needs to stabilize in order to progress in the game. A room's *current* (Figure 2, A) and *target stability* (Figure 2, B) values are shown on a number line on top of the screen. To stabilize a room, the player must move the *current stability pointer* to the *target stability pointer*. This is done by fixing the right amount of *ghost blocks* (Figure 2, C) in the room. *Ghost blocks* are semi-transparent, intangible blocks of various fractional sizes (e.g. 1/2, 1/4, 1/3). A *ghost block* remains intangible until *sky fragments* (Figure 2, E) equal to its fractional size are attached to it. The yellow circle above a *ghost block* indicates its current value, while the fraction at its center indicates its target value.

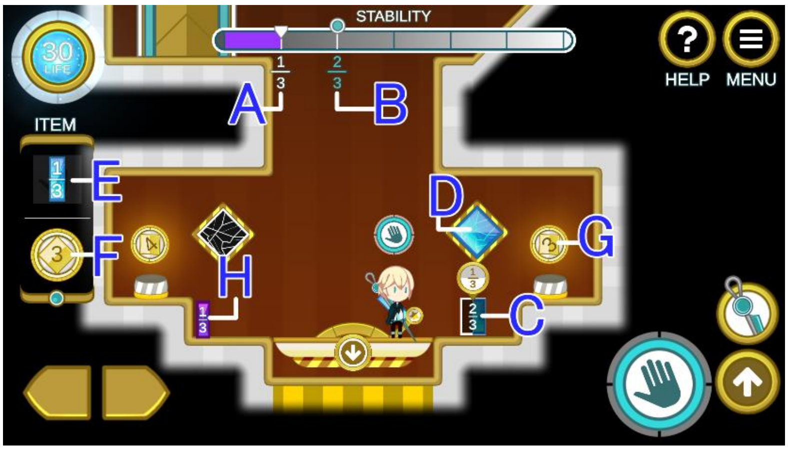

Figure 2. Game Elements

Sky fragments come from slicing *sky blocks* (Figure 2, D) using the *needle* weapon. *Sky blocks* are blue blocks that are used to represent one whole. The number of slices made to a *sky block* is determined by the value of a *charm* (Figure 2, F, G) attached to the needle. *Charms* have integer values (ranging from 2 to 9) and are used to represent denominators, since one *sky fragment* that a *sky block* produces is of the size *1/charm*. A room may have multiple *charms*, allowing the player to choose which of the *charm* values they see fit.

The player can then carry a *sky fragment* and drop it near a ghost block to attach it. If the player makes a mistake in attaching a fragment, it can be removed using the *hammer* weapon. Hitting an attached *sky fragment* will release it from the *ghost block*, while hitting a detached *sky fragment* will return it to the *sky block* it came from and will also allow the player to slice it further. The *ghost block* will immediately be fixed once the fractional value of the *sky fragments* attached to it equate to its fractional value. Once fixed, it becomes a tangible *filled block* (Figure 2, H) and its fractional value is added to the room's current stability. Fixing the right number of objects in a room will stabilize it and allow the player to proceed to the next room. The goal of the game is to fix enough rooms to encounter and defeat the Ringleader, which is the boss character of the game.

2.1.1 Fraction Pedagogy in Game Elements and Mechanics

Sky fragments are used to visualize fractions as part of a whole where a *sky fragment* represents a unit piece of the *sky block* it came from. Through visualizing fractions this way, the player gets a grasp of how much a unit fraction, a fraction whose numerator is 1, is in reference to one whole (Lamon, 2012). This also shows how slicing a whole with a higher number (denominator) produces pieces of a smaller size and allows students to grasp the concept of unit fractions (e.g. 1/3 is larger than 1/6). The presence of multiple *charms* (denominators) allows for estimation since players can visually compare the sizes of sky fragments (unit fractions) produced by different *charms* and estimate which pieces can be used to fill a *ghost block*. This follows Fazio & Siegler (2011)'s suggestion of answering through estimation before using a formal algorithm.

The ability to subdivide a *sky fragment* by returning it to the *sky block* using the hammer and slicing it again using the needle follows the concept of fractions as measures (Lamon, 2012). This concept focuses on the use of successive partitioning to show that a unit can be further divided into smaller parts with smaller values. For example, slicing a *sky block* with a 2 *charm* will produce two 1/2 pieces, which can't be used to fill a 3/4 *ghost block*. However, dividing one of the 1/2 pieces with a 2 *charm* will produce a 1/4 piece. One 1/2 piece and one 1/4 piece can then be used to fill the 3/4 *ghost block*.

The addition and subtraction of fractions are mainly expressed using the room's stability number line. This occurs when the player decides which *ghost block* values need to be combined in order to move the current stability to the target stability. Blocks may need to be filled (addition) or broken (subtraction) to reach the target stability. Through this, players can see how some fractions, despite being expressed with different denominators (i.e. 1/2 and 2/4), have the same underlying value when placed in the number line. This follows the principle of fractions as numbers with magnitude (Lamon, 2012).

2.1.2 Comparison of Mechanics to Successful Learning Games

The slicing, filling, and breaking of blocks can be likened to *Slice Fractions*' (Cyr, Riopel, & Charland, 2015) mechanic, in which players clear the path from obstacles by slicing an object accordingly so that its sliced fractional parts hit the appropriate obstacle of the same fractional size. In Discord's case, the blocks are not destroyed but, rather, their collective states (*filled* or *ghost*) contribute to a room's stability.

The stability number line is similar to *Motion Math: Fractions*' (Riconscente, 2013) mechanic, in which players guide a ball towards a specified point on the number line. The ball is labeled with a fraction and players must tilt the device in order for it to land on the value's proper place on the number line. The difference is in the controls; players tilt their device in *Motion Math: Fractions* to control the falling ball, while in Discord players use the *ghost blocks* and *filled blocks* to manipulate the pointer's position. Number line segments only appear when players make a mistake in *Motion Math: Fractions* while they are already present on the number line in Discord. Beyond that, the two mechanics are similar with respect to mechanic goals.

2.2 Learner Modeling Component

The learner modeling component is responsible for updating the learner model to keep track of the student's performance. Block fixing and breaking is observed by the learner modeling component's agent in the game component (called the *data recorder*). These actions are tallied and their scores (calculated by the pedagogical component) passed to the learner modeling component to update the learner model.

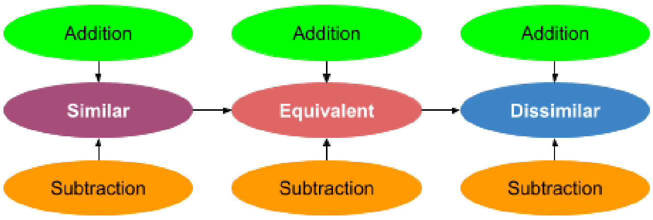

Figure 3. Learner Model as a Bayesian Network.

A simplified Bayesian network is used to model the student's learning. It is composed of 3 unobserved nodes, each representing a topic in fractions, namely (1) operations on similar fractions, (2) operations on equivalent fractions, and (3) operations on dissimilar fractions. The observed nodes are the addition and subtraction of these topics and are directly associated with the player's actions in the game component (i.e. filling a *ghost block* is addition, and breaking a *filled block* is subtraction). Observed node probabilities use the scores given by the pedagogical component. Probability calculation and propagation is done using Chong (2018)'s monadic probabilistic programming framework.

Apart from actions, the learner modeling component also makes use of the denominators involved in solving the rooms. A set of 6 numbers (4, 2, 3, 6, 8, 9) are predetermined to be given to the player. These were decided so fractions produced can result into equivalent fractions and to limit the number of the denominators to keep track. Each denominator has a corresponding mastery score represented as a rate from 0% to 100%. An arbitrary threshold score of 70% is used to determine if the player is fit to encounter the next denominator, which is the same threshold probability for the topic mastery. The denominators are presented sequentially but more than one denominator may be encountered in rooms on equivalent and dissimilar fraction operations. Once it has been inferred that the current topic's mastery probability and current denominator mastery has broken the threshold, the player is deemed fit to proceed to the next topic.

2.3 Pedagogical Component

The pedagogical component sets the values of a room's *ghost blocks* and number line according to the learner model and generates valid numerators and denominators to be given to the game component. It applies a problem-based learning approach by allowing the player to explore and solve newly generated rooms on their own using the knowledge acquired from the previous rooms and the tutorial stages. Before the room's block and number line values are generated, denominator mastery and topic probability mastery are checked in the learner modeling component and then sent to the pedagogical component's agent in the game component. After this, the room values are set up accordingly.

The topic is referenced to know if the problem is to be addition or subtraction and if the *ghost blocks*/*filled blocks* are to be similar, equivalent, or dissimilar. It should be noted that the pedagogical component adjusts the difficulty through the fraction values encountered by the player and the operation to perform (i.e. to break or fill blocks) only. The pedagogical component calculates the player performance score by taking the number of correct attempts (an attempt is either filling or breaking blocks) and dividing it by the total number of attempts. To determine if an attempt is right or wrong, the current values of the *ghost* and *filled blocks* are checked against the subset sum. The subset sum contains all combinations of block values in the room that can equate to the target stability. If the values match an entry, the attempt is marked as correct.

3. Preliminary Results

The GILE was playtested by twenty Grade 5 De La Salle University Integrated School students. Playtesting was done within 45 minutes for 2 days. Playtesters were then asked to answer a modified GameFlow model to evaluate the GILE in terms of engagement. The said model is based on Sweetser et al. (2017)'s model, with the addition of Fu et al. (2009)'s knowledge improvement element, which was added because the GILE aims to promote conceptual understanding in fractions.

The GILE received an overall rating of ~4.33 (~87%) which can be classified as above average. Clear goals had the highest average score of ~4.59, which may be due to the clarity of the tutorials. On the other hand, immersion had the lowest score, with a score of ~4.04. This may have been due to the limited time, as playtesters were asked to stop playing after 45 minutes, which may have affected their immersion. The negative correlation between the highest score (clear goals) as compared to the lowest score (immersion) may also be due to the specificity of the instructions. As some instructions explicitly state fractional values, it may have been viewed as unnatural, hence breaking immersion.

Table 1

Playtesting results from Grade 5 students

GameFlow Element	Section A	Section B	Section C	Section D	Average
Concentration	4	4.0938	4.0357	4.7	4.2074
Challenge	4.20	4.4	4.5714	4.32	4.3729
Player Skills	3.40	4.25	4.1714	4.6	4.1054
Control	4.25	4.4063	4.4286	4.7	4.4462
Clear Goals	5	4.4375	4.4643	4.45	4.5879
Feedback	5	4.1563	4.2857	4.55	4.4980
Immersion	3	3.9688	4.2857	4.9	4.0386
Knowledge Improvement	4	4.2571	4.3429	4.8	4.37
Overall Score	4.1063	4.2462	4.3232	4.6375	4.3283

4. Conclusion

A modified GameFlow model (Fu et al., 2009) was used to identify strengths and weaknesses of the GILE especially in its game component. Preliminary results show that the GILE has potential for engagement but has yet to be assessed with regards to its ability to promote conceptual understanding. The GILE is being improved based on the given feedback and the mechanics are undergoing continuous improvement and revision to better promote conceptual understanding.

References

Chong, J. (2018). C# Probability Monad. GitHub. Retrieved from https://github.com/joashc/csharp-probability-monad.

Cyr, S., Riopel, M., & Charland, P. (2016, July 4-6). Game Design of a Video Game on Fractions Concept Learning in Elementary School Students. Paper presented at the 8th International Conference on Education and New Learning Technologies. Retrieved July 24, 2016, from IATED.

Devlin, K. (2011). *Mathematics Education for a New Era: Video Games as a Medium for Learning*. Natick, MA: A K Peters, Ltd.

Fazio, L., & Siegler, R. S. (2011). Teaching fractions. International Academy of Education.

Fu, F. L., Su, R. C., & Yu, S. C. (2009). EGameFlow: A scale to measure learners' enjoyment of e-learning games. *Computers & Education*, *52*(1), 101-112.

Lamon, S. J. (2012). Teaching fractions and ratios for understanding: Essential content knowledge and instructional strategies for teachers. Routledge.

Riconscente, M. (2013). Results From A Controlled Study of the iPad Fractions Game Motion Math. Retrieved from https://static1.squarespace.com/static/56b55d7ad210b8dc59250285/t/56bd637f55598660590a7027/1455252354801/Riconscente+2013+Games+and+Culture+iPad+study+w+current+contact+info.pdf.

Sison, R., Espulgar, C., Madrigal., Obedoza, J., Romblon, N., Sace, J. & Si, M. (2018). Toward an Outcome-Based Methodology for Game-Based Learning Environments. Proceedings of the 26th International Conference on Computers in Education.

Sweetser, P., Johnson, D., Wyeth, P., Anwar, A., Meng, Y., & Ozdowska, A. (2017). GameFlow in different game genres and platforms. *Computers in Entertainment (CIE)*, *15*(3), 1.

Woolf, B. P. (2008). *Building Intelligent Interactive Tutors: Student-Centered Strategies for Revolutionizing E-Learning.* Burlington, MA: Morgan Kaufmann Publishers.