So, H. J. et al. (Eds.) (2020). Proceedings of the 28th International Conference on Computers in Education. Asia-Pacific Society for Computers in Education

Computational thinking and collaborative storytelling in learning Python

Chien-Sing LEE[a], Nandakumara VEGAN[b], Madhavan BALAN NAIR[b], James Ean-Huat OOI[b]
[a]*School of Science and Technology, Sunway University, Malaysia*
[b]*Lee Kong Chian Faculty of Engineering and Science, Universiti Tunku Abdul Rahman, Malaysia*
*chiensingl@sunway.edu.my, madhavan@utar.edu.my

Abstract: Creative and computational thinking are different but complementary. Combining them would bring significant benefits for young people. With interdisciplinarity in mind, FunPlay Code aims to bridge STEM disciplines with the humanities, and the Arts by combining collaborative storytelling and programming. Modelled after Scratch, FunPlay Code encourages users to express experiences in Python codes. These computational concepts/perspectives can be shared, commented, liked, modded collaboratively in a story format. A search function to enable filtering of stories further caters to users' interests. Functions are developed based on Feature Driven Development methodology. We investigate whether FunPlay Code would be perceived to be easy to use, useful and the likelihood of technology acceptance. User acceptance testing is done remotely with five participants due to the country's covid-19 Movement Control Order (MCO)/lockdown. Findings are relatively positive. The highest mean score is for social interaction/collaborative storytelling, possibly because the story is fun/surprising yet academic.

Keywords: design thinking, computational thinking, collaborative storytelling, Python, modding

1. Introduction

Universal efforts are put in place to develop policies for secondary and tertiary level education and to develop emerging research priorities in STEM disciplines. Researchers such as Freeman, Marginson, and Tytler (2019), however, note that many young individuals lack the motivation to carry out scientific inquiry during learning as it requires the use of models and processes which to some, can be rigorous and tedious. Thus, this problem needs to be addressed urgently and systematically to foster creative individuals who can create new and inventive technologies for diverse markets and conditions.

Dym, Agogino, Eris, Frey and Leifer's (2005) propositions for interdisciplinary Engineering Education, focuses on design thinking-based hypothesis formulation/theorizing, towards more viable and systemic human-centered problem-solving and innovation. Such technological narrative should propel society forward in a human-centric impactful manner, compared to a pure programming approach. In parallel with design thinking, computational thinking (Wing, 2006; Brennan & Resnick, 2012) benefits the Arts and humanities through decomposition, pattern recognition, abstraction, and algorithmic thinking. With more practice, computational perspectives and concepts can be reified.

1.1 Problem statement

Schools in Malaysia, a middle-income emerging economy, generally, do not teach programming. Programming is taught only in high school, and even then, often for those in the Science stream. In recent years, however, some children interested in programming would take part in programming classes either informally (online or vendor-based) or formally, such as those organized by the Malaysian Digital Economy Centre (MDeC). Hence, there is a gap between practices in developed countries such as Europe and the US and many emerging economy countries, such as Malaysia.

There are generally two main approaches to computational thinking education. One is non-native (indirectly linked computational thinking-STEM education) whereas the other is native (directly linked). Two exemplary non-native studies to synergize computational thinking-STEM education are generic educational modelling language, e.g. the IMS Learning Design (LD) and Miao, Sodhi, Brouns, Sloep and Koper's (2008) domain-specific modelling (DSM) approach. DSM synergizes pedagogic experts and technical experts. Pedagogic experts develop notations directly relevant to the concepts and rules for each domain. In contrast, technical experts create transformation algorithms, which map the models represented in the pedagogy-specific modelling language into computable LD to create context- and needs-specific authoring tools.

Hoppe and Werneburg (2019) also regard data and process structure abstraction as core to "representational flexibility." However, they mainly focus on mapping simulated models with actual natural phenomena. They utilize "reactive rule-based programming"-type of program analysis to enable learners to begin with situational specifications of action and then further expand these (functional/non-functional requirements) into more standard block-based iterative programs. Their functional/component-based approach allows easier transition between different computational approaches. Consequently, their next study would be towards meta-level programming to complement metacognitive strategies.

More native to computational thinking-STEM education, is Hasan and Biswas' (2017) domain specific modelling language design with heavier foci on pedagogy while enhancing reusability, interoperability, and rich personalized units of learning.

1.2 Project Objectives

Computational thinking draws on ideas fundamental to computing to solve problems in other fields or domains. The essentials in computational thinking are defining and working with multiple layers of abstractions while understanding their relationships. Wing (2006) focuses on decomposition, pattern recognition, abstraction and algorithmic thinking. Brennan and Resnick (2012) however, focus on much younger learners and thus focus on developing computational perspectives, computational concepts and computational practice.

This paper continues from Lee and Wong (2015; 2018), Lee and Jiang (2019) and Lee and Ooi (2019). Lee and Wong (2015) summarizes design thinking-based research in the creative industries and the development of theorizing and dispositions while Lee and Wong (2018) summarizes research in computing and information systems. In Lee and Jiang (2019), findings indicate that the difference between novices and experts are in perspectives and abstraction mediated by multiple media. In Lee and Ooi's (2019) conceptual paper, codes are used to express experiences and these codes are in small blocks, emulating functions. Findings indicate the importance of human factors especially familiarity to the task and the need for more examples in the database. The ensuing prototypes aim at developing and reifying computational perspectives and computational concepts through computational practice.

The Web-based version is reported in the 2020 Computational Thinking Conference (Phuan, Lee & Ooi, 2020). Since it is Web-based, an additional future interest is the analogical portability of computational concepts to Internet of Things (IoT) and blockchain. This paper presents the Android-based version of FunPlay Code. The programming concepts/skills we hope to develop is through the language Python. This is due to popular usage among universities in Malaysia. The objective is to test user acceptance of FunPlayCode's concepts and prototypes. Hypotheses are:

Hypothesis 1: Perceived ease of use of FunPlay Code will positively affect perceived usefulness.
Hypothesis 2: Perceived usefulness of FunPlay Code will positively affect attitude towards using it.
Hypothesis 3: Collaborative storytelling will positively affect user's attitude towards FunPlay Code.

2. Related work

2.1 Importance of developing interest and habit

Simulated models are often used to externalize and reify human ideas. The Learning Sciences, Educational Technology, Artificial Intelligence communities have progressed far with diverse initiatives aimed at inclusiveness and the Learning Sciences at developing interest and habit. Examples are Lee and Wong's (2015; 2018) design thinking-based research in Malaysia and Wong, Chan, Chen, Looi, Chen, Liao, King and Wong's (2020) interest-driven creator (IDC) theory. The latter aims to co-construct a holistic developmental/design framework to guide students to develop their own their learning interests, capability to create, and learning habits. This corresponds with the **interest loop,**

creation loop and habit loop. The "**interest** loop" involves *triggering interest*, *immersing interest*, and *extending interest*. Consistent with creativity theories, foci are on developing curiosity, flow, and meaningfulness. The "**creation** loop" is based on constructionist philosophies, a key tenet in the Learning Sciences. Consequently, interest drives practice and meaningful habits; the "**habit**" loop.

2.2 Collaborative storytelling

Despite their age or upbringing, everyone undoubtedly has a story to tell. It has helped in communication of sporadic thoughts, ideas and experiences in a structured manner, which ultimately leads to the forming of societies capable of creating stories that span generations. To illustrate the effects of collaborative storytelling, Djerassi's (1998) experiment with Renga, a collaborative style of Japanese linked-verse poetry genre, comprises a minimum of two stanzas that alternate between different authors, who do not know each other's identity. Each student contributes a paragraph to his 'science-in-fiction' experiment aimed at forming a short story addressing a scientific ethical conundrum. By masking science as fiction, collaborative storytelling transcends conventional learning tools. Renga's conversational alternating technique, is analogical to extreme programming.

Similarly, to Cao, Klamma and Martini (2008) facts connected with positive user experience can make people retain information easier in a knowledge sharing environment. To bring together different types of thinking, their Personalized Storytelling Environment (PESE) Community of Practice (CoP) allows a user to a) start a story project; b) request other PESE users to join the project; c) search for a story by title or tags; d) search for a story with an algorithm that searches through each user's profile; e) seek an expert's help. To motivate mentoring, a user can be promoted to an expert based on number of useful advice. We find these initiatives interesting and inspiring.

Finnish schools (Tenhunen, 2018), Hahai (Tasso, Gervasi, Locchi, A., Sabbatini (2019) and many countries, motivate students towards problem-solving and storytelling in younger grades, the former as young as grade 1. At such a young age, when imagination is unconstrained, fun and learning-by-exploring, creating and collaborating take on new dimensions.

3. System Design

FunPlay Code (Lee & Jiang, 2019; Lee & Ooi, 2019) attempts to provide a platform for young people to explore complex STEM topics in a creative and fun way through collaborative storytelling and coding. This is reflected by the contributor and creator roles in FunPlay Code. The social element engages and stimulates them to have fun and to collaborate and learn with their peers. Requirements are to:

- design and develop an application that can interpret Python code typed using a smartphone interface and display the output.
- enable multiple users to collaboratively contribute to a single story;
- enable story creators to restrict chapter posting access to select users and him/herself only by being able to manually add contributors (initial design);
- develop a filtering function to search stories by typing keywords that match any title or description;
- design and implement a code editing space with syntax highlighter.

4. Software development methodology

To satisfy the fluctuating change in requirements at different phases of development, flexible methods would be more suitable. The agile methodology for this project is the Feature Driven Development (FDD). Development is incremental and revolves around five processes (Figure 1). Hence, it is suitable for a learning environment which emulates Resnick's (2002) "playground" spirit. Development tools for the project are PyCharm, WebStorm, MySQL 8.0 Command Line Client, Python, Flask, Javascript and React Native.

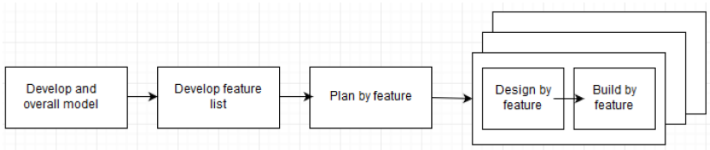


Figure 1. FDD's five processes

4.1 *System development*

Sample screenshots of the developed prototype are presented in Figures 2a, b.

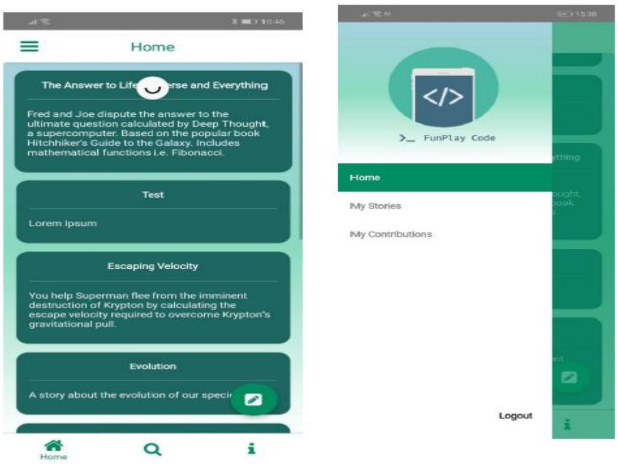


Figure 2a. Home screen

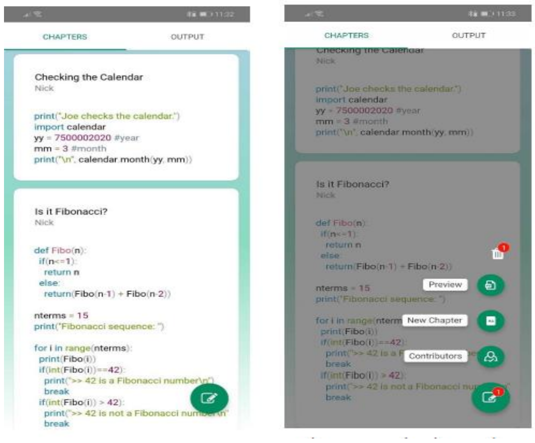


Figure 2b. Chapters and floating action button

5. System testing

This twofold testing takes the form of a software engineering testing and a user acceptance testing. The latter is based on the Technology Acceptance Model (TAM). The story that would be used in the test, to comply with the objective of the project, has elements of Mathematics, namely the Fibonacci sequence and Decimal to Binary conversion. It even contains some relatively complex Python functions to test the robustness of the application as partly a Python interpreter. The print function is used generously to output the results of the computed functions and to display the text that were used to describe events of the story and for dialogue. The story draws inspiration from the successful book that is later adapted into a movie, *Hitchhiker's Guide to the Galaxy* by Douglas Adams.

5.1 *Participants*

Five consenting participants (3 male and 2 female) with ages ranging from 23 to 25 have taken part in this user testing. The small sample size is because the user testing is carried out during the country's covid-19 Movement Control Order/lockdown. These individuals have backgrounds in engineering, physiotherapy, programming and mathematics respectively. Due to the small sample size, our study is a case study, similar to a pilot test.

5.2 User testing methodology

Before the date of the User Acceptance Testing, the server of the application is successfully deployed to Google Cloud, as well as the database which was imported into Cloud SQL to enable transmission of data to all the users concurrently. Since the software has to be tested remotely due to the country's covid-19 Movement Control Order (MCO)/lockdown, a communication line is established with the participants via a WhatsApp group chat.

An APK file containing the actual application with established connection to the server is sent to all the users, ahead of time for the participants to familiarize with the application. They are also politely requested to visit the Info screen for a short tutorial on how to use the app. A copy is also downloaded and installed into the android smartphone of the author to monitor the participants' actions.

The User Acceptance Testing officially begins when the author is notified by each participant of their online presence on the group chat. All the participants are then asked to login with their unique credentials that are previously assigned. One participant is selected at random to be the creator. The rest play their roles as contributors. The designated creator is tasked to create a story and add the other participants as contributors by identifying their usernames from the multi-select dropdown that would appear once the correct sequence of actions is taken by the creator, in a round-robin fashion. They would be given access to the script written on Notion, a collaborative multi-purpose note-taking tool, to retrieve the code for each chapter. Afterwards, the other participants are instructed to pull-to-refresh their Home screens and locate the new story that had been created by identifying the title. They then navigate to the Chapters Screen. The entirety of the testing occurred in one session encompassing all the participants as this application is intended to be used collaboratively.

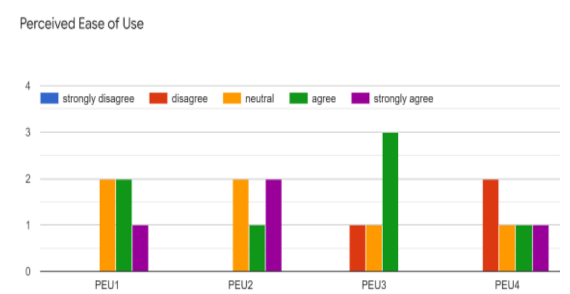
5.3 Technology Acceptance Model instrument/questionnaire

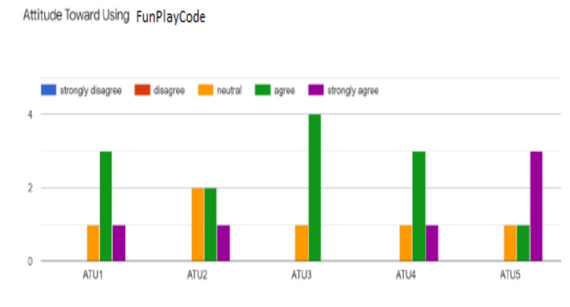
In this project, a questionnaire is utilized to investigate the acceptance of FunPlay Code. The survey items are based on a slightly modified version of the original Technology Acceptance Model (TAM; Davis, 1989) which proposes certain constructs or factors that determine the extent of users' acceptance. This questionnaire includes 4 items for perceived usefulness, 4 items for perceived ease of use, 5 items for attitude toward using, 4 items for intention to use and an additional 3 items for social interaction to validate the social component of this application. The questionnaire is measured in a 5-point Likert scale that ranges from "strongly disagree" to "strongly agree".

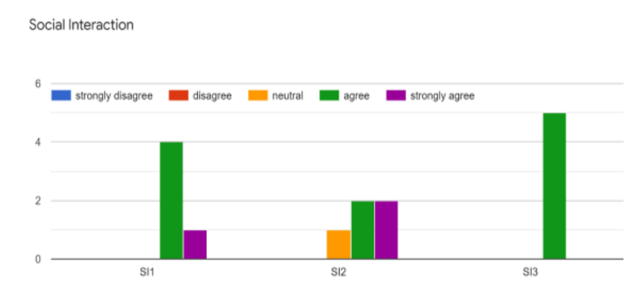
6. User testing results and analysis

The user testing results and analyses are presented in Table 1.

Table 1. *User testing results and analyses*







Factor/ Construct	Questionnaire Item	Mean	Cronbach's α (if item excluded)	Cronbach's α
Perceived Usefulness (PU)	PU1	3.4	0.9355	0.9296
	PU2	3.8	0.9	
	PU3	3.6	0.875	
	PU4	3	0.9124	
		Average = 3.45		
Perceived Ease of Use (PEU)	PEU1	3.8	0.7279	0.7619
	PEU2	4	0.5943	
	PEU3	3.4	0.65	
	PEU4	3.2	0.8421	
		Average = 3.6		
Attitude Towards Using (ATU)	ATU1	4	0.7799	0.8434
	ATU2	3.8	0.8302	
	ATU3	3.8	0.7957	
	ATU4	4	0.7799	
	ATU5	4.4	0.8727	
		Average = 4		
Intention of Use (IU)	IU1	3.4	0.75	0.642
	IU2	4.2	0.7174	
	IU3	2.8	0.1875	
	IU4	4.2	0.3	
		Average = 3.65		
Social Interaction (SI)	SI1	4.2	0.5345	0.6154
	SI2	4.2	0.2857	
	SI3	4	N/A	
		Average = 4.133		

- For all of the constructs, the total average score is over 3.45 which indicates that the users perceive the application quite favourably.
- Social Interaction at 4.13 is the construct with the most positive response. These responses imply that the users are attracted to the storytelling and social aspect the most.
- Among the subitems, the best response with a mean score of 4.4 is for ATU5, i.e. the fifth item in Attitude towards using FunPlay Code, "I like the storytelling aspect of FunPlay Code."
- Cronbach alpha values are positive but sample size is too small.

7. Conclusion

In conclusion, the FunPlayCode collaborative storytelling mobile application using Python code has potential. However, collaborative storytelling in a non-academic manner is attractive to young coders, if the story is fun/surprising. Despite the remote testing, the sharing of screenshots and the Info screen are helpful as guides whenever a participant faces any complications. There are however, limitations to the study e.g. small sample size. Moreover, the learning curve would reduce if there are more examples to build on. Future work may include linking to social media accounts to invite friends to code with.

Acknowledgements

We thank Dr. K. Daniel Wong and Universiti Tunku Abdul Rahman for introducing/supporting design-computational thinking since 2013 when the first author was a Faculty at UTAR, and the anonymous reviewers' detailed feedback. This paper is extended from Nanda's final year project thesis.

References

Brennan, K. and Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *American Educational Research Association*, Vancouver, Canada (*1*, 25).

Cao, Y., Klamma, R. and Martini, A. (2008). September. Collaborative storytelling in the web 2.0. In *International Workshop on Story-Telling and Educational Games (STEG 2008) at ECTEL* (vol. 8).

Djerassi, C. (1998). Ethical discourse by science-in-fiction. *Nature*, *393*(6685), 511.

Dym, C. L., Agogino, A.M., Eris, O., Frey, D. D. and Leifer, L. J. (2005). Engineering design thinking teaching and learning. *Journal of Engineering Education*, *94* (1), 103-120.

Freeman, B., Marginson, S. and Tytler, R. (2019). An international view of STEM education. In Sahin, A. and Schroeder, M. (Eds). Myths and truths: What has years of K-12 STEM education research taught us? *STEM Education 2.0*, 350-363.

Hasan, A., and Biswas, G. (2017). Domain specific modeling language design to support synergistic learning of stem and computational thinking. Kong, S. C. (Ed.) Hong Kong, 28.

Hoppe, H. U., and Werneburg, S. (2019). Computational Thinking—More Than a Variant of Scientific Inquiry! In *Computational Thinking Education* (pp. 13-30). Springer, Singapore.

Lee, C. S. and Wong, K. D. (2015). Developing a disposition for social innovations: an affective-socio-cognitive co-design model. *International Conference on Cognition and Exploratory Learning in Digital Age*, October 24-26, 2015, Ireland, 180-186.

Lee, C. S., Wong, K. D. and Lau, S. B. Y. (2015). Scaffolds and design factors to increase creative outcomes in teaching Software Design and Testing. *IEEE International Conference on Industrial Engineering and Engineering Management*, December 9-12, 2015, Singapore, 451 – 454.

Lee, C. S. and Wong, K. D. (2018). Design - computational thinking, transfer and flavors of reuse: Scaffolds to Information and Data Science for sustainable systems in Smart Cities. *IEEE International Conference on Information Reuse and Integration*, IEEE Computer Society, Salt Lake City, Utah, July 7-9, 2018, 225-228.

Lee, C. S. and Jiang, B. (2019). Assessment of Computational Thinking (CT) in Scratch fractal projects: Towards CT-HCI scaffolds for analogical-fractal thinking. *International Conference on Computer-Supported Education*, May 2-4, 2019, Crete, Greece.

Lee, C. S. and Ooi, E. H. (2019). Identifying design factors to encourage reframing, reuse through granular coding-analogical thinking-storytelling, presented *International Conference on Engineering Technology,* 2019; published Design to encourage reframing and transformations through digital storytelling and analogical thinking, *International Journal of Advanced Trends in Computer Science and Engineering*, *9*(1.3), 2020.

Miao, Y. Sodhi, T., Brouns, F., Sloep, P. and Koper, R. (2008). Bridging the Gap between Practitioners and E-Learning Standards: A Domain-Specific Modeling Approach. *5192*, 284-289.

Resnick, M. (2002). Rethinking learning in the digital age. The global information technology report: Readiness for the networked world. Geoffrey S. Kirkman, Peter K. Cornelius, Jeffrey D. Sachs, Klaus Schwab (Eds.).

Tasso, S., Gervasi, O., Locchi, A., Sabbatini, F. (2019). Hahai: Computational Thinking in Primary Schools, *International Conference on Computational Science and Its Applications*, 287-298.

Tenhunen, M. L. (2018). Reform of the Finnish Education System. Workshop on playfulness in the teaching and learning of STE(A)M, *International Conference on Computers in Education,* November 26-30, 2018. Manila, Philippines.

Wing, J.M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *366*(1881), 3717-3725.

Wong, L. H., Chan, T. W., Chen, W., Looi, C. K., Chen, Z. H., Liao, C. C. Y., King, R. B. and Wong, S. L. (2020). IDC theory: interest and the interest loop. *Research and Practice in Technology Enhanced Learning 15*(3), 2-16.