Iyer, S. et al. (Eds.) (2022). Proceedings of the 30th International Conference on Computers in Education.
Asia-Pacific Society for Computers in Education

PyGuru: A Programming Environment to Facilitate Measurement of Cognitive Engagement

Daevesh SINGH[a*], Hema SUBRAMANIAM[b], Ramkumar RAJENDRAN[a]
[a]*IDPET, Indian Institute of Technology Bombay, India*
[b]*Department of Software Engineering, Universiti Malaya, Malaysia*
*daeveshsingh@iitb.ac.in

Abstract: There is a vital link between students' engagement and their performance. Despite this, a dearth of studies exploring student engagement (especially cognitive) in computer programming, where high failure rates are a norm. Interestingly, online environments capture fine-grained interactions; this data has the potential to detect students' cognitive engagement. The contribution of this paper is two-fold. Firstly, it introduces an online learning environment to learn and practice python programming called PyGuru and the data logged in the system. Secondly, we provide a preliminary analysis of students' interaction with the system to reveal their level of cognitive engagement in different system components. The results of this study conducted for two weeks and 21 students demonstrate that students perform more actions that correspond to passive engagement than other levels of engagement.

Keywords: Programming, Learning Environment, log data, PyGuru, Cognitive Engagement

1. Introduction

The significance of Student Engagement(SE) in learning is well established in the literature (Duderstadt et al., 2002). Empirical studies demonstrate that SE is linked significantly with critical thinking, problem-solving and academic success (Carini et al., 2006; Wang & Holcombe, 2010; Zhao et al., 2021). On the other hand, introductory programming courses suffer a fate of high failure and withdrawal rates (Watson & Li, 2014). Due to the strong association between SE and academic success, there is a keen interest among researchers in identifying SE levels to assist at-risk students.

SE is "*participation in educationally effective practices, both inside and outside the classroom, which leads to a range of measurable outcomes*" (Kuh 2007). In literature, various dimensions of engagement have been defined, including behavioural, affective, cognitive etc. Our work focuses on Cognitive Engagement (CE) as it is not directly observable, making its measurement challenging. There is also a paucity of literature that studies CE in the context of computer programming.

Although some studies focus on CE in online learning in other domains, they mainly focus on only one of the online learning components like video watching, discussion forums, etc. Hence, there is a need to analyse all the actions students' perform in a Computer-Based Learning Environment (CBLE) to understand CE better. This analysis will enable the instructors to identify at-risk students and provide interventions to prevent failure and dropouts.

This study proposes a CBLE for teaching-learning computer programming called PyGuru based on the ICAP framework. PyGuru captures students' interaction in four online components: book-reader, video player, discussion forum, and IDE. This paper provides details about the design and the data captured by the system. Further, we present learners' interaction data to demonstrate how they interact with PyGuru and the different modes of engagement they involve. This study was conducted for two weeks in a public university in Malaysia in the research methods classroom with 21 postgraduate students. The results obtained after analysis of log data reveal that students perform more actions that correspond to lower levels of engagement. This lower engagement could be one of the reasons for lower learning levels.

2. Background and Related Work

CE is the thinking that students get involved in while learning (Helme & Clarke, 2001). As per Fredricks et al. (2004), "*cognitively engaged students would be invested in their learning, would seek to go beyond the requirements, and would relish challenge*" (Fredricks, Blumenfeld, & Paris, 2004). CE is the most misunderstood and ill-defined construct (Greene, 2015) and thus making its operationalisation and measurement difficult and challenging (Chi et al., 2018). To overcome this gap, Chi and Wylie (2014) proposed a hierarchical framework called ICAP to understand CE (Chi & Wylie, 2014). This framework tries to classify students' overt behaviours into different modes of engagement: interactive, constructive, active, and passive.

The passive mode of engagement is when "*learners receive information without overtly doing anything related to learning*" (Chi & Wylie, 2014, p. 221). In this mode, learners attend to the information (without performing any actions like note-taking) and store it in episodic form rather than integrating it with prior knowledge. Active engagement occurs when the learner's information acquisition is accompanied by specific physical or motoric actions that support their learning. This engagement includes taking notes in the classroom, pausing and recapping the videos or highlighting the text while reading. The cognitive process involved during active engagement demands the activation of prior knowledge and integration of the new information into the existing one. Constructive engagement happens when the learners attempt to produce artefacts using information that goes beyond the information available in the environment. Constructive engagement includes "*elaborating, comparing and contrasting, generalising, reflecting on, and explaining how something works*" (Chi & Wylie, 2014, p. 228). Interactive engagement occurs when learners, during interactions with partners, are constructively engaged, and there is sufficient turn-taking. These partners could be peers, instructors or even computer agents. ICAP framework also informs that the highest learning occurs at the interactive level followed by constructive, active, and the least when engagement is passive.

ICAP model was initially developed for classroom learning; however, this framework was later extended to online mode as well. Yogev et al. (2018) examined students' CE in reading material using the Nota Bene annotation platform. They firstly analysed CE anchored in the text by manually labelling students' annotations and then developed an interactive decision tree to automate this process. Similarly, Dodson et al. (2018) developed a framework to classify students' different video-watching behaviours as per the ICAP framework. They classified behaviours as passive(playing video content), active(replay, pause, seek specific information), constructive(taking notes, highlighting), and interactive (cooperating and collaborating with others). The study by Atapattu et al. (2019) tried to automate the process of classifying the post in the discussion forum into active and constructive modes of engagement. The posts that were significantly different from the learning materials were classified as constructive, and the more similar ones were labelled as active.

To summarise, we have presented how ICAP is used in book reader, video-player and discussion forums. These studies are crucial in understanding students' CE in different online learning components. However, understanding students' overall CE is also crucial and to fill this gap, we have designed our system PyGuru, which combines these three components, namely book reader, video-player and discussion forum, along with an IDE. This is done to investigate how overall CE (in book reader, video player etc.) impacts students' performance in Python programming. To understand this, we present PyGuru- a learning environment for learning Python programming capable of logging the user actions.

3. PyGuru: Learning Environment

PyGuru (https://pyguru.personaltutoring.in/) is a computer-based learning environment developed to teach and learn Python programming skills. PyGuru has four components: book reader, video player, code editor, and discussion forum. This section describes each of these components.

The book reader (shown in Fig 1) in PyGuru contains textual information readers can highlight and annotate. The annotating feature in the learning environment comprises selecting a text, commenting on that text, and providing a tag to that text.

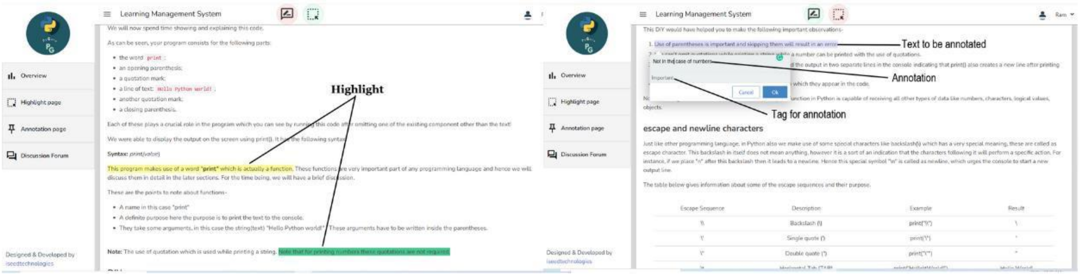

Figure 1. The figure shows the features of the book reader. The highlight feature is shown in the left image. The right image shows the annotation feature.

PyGuru has an interactive video-watching platform (Fig. 2) that allows learners to perform basic actions like enhancing the speed of the video, playing, pausing, and seeking. Along with this, more advanced interactive features are embedded into the system that allows the instructor to add questions within the video. The video automatically stops and waits for the learner's response.

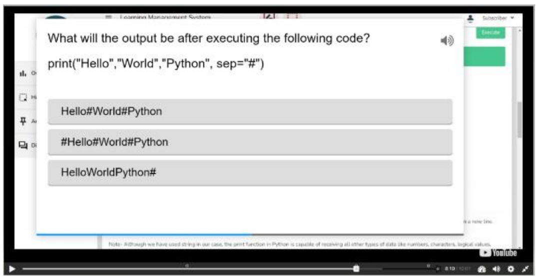

Figure 2. The figure shows the in-video question that appears on the screen. The video player has a panel below that allows learners to adjust the speed and volume. Also, the size of the window can also be adjusted.

Learning programming requires a code editor where learners can practise coding. PyGuru offers two kinds of code editors (Fig 3). The first kind of code editor is embedded into the book reader to facilitate learners to practice codes immediately after learning about the concept. As shown in (Fig 3 left), this code editor has a coding window and an execute button.

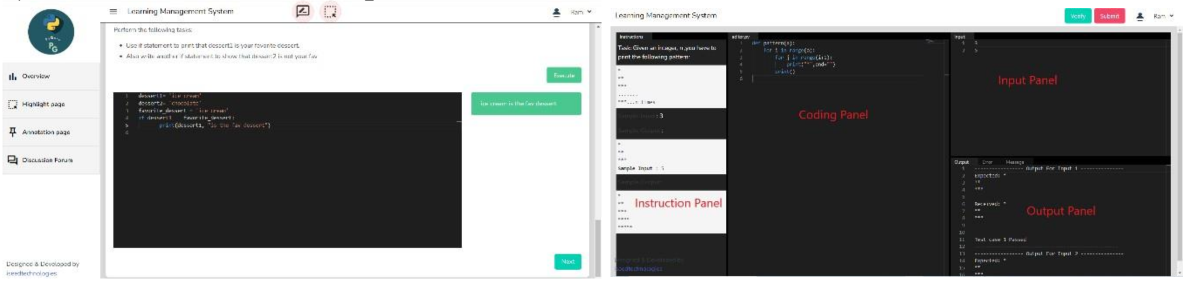

Figure 3. The figure on the left shows the code editor embedded in the book reader. The figure on the right shows the code editor with four panels.

The second kind of code editor shown in (Fig 3, right) is more advanced and used to assign programming questions to students. It evaluates the learners' code against the test cases. This code editor consists of four panels, a) Instruction panel provide details about the problem, b) Input panel contains the test cases for the problem, c) Coding panel is where the student is expected to write the code, d) Output panel displays the output once the program is run. It will also provide information about the number of test cases passed and failed.

PyGuru also has a discussion forum. The learners can use this discussion forum to put forth their queries which the instructor or their peers can then answer. The students can also like the post to show that their question has been answered.

4. Methods

This section describes the methodology. The first subsection informs about the participants. The second subsection provides details about the data captured.

4.1 Participants

The study was conducted with 21 students who used PyGuru to learn Python programming in 2022. These students were enrolled in a Master's program at a public university in Malaysia in the research methods course. Informed consent was obtained from the student, and the study was cleared by the Institute Ethics Committee (IEC). No monetary compensation was given to the students. The course instructor provided students with a demographic survey that involved questions like their name, age, gender etc. at the beginning of the course. The duration of this study was two weeks. During this time, the students interacted with PyGuru. Their interaction with the system (clickstream) and the time stamp was captured (more details about log data are provided in the following subsection). In addition to this, the students took a pre and post-test containing 12 multiple-choice questions along with two program writing questions from Python basics (variables, operators, conditional statements etc.).

4.2 Data

Log data from PyGuru was collected for all the 21 students while they interacted with the system for two weeks. In this subsection, we provide the details of the log data generated in each component of PyGuru. Table 1 provides the details of the interaction data. We will now explain the data presented in table 1.

Table 1. *The table shows the data generated when students interact with PyGuru.*

S_id	Time_Stamp	Action	Context	Context_Significance	Page
82	5/4/22 19:13	Log In			Home_Page
82	5/4/22 19:32	Log Out			
82	5/4/22 12:13	Reading			creating-a-varible
82	6/4/22 16:55	Highlight	\n		writing-first-program
82	6/4/22 17:15	Annotation	escape sequence	Error due to absence of \n	writing-first-program
82	7/4/22 20:11	Execute	success		writing-first-program
82	7/4/22 20:11	Watching			demo-on-print
82	7/4/22 20:19	Played	0:00		demo-on-print
82	7/4/22 20:22	Seek	8:01		demo-on-print
82	7/4/22 20:22	Paused	8:01		demo-on-print
82	7/4/22 20:25	VQ_opt_select	8:01	Correct	demo-on-print
82	9/4/22 10:25	VQ_retry	3:02		demo-on-input
82	10/4/22 8:30	Assessment			problem-on-if-else
82	10/4/22 8:37	Submitted	3		problem-on-if-else
82	11/4/22 13:25	Post viewed			Discussion Forum
82	11/4/22 13:25	Posted			Discussion Forum
82	12/4/22 9:30	Liked			Discussion Forum
82	12/4/22 9:32	Replied			Discussion Forum

The first and second columns provide the unique id of the learners and the timestamp of the action. The third column displays the actions in four different components. In book reader, the actions performed are reading, highlight, annotation and execution. The context column provides details about the text selected for highlight and annotation, and for execution, it informs whether there are errors. The Context_Significance for annotation offers information about the annotated text. The last column, named Page, gives the page's name on which the learner is currently present.

In the video player, the actions performed are played, paused, seek and the context provides the video time at which these actions were performed. Since the videos have in-video questions, the action VQ_opt_select correspond to the selection of one of the options in the in-video question, and the Context_Significance tells whether the correct option was chosen or not. If the option selected is incorrect, learners can retry the question and the action VQ_retry corresponds to this.

As mentioned earlier, PyGuru has two code editors. The second kind of code editor is more advanced and the action "Assessment" corresponds to using or accessing this code editor. The learners

can also click the submit or verify button to check if their code is thriving on the given test cases, and the number provided in the context informs the number of test cases successful.
The learners can view the discussion forum, write a post, like an existing post, and reply to the post in the discussion forum.

5. Results and Discussions

We analysed the log data from PyGuru to understand how learners interact with the system. As one of the main goals for developing this system is to measure the learners' CE. As a result, we looked into more granular data, i.e. what specific actions learners' perform in each component. Figure 4 presents the composition of different actions learners' perform in each component of the system. From these charts, it can be seen that the actions that correspond to lower levels of CE are more prominent. For instance, in book reader, the action of highlight and annotation indicate active and constructive engagement as per ICAP but simply reading without doing anything is passive engagement. We see that reading comprises 90% of the total activities done in book reader, indicating that very learners rarely have higher levels of engagement.

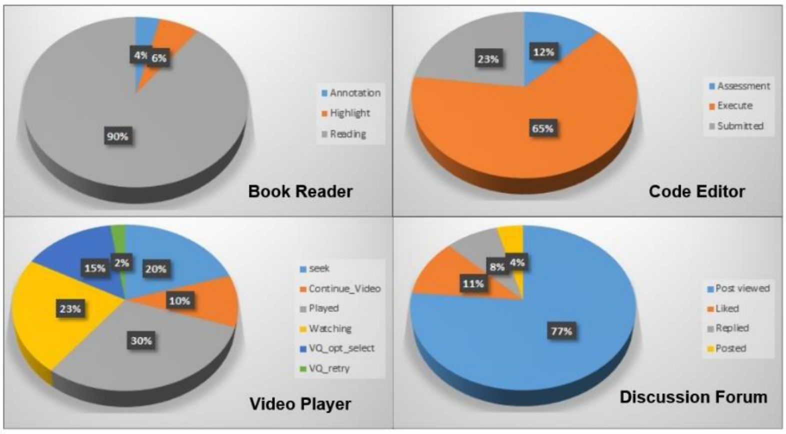

Figure 4. The figure shows the composition of different actions in the four components of PyGuru, namely book reader, video player, discussion forum and code editor.

In video player also, the actions that involve constructive engagement, like answering in-video questions, have a lower frequency than passive watching of videos. Also, the frequency of actions corresponding to active engagement that includes pause and seek behaviour is less. The same behaviour is seen in the case of discussion forum and code editor.

6. Conclusion and Future Work

This paper presents a learning environment called PyGuru for teaching-learning Python programming. PyGuru is based on the ICAP framework and consists of four components: book reader, video player, discussion forum, and IDE. The system captures the click-action of the learners along with the time stamp. This data was analysed to understand student interaction in the system. We found that students mainly interact with the system in passive mode. As suggested from the literature, lower levels of engagement affect students' performance. Hence, it is crucial to either motivate students to engage actively and constructively or design learning activities that promote higher engagement levels.

Since all the click-action of the learners along with the timestamp are collected. This fine-grained data has the potential to capture students' CE using the ICAP framework. Also, this data can be used to provide feedback to the students. For instance, in the case of in-video questions, each incorrect option will provide feedback on why a particular option is incorrect. Similarly, in IDE, learners will be informed about how many test cases are correct. Although the learning environment currently supports all basic functionalities, there are some limitations. For instance, it currently does not have any timer to keep track of time spent solving a programming problem.

In future, we propose to validate these log data by doing correlation analysis with self-report data. We also plan to investigate learning strategies using sequences of behaviours corresponding to higher modes of engagement. For example, what behaviours do students engage in before and after highlighting? Identifying common video-based learning strategies etc. Methods for measuring learning processes and outcomes in relation to ICAP will also be investigated.

References

Atapattu, T., Thilakaratne, M., Vivian, R., & Falkner, K. (2019). Detecting cognitive engagement using word embeddings within an online teacher professional development community. Computers & Education, 140, 103594.

Carini, R. M., Kuh, G. D., & Klein, S. P. (2006). Student engagement and student learning: Testing the linkages*. Research in Higher Education, 47(1), 1–32. https://doi.org/10.1007/s11162-005-8150-9

Chi, M. T., & Wylie, R. (2014). The ICAP framework: Linking cognitive engagement to active learning outcomes. Educational psychologist, 49(4), 219-243.

Chi, M. T., Adams, J., Bogusch, E. B., Bruchok, C., Kang, S., Lancaster, M., ... & Yaghmourian, D. L. (2018). Translating the ICAP theory of cognitive engagement into practice. Cognitive science, 42(6), 1777-1832.

Dodson, S., Roll, I., Fong, M., Yoon, D., Harandi, N. M., & Fels, S. (2018). An active viewing framework for video-based learning. Proceedings of the Fifth Annual ACM Conference on Learning at Scale. New York, NY, USA: ACM. Retrieved from http://dx.doi.org/10.1145/3231644.3231682

Duderstadt, J. J., Atkins, D. E., Van Houweling, D. E., & Van Houweling, D. (2002). Higher education in the digital age: Technology issues and strategies for American colleges and universities. Greenwood Publishing Group.

Fredricks, J. A., Blumenfeld, P. C., & Paris, A. H. (2004). School engagement: Potential of the concept, state of the evidence. Review of Educational Research, 74(1), 59–109. https://doi.org/10.3102/00346543074001059

Gorgun, G., Yildirim-Erbasli, S. N., & Epp, C. D. (2022). Predicting Cognitive Engagement in Online Course Discussion Forums. learning, 53(60), 61-62.

Greene, B. A. (2015). Measuring cognitive engagement with self-report scales: Reflections from over 20 years of research. Educational Psychologist, 50(1), 14-30.

Helme, S., & Clarke, D. (2001). Identifying cognitive engagement in the mathematics classroom. Mathematics Education Research Journal, 13(2), 133-153.

Kuh, G. D. (2007). What student engagement data tell us about college readiness.

Wang, M.-T., & Holcombe, R. (2010). Adolescents' perceptions of school environment, engagement, and academic achievement in middle school. American Educational Research Journal, 47(3), 633–662. https://doi.org/10.3102/0002831209361209

Watson, C., & Li, F. W. B. (2014). Failure rates in introductory programming revisited. Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education - ITiCSE '14. New York, New York, USA: ACM Press. Retrieved from http://dx.doi.org/10.1145/2591708.2591749

Yogev, E., Gal, K., Karger, D., Facciotti, M. T., & Igo, M. (2018). Classifying and visualising students' cognitive engagement in course readings. Proceedings of the Fifth Annual ACM Conference on Learning at Scale. New York, NY, USA: ACM. Retrieved from http://dx.doi.org/10.1145/3231644.3231648

Zhao, Y., Lin, S., Liu, J., Zhang, J., & Yu, Q. (2021). Learning contextual factors, student engagement, and problem-solving skills: A Chinese perspective. Social Behavior and Personality: An International Journal, 49(2), 1–18. https://doi.org/10.2224/sbp.9796