Learning Programming Language with Game-Like Elements Integrated on a Web-Based Platform and Assessment using Formative Feedback

Vivek Kosurkar¹, Abhishek Zade², Akash Tikas³, Sujeet Prasad⁴, Bhushan Sure⁵, Smita kapse⁶

Department of Computer Technology
Yeshwantrao Chavan College of Engineering, Nagpur, India

Abstract:
The proposed system presents a new concept in learning to be implemented on a web-based platform. This concept uses game-like fun elements and an attractive user interface. The aforementioned platform will be helpful in learning programming languages in interesting and stress-free way. This paper reign in the theory of cognitive load, which relates to internal working of mind and our memory by their concepts, this paper creates a design and framework which will help improve programming education drastically. Students and anyone interested can learn the theory and solve the quizzes and challenges in increasing loads of difficulties. They form a basic part of assessment that is they taken progress of student. This paper intend to give rating based on the time and level of questions. Assessment will be formative and in continuous form to better the learning experiences. This unlikely formulation of game-like elements and formative feedback based learning is better than conventional learning. This paper design and implement the framework and different levels of questions in details, this paper provide the theory on Java and its topics and ask users to solve quizzzes and maintain a leader board of rating.

Keywords: learning; cognitive load theory; game-like element programming language, formative feedback, Java; online web-based learning and assessment platform.

I. INTRODUCTION

The difficulty of programming language learning and its practical application can be attributed to the load of new concepts to be learnt (like various syntax) and students perspective towards it (like fear for new things, motivation etc.). Also pre-conceived notions lead to unsuccessful learning. The perception about programming needs to change. It should not be considered as an upheaval task; rather learners must themselves show interest and have high morale to learn it. This should be undertaken by the process of learning which should implement enjoyable experience of learning to program by using game-like elements. There is a lot of experience available in the field of gaming, Smartphone android, windows and ios apps, social utility apps and global network connection. Web applications are dime-a dozen available to learn anything that user wants. But none of the above fields have the working combination of learning while being fun and learning being competitive. People spend more and more time on the enjoyable software while learning nothing. We try to make the use of game like fun elements and competitive rating through social networking notifications in the learning fields. We aim to make programming addictive. The proven ways, the formative feedback and game-like elements will enhance the learning to program further more. Learning is made competitive by sharing the score, rating or credits earned on social platforms and keeping a leader board on our platform. Programming learning is particularly found difficult due to learning new and interrelated concepts simultaneously. This is derived from theory of cognitive load that validates humans cannot learnt to “high cognitive load”. The case worsens for first timers of programming which seeds the fear of programming in them due to little understanding. This has permanent negative impact of instigating fear of learning programming and decreases their self-confidence. In this paper, we have explained and designed web-based online learning platform for formative and continuous assessment, comprehensible by the student in terms of their progress. Its application is any learner counting to learn basic java theory and implementation. The quizzes and challenges are meticulously designed to decrease cognitive load and give the learners window for improving and succeeding in learning programming, particularly for new ones helping them build confidence to proceed. The learners on our web-app are surely to have learned the programming language thoroughly and better than those with conventional means. Our platform increases the efficiency and performance of learner by boosting self-confidence. The outcome of learning is definitely going to be better than traditional learning approach since motivation to learn due to need to get better rating on leader board will push the learner to do more and better. Structure of the paper goes like this; section 2 gives base and background of our platform. Section 3 explains the literature survey and section 4 explains design and implementation of platform with its framework and creation of material and its method of implementation of the programming language learning theory, solving quizzes and examples to be solved. Section 5 is conclusion.

II. BACKGROUND

A.Game-like elements/design

It is a use of elements of a game in a non-game design. This implementation is presently found in applications of commercial use but is yet to transcend into learning domain. E-learning materials with weak framework of game-like design
are appearing with little success. Rather they should help learners to make correct mental model of the problems. These elements should be used to make the learning process enjoyable helping in making the learner move forward. The quizzing should be made from easy to complicate with even complicated game like elements which should step by step clear the theoretical concepts well as practical implementations stronger to improve skills in programming.

B. Formative feedback

Ideally, what must be learnt and what is learnt should always be same. But seldom does this happen which is why we need assessment for evaluating what has been learnt and measuring it with reference to set standard of learning. Two different approaches of assessment are summative assessment which occurs after a package is learnt and learners are categorized into a grade brackets. Another type is the formative assessment, which is continually happening while learning take place. This method correct the mistake immediately which in turn makes the concepts learnt and constructs implemented, robust. It helps in providing immediate feedback, to rectify and resolves the mistakes and misconceptions. This feedback is also helpful to material and challenges designer for constantly improving the teamwork and design of platform. We can provide feedback to learner and improve instruction well.

C. Cognitive load theory

Cognitive Load Theory (CLT) [5] assumes that knowledge is encoded in schemata that are saved in long-term memory with an unbounded storage capacity. Learning takes place through the generation of new schemata. To generate new schemata or to combine multiple lower-level schemata in a single more complex one, the schemata have to be loaded into working memory. In contrast to long-term memory, working memory has only limited space (about seven items [7]). Cognitive load describes the amount of space that is needed in working memory to process new information. Intrinsic cognitive load depends on the complexity of the material to be learnt, specifically on the interactivity of the elements. To process material with high element interactivity, as is the case for learning to program, multiple schemata have to be loaded from long-term into working memory, leading to high intrinsic cognitive load. According to cognitive load theory, intrinsic cognitive load cannot directly be influenced by the instructional design, but there are two ways in which the intrinsic cognitive load can be minimized: Either through automation of schemata, so that the encoded procedural knowledge can be activated without imposing cognitive load at all [6]. Or, when previously unconnected schemata are combined to a single, more complex schema, loading it consumes only a single space in working memory. But this can only happen, when germane cognitive load is high, which can be defined as the cognitive load imposed on the learner to generate new schemata. Van Merriënboer and Kester [5] propose a comprehensive model for the instructional design to manage cognitive load in the course of a learning intervention. They take the expertise of the learner for the design of a sequence of tasks into account, so that students initially receive much support (worked examples) that gradually fades away (completion tasks) until finally no support is given (conventional tasks).

III. LITERATURE SURVEY

There are several researches within the field of on-line Gamification system and numerous papers are bestowed during this field. These paper foremost acceptable papers printed during this field and up the data that best suites North American country and can function a facilitate in developing this paper:-

The employment of the granular linguistic model of a development (GLMP) to model the assessment of whole number arithmetic learning and implement the machine-controlled generation of a formative criterion-based assessment report in tongue, similarly as a numerical grade [11]. Gamification could be a new technique in tutorial style that gains additional and additional attention within the e learning field. The implementation of gamification systems varies on totally different fields, like sales, web design, marketing, education etc. [2]. One among the quickest growing use of technology is for on-line education that could be a trend of latest education. And it's frequently up with perpetually developing and rewriting its programs to suit today's students [3].

IV. IMPLEMENTAION

A. System design and implementation

A key conception shared by each formative assessment and sensible games is feedback concerning get to clear, doable goals. In system design the content will be divided into completely different learning levels, which hold the challenges, classified by problem into stages. The difficulty level will be increased becoz of the programmatic questions which is going to be introduced in the next level .The next level get unfastened and may be difficult than the first level quiz becoz the second level deals with more tough and difficult programming questions which will challenge the user. User can have the expertise points (XP) at the top of the quiz so as to match their progress with another user. The content designer has outline the quiz check for every topic that the user reaching to learn so they're going to get the transient data for every topic.

1) Varieties of challenges

To control task problem and obligatory psychological feature load, the content designer has 2 differing kinds of challenges: multiple alternative challenges and programming challenges.

1) Multiple-Choice-Challenges

Multiple alternatives (MC) challenges area unit multiple alternative queries wherever either only 1 or multiple selections will be designated. As we all know in any game, we goto future level once the initial level gets cleared. There can a progress bar that shows the letter of quiz cleared thus on apprehend the progress of the player. There can (MC) on every that may be tutored, at the top to indicate the progress the result chart is provided. This chart consists the marks of every topic quiz given by the player in bar graph manner analysis.

![Figure 1. Multiple-choice-challenges](http://ijesc.org/)
2. Programming challenges

In Programming challenges, the focus is on increasing the competitive approach in programming among the players through time bounded challenges will to boot give rationalization feedback. Here the programmatic quizzes are given and user have to find the error or find the output of the program in limited time which will stress the brain to work faster and the quizzes are like as shown in fig.

3. What is the result when you compile and run the following code?

```java
1. class Top {
2.     static void myTop() {
3.         System.out.println("Testing myTop method in Top class");
4.     }
5. }
6. public class Down extends Top {
7.     static void myTop() {
8.         System.out.println("Testing myTop method in Down class");
9.     }
10.     public static void main(String[] args) {
11.         Top t = new Down();
12.         t.myTop();
13.     }
14. }
```

Choose the one below:

A) Compile Time error
B) Run Time error
C) Prints Testing myTop method in Top class on the console
D) Prints Testing myTop method in Down class on the screen

Figure 2. programming challenges

B. DESIGN OF FRAMEWORK

The design phase deals with the overview and quiz algorithm which is discussed below.

Figure 3. design of framework

The overview design or the dashboard contains learn, quiz, status and leadership parts which will redirect the user according to his need and maintain the flexibility.

Figure 4. flowchart

As shown in above flowchart, the user have to enter login credentials to get access to homepage where he can learn and then give assessment test and can also provide feedback on it. User cannot proceed to next level until the first level is cleared. User can check his progress whenever he want throughout the process in status board. One of the main feature element of it is that user can compare his performance with other user and get rank based on his performance.

C. CONTENT DESIGN AND IMPLEMENTATION

As the content (learning goals, levels and design) is aligned with the curriculum of the course, we first will give an overview of the course and its genesis.

A. Course syllabus

This course offers associate degree introduction into object-oriented programming and is control for college students from various backgrounds. The course is obtainable in conjunction with obligatory tutorials giving on particular topics. Basically the course syllabus contains all the core concepts in Java right from Introduction to I/O File topic which will provide in-depth knowledge about Java to the user.

Figure 5. course syllabus

Cognitive load concept is kept in mind while designing the learning goals and challenges. A most important design principle is that the difficulty of stages continually increases from level 1 to the next level. So, the challenges of level 1 stage deals with deep theoretical concepts and comprehension levels, while second level focuses on programmatic challenges (application).

1. Challenges, wherever students need to choose syntactically correct statements, as an example missing brackets or semi-colons, commas rather than semi-colons.
2. Challenges, wherever a tangle description is given and students need to choose the proper answer from a listing of alternatives.
3. Challenges, wherever a partial answer is given and also the player has got to choose the proper choice to complete the implementation.
4. Challenges, wherever code is given and also the student has got to choose that output are going to be generated once the code is dead.

5. Challenges, wherever code is given and also the student has got to choose underneath that circumstances it might be correct.

VI. CONCLUSION

The proposed system introduced a new approach of learning, a game-like element using platform and incorporating formative feedback and continuous assessment for programming language learning. This platform serves as a motivating tool involving different and wide-ranging quizzes and challenges. It can be further extended for learning other software or programming language learning courses i.e.; the frame work is easily adaptable for procedural programming, algorithms and data structure courses etc. and even for other courses where material with high element interactivity needs to be learnt. (Probability and statistical courses)

VII. REFERENCES

[1]. M. Gloria Sanchez-Torrubia; Carmen Torres-Blanc; Gracian Trivino 2013 IEEE Global Engineering Education Conference (EDUCON) Year: 2013. ‘Formative criterion based assessment for Moodle quizzes using intelligent computing’

[2]. Fotini Paraskeva2015 International Conference on Interactive Mobile Communication Technologies and Learning (IMCL) ‘Gamification design framework based on SDT for student motivation Dimitra Lamprinou’


[7]. Miller, G. A. (1956). The magical number seven, plus or minus two: some limits on our capacity for processing information. Psychological review, 63(2), 81.