DICE: A **D**YNAM**I**C **C**ROWDSOURCED **E**DUCATION ARCHITECTURE

Research Question:

Could a dynamic, continuously evolving education architecture powered by open source and version control systems establish a better student performance in the domain of education?

Problems with the current education:

1. Dependency to individuals and institutions

    a. DRY principle

    Each individual / institution creates their own content (course material), which is inefficient. One of the main purposes of DICE is to have single content for each course that each individual / institution could contribute to.

    b. Opinionated content

    Dependency of individuals creates personal bias. DICE prevents personal biases via a crowdsourced open source architecture.

    c. Outdated content

    In today's world, information evolves really fast. Individually created contents could be outdated very quickly. DICE consistently updates its learning materials through crowdsourced contributions.

2. Lack of Personalization

    a. According to personal knowledge level & pace

    There needs to be separate contents for each knowledge level. DICE's knowledge level structure provides a systematic way to teach any knowledge level. Since the education happens online, any learner could learn at his/her own pace.

Solution

The proposed architecture (DICE) will be designed to work on cloud. A web application followed by mobile applications will be the outcome of this research. In order for the system to be crowdsourced and democratized, all the content and the source code are completely open sourced. Below, 4 core concepts of DICE are introduced.

Version Control System

A version control system is at the core of DICE. It enables users to contribute to the source code and to the education material. Although the most popular version control system today is Git, a new more user-friendly VCS is needed so that any learner even without a technical background could contribute to any content.

Knowledge Levels

Each course in DICE is composed of knowledge nodes (blue squares in the Figure 1). These nodes can be described as subjects in a course. Each node consists of learning assets such as slides, animations etc. and each node's performance is assessed with a test at the end.

DICE has a very common tree structure for any course. Education material needs to be prepared according to knowledge levels. A user always starts with the first level and if s/he wants s/he can go as deep as s/he wants through the levels. Figure 1 shows a sample hierarchy of the nodes. Nodes are sub-subjects of their mother node in the upper level. Overall, DICE has a top-down knowledge hierarchy for learning.

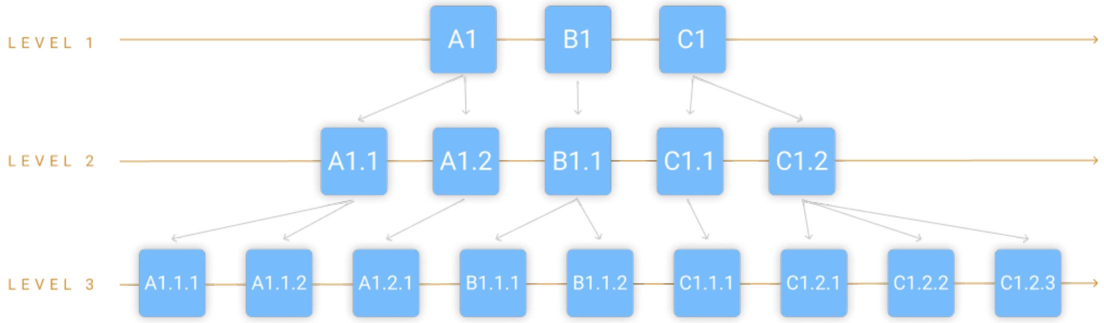

Figure 1: Knowledge Levels of a Course

Continuous Improvement

DICE continuously updates its contents in order to increase the student performance.

When there is a new contribution, it goes into the **proposal stage**. A contribution in the proposal stage goes for approval from the course board. Members of the course board approves/disapproves the contribution. If a certain threshold is exceeded, the change is submitted to the trial state.

When a contribution is in the **trial stage**, a new node is created in the learning path of the course. Each node has a possibility which indicates the chance of being displayed to a student through learning. When there are trial nodes along any learning pathway, DICE will try the new nodes with a total possibility of 20%, which means that it shows the new proposed contents to the 20% of the students. Remaining 80% of the students will be displayed in the master branch of the content. This 20% of possibility is again distributed among the number of new proposed nodes. For example, if there are 2 new proposed nodes, each will have 10% chance of display.

Trial stage continues until there is enough data to assess the performance of the proposed nodes. This threshold will be calculated according to the population size of the students and the tested sample need to represent the whole population. Once the threshold is reached, master node performance is compared to the new node performance. If the new one is more successful, it is merged with the master learning path. For example, when B1.1’ is completed its trial stage and found to be more successful than B1.1, it is merged to the master branch and the possibilities will be updated as: B1.1 (0), B1.1’ (0.8), B1.1’’ (0.2). DICE will continue to test the B1.1’’ node after the merge.

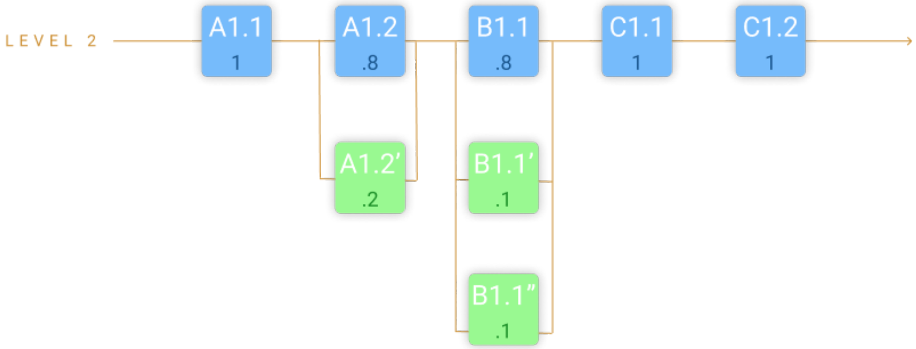

Figure 2: Continuous Improvement in DICE

Incentive Mechanism

When a new node is merged into the master branch, its creator will gain teaching points (think of it as the stars in GitHub.). Users’ teaching points will be displayed on their public profile pages. When a user’s teaching points for a subject exceeds a certain threshold, DICE will ask the user if s/he wants to be included in the course board.