What to do after AP CS? A case study on short technical modules for post-AP-exam high school students

Sabeel Mansuri[1], Hannah Zhou[2], Hemanth Battu[2], Alex Fong[1], Tiffany Ho[1], Eric Xiao[1], Niema Moshiri[1]

[1]Department of Computer Science and Engineering, Jacobs School of Engineering, University of California, San Diego, CA, USA
[2]Department of Mathematics, University of California, San Diego, CA, USA

UC San Diego

Abstract:

Students who complete the Advanced Placement (AP) Computer Science exam typically lack structured content post-exam. While students may be interested in applying their newly-gained skills to practical real-world problems, AP Computer Science teachers may not have the educational content nor the expertise to execute such supplemental lessons. In this case study, we introduce a series of short (5-day), self-paced instructional modules implemented as Massive Adaptive Interactive Texts (MAITs) on the Stepik platform. These self-paced modules can be completed primarily in the user's web browser, and can thus be integrated into existing courses with ease. We discuss the pedagogical methods utilized in their development, and we provide an overview of the project results of the development of the first two mini-courses: "Tic-Tac-Toe Minimax" and "An Introduction to Computer Vision".

Introduction

Having taken AP CS only a few years back, we've experienced first-hand the lack of structure in classrooms following the AP exam. We set out to rectify this by connecting high schools with the technical expertise offered by the Computer Science and Engineering department at the University of California, San Diego.

Nearly a century ago, Professor of Psychology Sidney L. Pressy observed, "Education is the one major activity which has thus far not applied ingenuity to the solution of its problems." We, like Professor Pressy, were unsatisfied with the prominent educational techniques available for implementing our AP CS outreach program. Consequently, we chosen to explore our own.

Inspired by the active-learning approach offered in Phillip Compeau and Pavel Pevzner's text *Bioinformatics Algorithms*, we've developed two MAIT modules using an online educational content platform called Stepik. We intend to use these modules to collect data on AP CS student performance and readiness.

Methods:

Tic-Tac-Toe Minimax

This module allows students to explore basic artificial intelligence principles by building their own Tic-Tac-Toe AI.

Day 1) High-level introduction to active learning, artificial intelligence, zero-sum games, and adversarial search.

Day 2) Introduction to decision trees with and emphasis on a zero-sum context.

Day 3) Implement core Minimax algorithm, converting logical principles from Day 2 to Java code.

Day 4) Build the Tic-Tac-Toe game class and the decision tree.

Day 5) Completion of project; playing against AI using a GUI facilitated by Jupyter notebooks.

Introduction to Computer Vision

This module introduces students to basic computer vision concepts by allowing them to rebuild Snapchat filters.

Day 1) Review of basic object-oriented principles and introduction to relevant data structures.

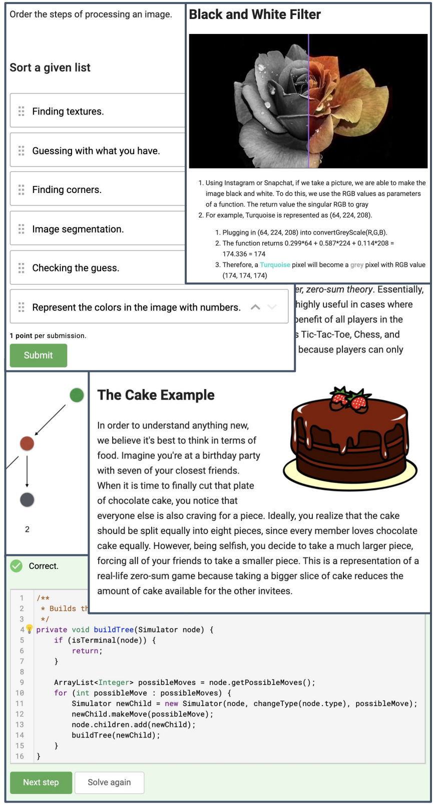
Day 2) Explanation of how images are represented in computers: pixels, RGB, and HSV

Day 3) Deep-dive into core computer vision operations: thresholding, masking, segmentation, and bounding boxes.

Day 4) Introduction to OpenCV library and interactive exploration of basic image recognition techniques for Snapchat filters.

Day 5) Implementation of final project; developing own Snapchat filter using Jupyter notebooks.

Results



Order the steps of processing an image.

Sort a given list

- Finding textures.
- Guessing with what you have.
- Finding corners.
- Image segmentation.
- Checking the guess.
- Represent the colors in the image with numbers.

1 point per submission.

Black and White Filter

1. Using Instagram or Snapchat, if we take a picture, we are able to make the image black and white. To do this, we use the RGB values as parameters of a function. The return value the singular RGB to gray
2. For example, Turquoise is represented as (64, 224, 208).
    1. Plugging in (64, 224, 208) into convertGreyScale(R,G,B).
    2. The function returns 0.299*64 + 0.587*224 + 0.114*208 = 174.336 = 174
    3. Therefore, a Turquoise pixel will become a grey pixel with RGB value (174, 174, 174)

er, *zero-sum theory*. Essentially,
highly useful in cases where
enefit of all players in the
s Tic-Tac-Toe, Chess, and
because players can only

The Cake Example

In order to understand anything new, we believe it's best to think in terms of food. Imagine you're at a birthday party with seven of your closest friends. When it is time to finally cut that plate of chocolate cake, you notice that everyone else is also craving for a piece. Ideally, you realize that the cake should be split equally into eight pieces, since every member loves chocolate cake equally. However, being selfish, you decide to take a much larger piece, forcing all of your friends to take a smaller piece. This is a representation of a real-life zero-sum game because taking a bigger slice of cake reduces the amount of cake available for the other invitees.

✓ Correct.

```
/**
 * Builds th
 */
private void buildTree(Simulator node) {
    if (isTerminal(node)) {
        return;
    }

    ArrayList<Integer> possibleMoves = node.getPossibleMoves();
    for (int possibleMove : possibleMoves) {
        Simulator newChild = new Simulator(node, changeType(node.type), possibleMove);
        newChild.makeMove(possibleMove);
        node.children.add(newChild);
        buildTree(newChild);
    }
}
```

Conclusion

Having completed the development of these modules, we now turn our focus to implementing them in local AP CS classrooms. We will work with educators to streamline the process of offering the material to work toward a seamless integration into post-exam curriculum.

We look forward to analyzing student performance data as a means of evaluating the efficacy of:

1) these online modules in providing students' with valuable instruction
2) the AP exam in preparing students for real-world applications

Additionally, given the recent need for remote instruction in light of COVID-19, we hope this initiative can serve as a means to alleviate educators' stress as they attempt to develop their own online material. We see these scalable, modular lessons as a useful example of how to effectively implement remote instruction.

References

1) Compeau, Phillip, and Pavel Pevzner. *Bioinformatics Algorithms: An Active Learning Approach*. 3rd ed., Active Learning Publishers, 2018.

2) Compeau, Phillip, and Pavel A. Pevzner. *Life after MOOCs.* Communications of the ACM, vol. 58, no. 10, 2015, pp. 41–44., doi:10.1145/2686871.

3) Compeau, Phillip. "Establishing a Computational Biology Flipped Classroom." PLOS Computational Biology, vol. 15, no. 5, 2019, doi:10.1371/journal.pcbi.1006764.

4) González-Velasco, Donna M. *Fun with Fibonacci; or, What to Do after the BC Calculus Exam.* The Mathematics Teacher, vol. 81, no. 5, 1988, pp. 398–400. JSTOR, www.jstor.org/stable/27965840.

5) Moshiri, Niema and Liz Izhikevich. *Design and Analysis of Data Structures*. Self-Published, 2018.