2022

Computer-Based Scaffolding In Computer Science Education

Rebecca Trinh

Simone Levy

COMPUTER-BASED SCAFFOLDING IN COMPUTER SCIENCE EDUCATION

Simone Levy & Rebecca Trinh

Advisors: Dr. David Shepherd, Ph.D. & Juliana De Souza, Ph.D. Candidate

Introduction

Contemporary STEM instruction creates a challenge for professors to introduce students to complex learning topics. With little exposure to these topics before college, students will find it hard to fully comprehend syntax, algorithms, and a large volume of code without past exposure.

Scaffolding allows professors to slowly introduce complex topics to students by using **progressive information building** and encouraging an **explanatory** and **exploratory** work process rather than strictly instructional [1]. These methods were implemented into a large scale coding project where students operate on various tasks in C with the objective of gradually improving their software engineering skills.

Methodology

- Code and assignment comparisons were made from previous class instruction.
- Comments and an “Assignment Information.txt” file were added to help familiarize students with the Xfig environment.
- Asking questions, providing syntax information, and encouraging students to search for specific information in tasks were used to allow students to grasp information on their own.

```
// #taskEnhancedRotation
/*-------------------------------Code Starts Here-------------------------------*/
// This code enables xfig to rotate shapes to different degree angles. Currently, xfig is locked
// to 90 and 180 degrees. How can you change xfig to accept more angles options than the ones
// defined below? Eg. 0, 33, 45 and 310 degrees.
// This project has infinite solutions, you can make the program accept any type of value.
// The function 'fabs(act_rotangle)' is updating how much the object will rotate.
// Challenge: Verify if the angle is valid. If it is not, convert it to a valid angle.

    F_line          *l;
    F_compound      *c1;

    if (fabs(act_rotangle) == 90.0 || fabs(act_rotangle) == 180.0)
        return 1;
    else if (!valid_rot_angle(c1))
        return 0;

// Once you are done, save the file and go to the next file.
/*--------------------------------Code ends Here--------------------------------*/

    return 1;
}
```

Fig. 1: Example of comments in a task [1]

Results

The methodologies in the improved tasks were implemented to be taught to students in the coming semester. These revamped tasks have increasing difficulty that allow students freedom in their learning and encourages them to seek out solutions on their own.

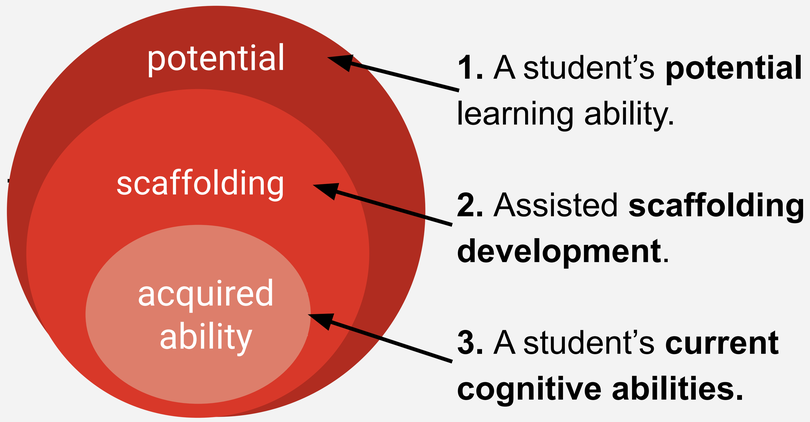

Fig. 2: Derivation of the Zone of Proximal Development by Lev Vygotsky [2]

Discussion

The “Assignment Information.txt” file provides multiple resources to help guide the student–such as written and video tutorials–and takes into consideration of different learning styles.

As students progress through each assignment, their base of understanding expands and can be applied to the next tasks. This method allows students to learn through **association** [3].

To keep students engaged and avoid varying knowledge levels, an additional challenge task was added to each assignment, allowing students with a higher skill level to make different optimizations.

Conclusion

In this project, we utilized computer-based scaffolding strategies to analyze how students respond to scaffolding in larger scale projects.

By creating an exploratory method of learning, we can allow students the opportunity to comprehend larger projects at their own pace. Over time this method should expand a student’s potential learning ability and encourage their ability to understand harder topics in computer science education.

It is important to create a progressive learning environment for classrooms teaching complex themes, as this will give students the confidence they need to succeed in the classroom and ingrain in them learning abilities that will assist them in future classes and outside of the classroom.

Future modifications and use of different strategies are still open to research based on the students’ response to the assignments. Finding the best way for students to learn computer science would be worth exploring.

References

[1] Wu, H.-L. (2011). *Scaffolding in technology-enhanced science education* (dissertation). Texas A & M University, College Station, TX, Texas.

[2] Moore, S. A., & Rhodes, D. (2004). *Theoretical and practical perspectives on Vygotsky's concept of the zone of proximal development* (dissertation).

[3] Mayer, R.E. (1981). *The psychology of how novices learn computer programming*. ACM Computing Surveys (CSUR), 13(1), pp. 135