Exploring Inquiry Learning: An EngageCSEdu Author and a User Discuss POGIL

When structured well, collaborative learning can build computer science (CS) knowledge while growing a more inclusive student community [1,2]. In this issue, we explore a specific kind of collaborative learning—POGIL—Process-Oriented Guided Inquiry Learning. As the developers of POGIL in CS describe it, in POGIL students work in teams on activities that are specifically designed to guide them to construct their own understanding of key concepts, and at the same time to develop important process skills such as communication, critical thinking, and problem solving. The instructor is not a lecturer, but an active facilitator of student learning [3].

POGIL can also help dispel the persistent stereotype of the lone coder that can turn off some students from computer science. Instead, through POGIL, students experience computer science as a collaborative, active, problem-solving activity.

In this article, Clif Kussmaul, a contributor of several POGIL assignments to the EngageCSEdu collection and a lead researcher in POGIL in computer science, and Bo Brinkman, a computer science faculty member who is newly implementing POGIL in his introductory courses, share how they discovered the technique and offer tips for effectively using it in computer science classrooms. The conversation is facilitated by Beth Quinn, Director of NCWIT’s EngageCSEdu project. The interview has been edited for clarity and length.

How did you start using POGIL in your classes?
Clif: I had a mentor years ago who was doing POGIL before it was called POGIL. He kept talking about how he didn’t give lectures. I thought, “Yeah, okay, that sounds kind of interesting,” but he was a chemist. It wasn’t until 2009 at ASEE, the American Society for Engineering Education conference, that I attended a workshop on POGIL. The “ah-ha moment” for me was when I started writing questions about for loops and realized that it might really help students understand something that they struggled with every semester. Later that year, I was on sabbatical in India and teaching a new course, and thought this was my best chance to try it. It worked remarkably well. I was really surprised by how engaged the students were and how much interaction and discussion there was. I came back to the US and wrote my first NSF grant to develop POGIL in computer science.

At this point, I use POGIL in nearly all my courses and especially in the introductory courses I teach regularly, since I’ve had more time to develop and refine those materials. In some courses, two days out of three will be POGIL, and maybe one day out of three in other courses. But it’s pretty much always there.

Bo: I got interested in POGIL about four years ago when I saw a talk at SIGCSE. But I had never found the time to do it until two years ago. I was asked to participate in a faculty learning community at my institution that was charged with reforming the way we teach in STEM, and specifically with exploring how we might use active and inquiry-based learning. There were a lot of naysayers, especially of inquiry-based learning. People were saying, “I tried this and there’s no evidence that it works.” So, I was unclear whether inquiry-based learning was a good idea. What won me over was an unrelated workshop from Clif and his collaborators. They conducted most of the workshop in POGIL, and when I experienced learning in that mode, I really saw its potential.

I started taking a couple of Clif’s assignments—just once or twice across the semester—and sticking them in to see how the students reacted. The students loved them! This semester is the first time I’ve done one POGIL every week in my Foundations of Computing course (a CS0 course). Recently, I had a student stop me to say that she was learning a lot from the labs and it was her favorite thing about the course. This is a big class! The lecture is a hundred students—which for me is big—so to have a student bring that out is a good sign. When I’ve done them in previous semesters and asked students at the end of the semester which lab sections were the best ones, the POGIL sessions were their favorites.

An Example POGIL Assignment in EngageCSEdu
Beth: One of the POGIL assignments you have in the EngageCSEdu collection, Clif, is called, Searching in Hi-Lo [7]. Tell us about how it works.

Clif: Yes, this is something I was doing pre-POGIL. Then, I realized it would work even better with POGIL. It’s an activity that I typically do the first day of class of CS1.
I put the students in groups and I present them with a simple game. One person thinks of a number, another person guesses. The first person either says, “too high,” “too low,” or “you’re right.” Each group plays the game a couple of times. Then I have them describe different ways you might play the game to come up with an answer. Each team comes up with several different strategies. Then, as a class, we discuss the strategies because not every team is going to come up with all the ones I want them to think about. Specifically, binary search is the strategy that a computer scientist would think about, but that’s not always obvious to a first-day student, right? Usually we do a qualitative analysis round where we simply rank the strategies from slow to fast and easy to hard. Then, we do a quantitative round where we discuss the maximum number of guesses you would need to win with this strategy. So, on the first day of class, students basically invent Big O-style notation. And on the first day, they see that computer science is about working in teams, about analyzing problems, designing multiple solutions, evaluating solutions, and that we use both qualitative and quantitative techniques. It’s a great activity for the first day because it’s going to make a first impression. Students are deciding either “This is going to be an awesome class” or “Boy, this is going to be a long semester.” So, I’d much rather do this on the first day then pass out the syllabus and spend 50 minutes reciting policies at them.

**Bo:** Yes, I’ve used this assignment multiple times in two different Intro to Programming courses. It works on so many levels and it flies in the face of the stereotypes about what computer science is about. I think it’s more truthful. Computer science is not about coding all day, except at the entry level. A lot of it is about design and multiple people working together.

**Strengths and Pitfalls**

**Bo:** My mistake this semester is not sufficiently preparing my lab facilitators. My course has two lectures with 100 students. They split into six lab sections that run concurrently with an undergraduate teaching assistant in each. The TAs are fantastic, but with POGIL, the facilitator must play a strong role. They need to be comfortable “cold calling” on groups and asking good follow-up questions. I didn’t do a perfect job of preparing my teaching assistants to do that. You need to plan ahead if the person who’s facilitating the POGIL is not an experienced teacher.

**Clif:** You know, the best way for a faculty member to prepare is to attend one of the POGIL Project’s three-day workshops that are held every summer[3]. There are a lot of pieces to POGIL and they all need to click together for the guided inquiry part of POGIL to really work. If I were in Bo’s situation, I would do something like a half-day, or even a day, of POGIL training with the teaching assistants. POGIL assumes that there’s a teacher or a TA there in the room to facilitate [8]. That’s a big part of it. It’s not just a worksheet for students to go through. The point of guided inquiry is that you’re giving students the right scaffolding to help them do the things that an expert does automatically.

**Bo:** I do think that’s one of the strengths. When I was hearing all these complaints that inquiry learning doesn’t work, I think it was because people were
trying to do inquiry learning that was not sufficiently guided. In my prior attempts at it I would say, “Do this, do this, do this” and then ask, “What did you observe?” And that’s a terrible process! How are students going to make this magic leap to observing the thing I want them to observe? But Cliff has this great POGIL design canvas that has helped me avoid this pitfall by having me think through a series of questions: What are you going to show them? Then, what are they going to be able to do by looking at the thing you showed them?

Beth: You see the same thing with pair programming. Just throwing people together without instruction is not so helpful. But really enforcing the roles, having people switch roles regularly, and matching skill levels, works better [1,9]. Is there also a mechanism in POGIL for people to switch roles and groups, and to form groups in the first place?

Cliff: Typically, I try to keep groups together for, if not the whole semester, at least several weeks because it takes a while to learn how to work effectively with a group. In fact, that often helps students work more effectively. When you’re with three buddies and two of the buddies aren’t taking things very seriously, it’s harder for the other two to enforce things. Where-as if you’re with three people that are not your buddies, everyone is on slightly better behavior.

Bo: We had been letting students form their own groups. But we had some issues. For example, we have a large population of Chinese students. Some groups were all Chinese students and sometimes they struggled with vocabulary. We went to randomizing because it’s much more likely that you’ll have at least one English speaker in a group and that helps students get over one speed bump. But I’m not 100% sure that’s the right way to do things. What I would like to be able to do is to make sure that in each group no one is “the only _.” There’s no group where there’s only one woman, or one African American student, or one Chinese student. Right now, I don’t have a good mechanism for doing that at the scale that I’m currently teaching, but that’s what I want to do.

Beth: Yes, NCWIT recommends not isolating women or minority students in a group, if possible, when doing group work in class [5].

Cliff: I agree that not isolating students is a best practice for this kind of work. Again, it’s also about learning to work with people that aren’t like you and that you may not know well. But it’s in a controlled environment. The roles help people to work together, the questions structure the pace and direction of the work, and the instructor is there as a guide or coach.

And then, what is it that you’re going to get them to try to invent? And finally, how are they going to apply that? My guess is that most people who have tried to do inquiry-based learning don’t have a good structure like that to work from.

Cliff: I think people hear “inquiry” and they think students try to figure things out completely on their own. That’s great in certain settings but it’s a very different thing from what POGIL tries to do.

Forming Groups and Assigning Roles

Bo: One of the things I appreciate about the POGIL approach is that there are defined roles for members of each team: the manager, the recorder, the speaker, and the reflector. Each person has specific responsibilities. Whenever I’m in a group, I tend to take over. What I like about POGIL is that it doesn’t let people like me do that! It means that everyone is contributing something of value. That structure—that everyone has meaningful work to do—is really helpful.

Beth: Yes, NCWIT recommends not isolating women or minority students in a group, if possible, when doing group work in class [5].

Cliff: I agree that not isolating students is a best practice for this kind of work. Again, it’s also about learning to work with people that aren’t like you and that you may not know well. But it’s in a controlled environment. The roles help people to work together, the questions structure the pace and direction of the work, and the instructor is there as a guide or coach.

Conclusion

Cliff: I started using this metaphor recently to explain POGIL. When you learn to play a sport or to play a musical instrument, your primary learning isn’t from reading a book or watching videos or listening to someone talk. You learn by doing the thing with an expert watching you and giving you feedback. That’s what a coach does. They watch you and say, “That was good, keep doing it” or “That was bad, try this.” Right? Using POGIL in a computer science classroom is a way to have students do much more authentic work with more direct
Exploring Inquiry Learning: An EngageCSEdu Author and a User Discuss POGIL

feedback from an instructor. If I want my students to learn to work on teams, I can assign them a team project and hope that, somehow, they learn to work in teams. But we know that most of the time one or two of the students does the work and the rest coast. If I lecture and they copy from slides, those are not the skills my students will need when they graduate. The more that I can have them, in class, analyzing problems, designing solutions, looking at code, designing test cases, analyzing pros and cons, the better. I can also coach them on teamwork, interpersonal skills, and time management. When I’m right there as they work together, I can see the interpersonal problems and the conceptual blocks. I have more data and I become a better teacher faster as a result.

Bo: I think that my big take-away is that in POGIL I found a systematic way to do inquiry learning that I believe works. And I’ve found a systematic way to develop new course materials that works. That’s something that my institution had been looking for and that nobody knew how to do. POGIL is a way to do it.

References

More POGIL Assignments in EngageCSEdu

POGIL Activities (3) on Unit Testing in Java with JUnit by Clif Kussmaul. Three to four students work in teams to learn about unit testing in general, and JUnit specifically [6].

Beetle by Peter Drake. In this lab, students dissect a working implementation of the Beetle (Cootie) game. Learning objectives include understanding class anatomy, using objects, and problem solving [4].

Clifton Kussmaul
Muhlenberg College
Department of Mathematics & Computer Science
kussmaul@muhlenberg.edu

Bo Brinkman
Miami University
College of Engineering and Computing, Computer Science and Software Engineering
bo.brinkman@miamioh.edu

Beth A. Quinn
University of Colorado at Boulder
National Center for Women & Information Technology
beth.quinn@ncwit.org

DOI: 10.1145/3125650 Copyright held by authors.