Fun and Interactive Activities for an Introductory Computer Science Course of 200 Students

Stephany Coffman-Wolph and Kimberlyn Gray

The University of Texas at Austin
2317 Speedway, Austin, TX, 78712, USA
E-mail: sscw@cs.utexas.edu

West Virginia University Institute of Technology
512 S Kanawha St, Beckley, WV, 25801, USA
E-mail: Kimberlyn.Gray@mail.wvu.edu

Abstract

Research has shown that students learn and perform better in learning environments that are interactive [1]. Teaching a freshman-level introductory course in computer science (CS) can be challenging, because most students are unaware of what computer scientists do and have never been introduced to basic computer science concepts. Using a series of hands-on interactive activities throughout the semester can introduce CS topics in a fun way while relating the topics to familiar everyday experiences. All the activities listed below do not require a computer lab, are extremely cost-effective, and require minimum preparation: (1) Understanding Variables and Arrays with Paper Bags [2], (2) Branching and Looping Statements with Starburst Candies [2], (3) Loops with Music [3], (4) Arrays with Tissue Boxes, DVD Sets, Paper Plates, and other Household Goods [2], (5) Monsters Hate Chocolate: Learning Try/Catch Blocks [2], (6) General Class Structure with Bags, Boxes, and a Bin [2], (7) Dr. Doolittle's Vet Office: Learning Classes with Stuffed Animals [2], (8) Sorting Algorithms with Paper Bags [3, 6, 7], and (9) Recursion Introduction: Simple Tower of Hanoi with Colored Paper [3]. The authors designed these activities to be done with a variety of age groups and various numbers of students, however, the activities have never been tired with classes larger than 50 [2, 3]. This paper will explore the challenges and logistics of adapting these activities to a forum style lecture hall with approximately 200 students in attendance. Additionally, any revised activity instructions from [2, 3] will be provided.

1. Introduction

Teaching lectures with hundreds of students in a large lecture hall comes with a variety of challenges. The addition of active learning into large classrooms has been shown to be highly effective overall [4] but it has been reported that is often less effective with freshman than senior courses [5] and comes with logistical issues [4]. The authors will be focusing on overcoming these logistical issues for activities designed for a smaller classroom. This paper will discuss each activity, the challenges in implementing it in a large classroom, and possible solutions to these challenges. (If the reader wishes to find more details on the activities, please refer to [2, 3]).

2. Overview of the Activities

The nine activities that will be covered in detail below are as follows: (1) Understanding Variables and Arrays with Paper Bags [2], (2) Branching and Looping Statements with Starburst Candies [2], (3) Loops with Music [3], (4) Arrays with Tissue Boxes, DVD Sets, Paper Plates, and other Household Goods [2], (5) Monsters Hate Chocolate: Learning Try/Catch Blocks [2], (6) General Class Structure with Bags, Boxes, and a Bin [2], (7) Dr. Doolittle's Vet Office: Learning Classes with Stuffed Animals [2], (8) Sorting Algorithms with Paper Bags [3], and (9) Recursion Introduction: Simple Tower of Hanoi with Colored Paper [3]. Key topics covered by these activities include: (1) variables, (2) arrays, (3) if statements, (4) loops, (5) Try/Catch blocks, (6) Exceptions, (7) classes, (8) sort algorithms, and (9) recursion. Additionally, these activities help students will the following skills: (1) teamwork, (2) code review, (3) translating real-world problems into code, and (4) algorithm design. For a detailed description of all activities and a set of instructions for using them in a class, please see [2] and [3].

3. Understanding Variables and Arrays: Paper Bags Visualization

For this activity, the idea is to introduce the concept of variables and arrays using a variety of different sized paper bag or same sized bags attached together in a row [2]. Each size bag represents a different data type such as string, integer, etc. A variable name is written on the bag and a slip of paper with a number or word is inserted into the bag as the data stored in the variable. As the bags are named and data is assigned, the same information is written in the appropriate programming language for students to see.There are four reasons that paper bags were chosen: (1) inexpensive and easily available, (2) visual representation of size difference between variable types, (3) students would understand that a variable holds a single value, and (4) each variable or array has a unique name. Additionally, the use of bags attached together visually shows that an array is simply a set of variables of same size that each hold a single value. This activity is done in a demonstration style (i.e., the "action" happens in the front of the room).

There are several challenges to adapting this activity for a large classroom: (1) students in the back of the room seeing the demo and (2) students participating during the demo. The first of these challenges is easily remedied by increasing the bag size or switching to foldable bins. Additionally, we enlarge the size of the signs for the variables (and the values being put into each variable) so that students can see. Another workaround is to utilize a camera and the overhead projector. With the help of a teaching assistant or student volunteer, a second demonstration could be set up towards the back of the lecture hall, on a row kept empty for this purpose. Once students understand the concept of a variable, the bags or bins can be attached to show the concept of an array. The bags can be attached using a brightly colored ribbon or placed in the rungs of a rope ladder laid onto the floor. To have all students interact during the demo, use a response system (iClickers, quiz software, chat room, bulletin board, or etc.) to quiz students about the demo. This will allow all students within the room to participate with answers shown anonymously.

4. Branching and Looping Statements: Starburst Candies Group Activity

This activity uses real-world situations to introduce if statements and while loops to students [2]. The activity has two parts; the first uses a series of if statements in pseudocode to select a team leader for the activity. Once a leader is selected, the students simulate a while loop by passing a bag of Starburst candies around, each drawing a single Starburst, until someone draws a red one. The students then create the corresponding flowcharts for the activity actions and translate the steps in the chart into code, allowing them to see real world actions written in a programming language [2]. Unlike the previous activity, this one is done by the students in groups with the instructor and teaching assistants moving around the room answering questions or gently re-directing groups that are going in the wrong direction.

There are several challenges to adapting this activity for a larger classroom: (1) dividing the students into groups, (2) working with a forum style lecture hall sitting structure, (3) walking around the room, (4) cost of Starburst candies, and (5) submission of flowcharts and code. There are several ways to handle dividing the students into groups. The first is to pre-select groups for the semester and have the students automatically get into these groups when they arrive in class. Another is to hand out numbers randomly at the lecture hall doors as students arrive to create somewhat random groups. A third option is to just use rows (or columns) to create the groups once students have arrived. Most large classes are assigned to forum style lecture halls that contain inflexible seating that often does not work well for group work. Depending on the room structure, a way to overcome this challenge is to provide a seating chart upon arrival that would as best as possible spread the groups out in little clusters with spaces left open to accommodate instructor and TA movement during class. Sections of the lecture hall would be assigned to a particular TA or instructor to increase the amount of time each group has for assistance. In regards to the cost of Starburst candies challenge, one could replace the candies with a cheaper material (beads, pieces of paper, etc.). Reusable materials would cut down on long term costs and make repeat preps faster especially if teams help to reset their own material.

The final challenge is the submission of flowcharts and code and the grading associated with it. The logistics of collecting all the papers and grading them can be challenging with so many students. For in class hardcopy submission, the use of a predetermined seating plan and TA assignment would mean faster collection of papers. The predetermined seating arrangement method would also work with random group assignment in class as students could simply list all team members on the assignment. For either method, groups could swap papers and grade either other. However, this method should be used only after students have received feedback from the instructor or had the instructor demonstrate code review and feedback in class. Digital documents are often less problematic (less to carry, do not need to be handed back in-person, easier grading, etc.) but can be an issue if everyone does not have access to a laptop or tablet at class. However, most students have a cellphone and can take a picture for electronic submission. Also, for group submissions, teams would need to be chosen in advance of class and created within the submission system, limiting group assignment. However,

group submissions decrease the number of solutions that must be graded and speed up feedback response time.

5. Loops: Practicing with Music

This demo style activity scales up to a large classroom style with very few challenges [3]. The entire class participates by doing certain actions during various conditions (i.e., loop). The students are divided into two groups, the clappers and the snappers. Using music, the students experience various loop types and loop conditions physically. Beginning with a simply while loop with the condition that if the instructors right hand is raised the clappers clap and when the instructors left hand is raised the snappers snap. The corresponding code is displayed to the students for each loop and loop condition throughout the activity. The activity continues using various combinations of conditions (e.g., both hands are raised) and other loop types (e.g., clap and/or snap for so many counts of the music for a for loop).

The challenge of providing enough materials does not apply in this case. All students participate during the activity and the only required materials are music and a speaker (although most traditional large lecture hall set ups have a sound system). Preparation for this activity is minimal: (1) select music, (2) divide the room into sections, and (3) have PowerPoint slides created for the various loops demonstrated in class (if the chalk / marker board is inaccessible for viewing in the classroom when a computer is being used). The original activity used a simple division down the middle for creating two groups. In a large lecture hall, there might be more options (designating specific rows, using aisles as dividers, leaving a row or column of seats empty, etc). However, with a larger group, one could expand the activity to have more groups and more complex loop conditions.

6. Arrays: Tissue Boxes, DVD Sets, Paper Plates, and other Household Goods

The instructor displays a variety of items at the front of the classroom and asks students what all the items have in common [2]. Next, the instructor explains how each item as a whole contains multiple items that are the same (e.g., a DVD set contains several DVDs). Using this as a launching point for the lecture, the instructor goes on to demonstrate the programming syntax for arrays and makes comparisons back to the items at the front of the classroom. Including pointing out that a single name is given to the overall item but each element can be referenced by a number (e.g., DVD set for a particular season of a tv series contains DVD discs numbed 1-5).

This, like the previous demo style activity, easily scales up to a large classroom style. The instructor simply needs to select large items that are highly visible or are extremely common that students instantly recognize from a vocal description. (Suggestions for objects: paper towel sets, large facial tissue sets, moving boxes with the glass/ stemware inserts). Student can discuss questions asked during the activity by turning to a neighbor. The challenge that remains is student feedback from those discussions which can easily be overcome by having the students submit responses via an online response platform (quiz, iClicker, chat room, etc). Other ways to increase participation is to give the questions in the form of multiple choice and have students select an answer with a show of hands for each choice or simply calling on random pairs to give their answers.

7. Learning Try/Catch Blocks: Monsters Hate Chocolate:

This activity was designed to be done in small groups. Each group receives a paper sack that acts as the "monster" for the activity, a program that serves as their instructions for the activity, and a bag of assorted candies. Each group tries to "feed" the monster based on a set of coding statements in the given program [2]. The statements that contain an exception (i.e., the chocolate candies) do not fit through the hole in the "monster" bag. The statements that are valid code match the Starburst candies and fit into the "monster" bag hole. The challenges are similar to the previous mentioned group activity (Branching and Looping Statements with Starburst Candies): (1) dividing the students into groups, (2) working with a forum style lecture hall sitting structure, (3) walking around the room, and (4) cost of candies. For this activity, one should employ the same tactics as mentioned easier for the challenges (1), (2), and (3). A predetermined seating chart and dividing the students into groups in some logical manner. For this activity, the fourth challenge of replacing the candy is slightly more difficult as you need two differently size objects that are similar and that students would believe a monster would eat. Possibilities include: plastic animals/ bugs in two sizes, beads in two sizes, mega bloks and/or legos, and different sizes of paper.

8. Class Structure: Using Bags, Boxes, and a Bin Demonstration

The programming concept of a class can be represented with a large bin and the elements of that class (variables and functions) being represented with bags and boxes [2].Students often have difficulty with the concept of a class, not realizing that they are creating their own data type, which can be more complicated than the built-in data types. A class can contain as many variables (of any or multiple data types) as well as perform functions with those data types. Each bag or box will represent a variable that can be used within the class or a function that the class itself will perform on that data. This activity, like several

others previously discussed, is demo style. Thus, it lends itself easily to a larger classroom setting. The challenges for this activity are: (1) consistency with earlier activities, (2) students being able to see, and (3) maintaining class participation. The earlier activity used paper bags or folded bins to represent variables (and arrays) and, for consistency, they should be reused here. It is important to use large size signs with large lettering so all students can see the variables and functions from the back of the room. As mentioned in the understanding variables and arrays activity, a camera might be helpful to allow better visibility to all students. The final challenge can be handled in a similar fashion to other activities, by moving student responses to an electronic and anonymous platform.

9. Learning Classes with Stuffed Animals: Dr. Doolittle's Vet Office:

Students design the class Animal for vet office software using stuffed animals as a visual aid. The objective of this activity is for the students to determine what characteristics the class would need for a provided software specification [2]. To further students' understanding of classes, they will select what variables/data types and functions to use for a class to accurately represent several type of stuffed animals. The stuffed animals are used as a visual aid to help students decide what various animals have in common (or how they differ). This activity not only helps student understand the elements of a class, but also allows them to practice designing a class and making coding decisions. The activity is demo styled with students interacting with their direct neighbors for discussion throughout the activity. There are several challenges for this activity: (1) students being able to see the animals, (2) collecting student responses, and (3) providing feedback on student responses.

The first of the challenges is fairly easy to overcome. As suggested with other activities, a camera could be utilized to project the images of the animals on the overhead project so that students in the back of the room can see. One could also replace the stuffed animals with pictures of animals on a PowerPoint slide. Another option would be to set up several display boards / easels with pictures in the front of the room. The second challenge of collecting student responses could be handled in as previous activities by using an online response platform. Another would be to provide a series of possible answers (some correct, others incorrect) and have the students vote on the best ones. The final challenge of providing feedback to students could be handled by having students exchange papers and grade, have them work in teams and submit one final version, or have the teaching assistants on hand to work around the room and provide verbal feedback.

A completely different approach would be to divide the students into sections and spread the stuffed animals or pictures throughout the room. This would help eliminate the issue of students being able to see. All sections would work on writing an Animal class, but using different visual aids. After the students have discussed within small groups their solutions, the groups would exchange papers (across sections) to another group. The students would practice code review and learning to provide professional feedback. (On a different day before this activity, students would have already practiced reviewing code and providing feedback with instructor and teaching assistant written examples. Thus, at this point the students would have some practice with this important skill). This approach to the activity would provide students with practice working effectively on a team and providing constructive feedback.

10. Sorting Algorithms: Design an Algorithm with Paper Bags

Basic sorting algorithms are fundamental to beginning programming courses. In this activity the instructor places a series of bags (similar to the bags used in the first understanding variables activity) at the front of the room each with a number written on the front. Students will form groups and write an algorithm to sort the bags into numerical order. The bags at the front of the room will also be used as visual aids when explaining each basic sort algorithm (bubble, selection, and insertion) [2, 6, 7]. The challenges for this activity within a large classroom include: (1) students being able to see the bags, (2) forming small student groups, (3) working with a forum classroom, (3) moving within the classroom, and (4) providing students feedback on initial algorithm design. The first challenged can be combated by setting up multiple sets of bags at various locations within the room or utilizing a camera and the overhead projector to eliminate any issues with students being unable to see. The second, third, and fourth are remedied using the techniques mentioned in the other activities within this paper especially as students will be familiar with seating positions and group divisions this late into the semester. For the fourth challenge of supplying feedback, that could be handled in multiple ways. The first is to have students volunteer their algorithm in class. The second is to use an electronic submission and randomly select a few to test and evaluate. The third is to use the method outline in the Dr. Doolittle activity and have students exchange papers which reinforces effective teamwork and constructive feedback skills.

11. Recursion Introduction: Simple Tower of Hanoi with Colored Paper

This activity is used as a starting point to the discussion of recursion. Many students have seen this concept in computer or video games and many know how to solve a simple three disc problem. However, students generally do not know the algorithm behind these solutions or how to

design one that extrapolates to these larger numbers of discs. The activity was designed for each group to receive five different pieces of paper cut into circles (discs) and a paper that contains three squares (pegs). The students begin with three discs to design an algorithm (and use the discs to expand the solution to the more general case) [3].

The challenges for upscaling this activity to the larger forum style classroom are: (1) forming small student groups, (2) working with the forum classroom, (3) moving within the classroom, (4) receiving student feedback during the activity and (5) providing student feedback. The first three challenges can be addressed with techniques discussed earlier. As in the sorting algorithms activity, students will be familiar with group divisions late in the semester. However, it is important to note, that for this activity large group sizes do not work well which would require careful placement of the students within the classroom. For the fourth and fifth challenges, groups can exchange algorithms as discussed in the Dr. Doolittle class activity and test the new algorithm. This provides students another opportunity to practice giving constructive feedback. The instructors and TAs can also choose different solutions to demonstrate with the entire class, so that student teams could see different approaches to the same problem.

12. Summary

Although large class sizes and forum style classrooms present increased challenges to implementing active learning techniques, the potential benefits to student understanding of the material are worth the difficulties. Many of the challenges can be lessened with careful classroom management and early instruction and explanation to students. This can be accomplished with early, simple activities to allow students to become comfortable with the style of learning. Similarly, early use of electronic in-class submission tools can make receiving student feedback faster and familiarize students with the technology. Several techniques utilizing group work will require careful planning with teaching assistants and proctors, and some light feedback to discuss the best teaching techniques to use with student groups.

References

[[1] Prince, M. J. and Felder, R. M., Inductive Teaching and Learning Methods: Definitions, Comparisons, and Research Bases. Journal of Engineering Education, 95: 123–138, 2006.

[2] Coffman-Wolph, S., Innovative Activities to Teach Computer Science Concepts Inside the Classroom and at Outreach Events Paper presented at 2016 ASEE Annual Conference & Exposition, New Orleans, Louisiana. 10.18260/p.25715, June 2016.

[3] Coffman-Wolph, S., Fun, Innovative Computer Science Activities for the Classroom and Outreach Paper presented at 2017 ASEE Annual Conference & Exposition, Columbus, Ohio. https://peer.asee.org/28394, June 2017.

[4] Diefes-Dux, H., & Montgomery, R., & Oakes, W., & Follman, D. (2003, June), Tips For Teaching Obscenely Large Lectures Paper presented at 2003 Annual Conference, Nashville, Tennessee. https://peer.asee.org/11954

[5] Catton, K. B., & Galang, A., & Bulk, A. T., Disruption in Large Classes during Active Learning Sessions Paper presented at 2016 ASEE Annual Conference & Exposition, New Orleans, Louisiana. 10.18260/p.26856, June 2016.

[6] T. Bell, et al., “Computer Science Unplugged: School Students Doing Real Computing Without Computers,” Computing and Information Technology Research and Education, New Zealand (CITRENZ), vol. 13, no. 1, pp. 20-29, 2009.

[7] T. Bell, et al., CS Unplugged: Computer Science without a Computer. www.csunplugged.org., 2015.