TeamStorms as a theory of instruction

M. C. Jadud
Indiana University
Bloomington, Indiana 47405

Abstract

LEGO® TeamStorms is a team-centered, problem solving based approach to instruction, stressing active participation on the part of students in a constructivist learning environment. Not all students readily think in the abstract; however, this is not to say that they cannot. TeamStorms, combined with the LEGO® Mindstorms™ Robotics Invention System™, attempts to provide a learning environment where the abstract world of computer science can be made real and concrete to students, teaching them knowledge and process skills foundational to the discipline.

1 Purpose

TeamStorms is an instructional approach for introducing non-CS-majors to problem-solving, robotics, and computer science. This paper presents an overview of TeamStorms, a theory of instruction prescribing a team-centered, multi-disciplinary, problem-solving based approach to teaching at the introductory level in computer science and related disciplines.

Concrete The LEGO® Mindstorms™ Robotics Invention System™ is used as a motivator and learning manipulative, an object that students can put their hands on and interact with in their exploration of the material at hand. Loops, communication protocols, and event-driven programming move from the realm of the abstract to the concrete when students' ideas as acted out before them, with real consequences for misconceptions and confusion. Because of it's very nature, the medium enables students to test ideas and evolve them, providing an organic learning and rapid-prototyping environment.

Authentic Problem-based learning is a constructivist-leaning theory of learning and instruction that promotes the use of "authentic tasks" or "real-world problems" in challenging students.[Nel99] In CSCI A290 at Indiana University Bloomington, this is achieved by couching algorithmic and programming challenges in the physical world, using the Mindstorms™ Robotics Invention System™ as our platform for exploration and learning.

Cross-disciplinary TeamStorms has been developed in a liberal arts setting, and attempts to be as inclusive as possible regarding the types of learners it attracts. It draws heavily from computer science, philosophy, literature, and biology in its implementation. Basic principles of programming and computer science are introduced in the context of real-world problems students must solve; this is similar in many respects to Physics by Inquiry[McD96], Calculus in Context[CCH$^+$95], and many other instructional methods using problem-based methods that are actively being researched today.

Team-centered Working together to complete a task or solve a problem is a common and natural situation in today's society. Grouping students to work together in a laboratory setting provides opportunities for learning to work together as well as learning from each-other. Also, the peer-pressure and motivational factors involved in group work can be a powerful force in encouraging students to achieve.[CHL96]

2 Method

As an instructional theory, TeamStorms provides a template for what instruction should look like and how it should be executed.[Rei99] There are five methods specified by TeamStorms: Objective Formulation, Thinkorithms, Discussion, Mini-Lecture, and Play. These methods are broken into various sub-methods, and criteria for when they should be applied guide an instructor in using TeamStorms in their classroom.

Values

It is important when discussing instructional theories to be up-front about the values involved. Unlike learning theories, which discuss how we learn, instructional theories model how instruction should take place in the classroom or learning environment. Teaching is a personal matter, and if an instructor cannot agree with the values behind the instructional approach in question, then it is likely that the techniques perscribed will be a poor fit for that instructor. In short, TeamStorms values:

- Active responsibility for learning on the part of participants
- Focus on process as well as product
- Exploration and discovery in learning
- Peer-peer interactions in learning
- Authentic, or real-world, learning situations
- Fun

2.1 Objective Formulation

A module or lesson implemented using the TeamStorms approach might be anywhere from one to three laboratories in length, approximately 3-9 contact hours. An instructor runs the risk of loosing student interest if a single lesson is dragged out for too long; some judgement will need to be used in gauging what constitutes the right number of knowledge and process objectives for one lesson, based on the instructor's target audience. Those categories are further broken into passive and active parts.

Passive knowledge objectives

Passive knowledge objectives are those where the student needs to do some sort of research that does not require them to actively engage the material (and produce something through their labors), but instead simply *learn* material for later use. Common resources for achieving passive knowledge objectives include various print media (textbooks and articles), the WWW, and video recordings or movies.

Print-based media, while very prolific, should be used judiciously. Reading is typically a very inactive process, as students do not typically have good study or reading skills. The WWW suffers from this as well, and is further hindered by its unpredictable and distracting nature; students are easily led astray from their intended objective, either by hyperlinks or their own will. In either case, instructors should consider supporting students in using the media chosen, either with guides to good reading habits or URLs and keywords that might be of use.

Active knowledge objectives

Active knowledge objectives differ from passive objectives in that they are developed or discovered by the student in some way. This is most easily achieved by presenting students with a project or assignment lacking information that is key to the successful completion of the assignment. Making students aware of the content they are expected to cover and learning goals they are expected to achieve in the course of the project is perfectly acceptable.[Bal96]

Passive Process Objectives

While "passive process" may sound like an oxymoron, it is possible to cover process skils in a non-active way. For example, a student might learn some skill through reading or watching a video. While this is not necessarily authentic learning, it may sometimes be the only way for students to learn a process before putting it into practice because of limited time, equipment, or materials.

Typical computer science texts fall into a category of tools that commonly support "passive process." Programming is a problem-solving process, from design to implementation, and many texts do not take this fact into account when they are written. These critical thinking skills are foundational to success in the discipline, and an instructor must make judicious use of typical texts to assure that students are learning more than just content, but also how to think.

Active Process Objectives

Active process objectives are typically achieved fairly easily by assigning students tasks that require a particular process to be put into practice. This kind of authentic instruction is particularly valued by the TeamStorms approach.

An active process may be discussed or studied in advance in some way by the students, perhaps best developed as part of a Thinkorithm. If the process is not considered in advance of it being put into practice, but instead left to be discovered by the students, the instructor should be prepared to guide the students as they attempt to complete the module so they learn the processes involved properly. Improperly learned processes can have lasting ramifications for students, and as such the instructor should take care to design active learning situations carefully, or be prepared to catch misconceptions early on.

An important, yet often forgotten aspect of active learning is *reflection*. While hands-on experience is a powerful mechanism for learning, students need time to digest and reflect on what they have learned, and where it fits into their existing ream of knowledge and tools. Good active process objectives allow time for reflection, written or verbal, individually or as a group.

2.2 Natural Considerations

An important part of exploring new material is researching some of what has come before, or at least grounding the new material in the body of existing knowledge. Because of the importance of grounding the abstract in the concrete, and the cross-disciplinary nature of TeamStorms as a theory of instruction, tying the abstract process into some natural process is a good way to get at many abstract concepts in math, physics, and computer science.

Natural Considerations should support the students success in a given module either by augmenting their knowledge goals, process goals, or Thinkorithm in some way.

Familiar and Accessible

The area chosen by the instructor should be familiar to the students. In the event that the students choose their own areas to research, the instructor must keep a close watch on the topics chosen by the students, and evaluate their reasons for researching that topic.

Examples of areas that might be good places to draw research topics might include:

TABLE 1. Potential Topic Areas for Natural Considerations Biology Physics Ecology Chemistry Sociology Current Events Sports Video Games

Furthermore, the topic chosen for the Natural Consideration should be accessible to the student. The Natural Consideration topic should not be so obscure as to be completely foreign. An element of the unknown is acceptable, but the instructor should remember that grounding the abstract in the real world is of key importance in this theory.

self-directed

The Natural Considerations, perhaps most importantly, may provide additional avenues for individual research and learning for the students. What motivates an individual student is a hard thing to assess, and giving them some freedom to explore other topics while working through this process can be a a very important part of the students growth as a self-actualized learner. While staying on task is important, students passions should not be ignored, and the instructor may wish to provide some contingency plan for an individual or group of students who decide to explore a topic the instructor considers to be secondary or related to the module at hand.

Support A Diversity of learning styles

Researching the Natural Consideration has been discussed here, to a greater extent, as a traditional go-to-the-library-and-read-some-books-and-report-back-what-you-found process. This process works well for students who assimilate textual information well, or enjoy the topic enough to explore it from this angle. However, there are many different types of learners, and likewise, many types of research.

The instructor should attempt, if possible, to capture different kinds of research in their Natural Considerations. Perhaps the notions of looping or self-similarity could be explored by listening to musics, and identifying these structures therein. The concept of preventing deadlock in multi-user computer systems (where everyone ends up waiting for one scarce computing resource), may be explored by having groups of students attempt to do accomplish one task at the same time, and then have them analyze what problems arose and how they might actually solve those problems (thus turning an abstract process into a concrete physical process).

2.3 Thinkorithms

Thinkorithms play important roles in the TeamStorms theory of instruction, as they provide an active background to the material students will encounter in the module. Where many pre-lab exercises require students to read and (perhaps) answer questions, Thinkorithms require that students solve problems or brainstorm ideas related to the material they will face in the module, without actually tackling the core problems or challenges to be faced in the module itself.

A Thinkorithm can be considered either a *thinking algorithm*, or a problem that requires *thinking about algorithms.* Thinkorithms explore processes: these might be naturally occurring processes (like the water cycle), or artificial processes (like the steps taken by an airline pilot preparing for take-off).

Thinkorithms shoud try and support the knowledge and process objectives for a given lesson. This might involve embedding some prerequisite knowledge to the lesson in the Thinkorithm, or perhaps having students complete problems similar to those they will encounter in the lesson. Thinkorithms should attempt, in general, to be fun, challenging, allow for creativity on the part of the students, and be accessible or understandable. Furthermore, there is nothing that says a Thinkorithm cannot be open-ended, to be revisisted later or left as "food-for-thought."

The "travelling salesman" problem, while NPcomplete, could be an excellent example of a possible Thinkorithm topic. It might be couched as a problem of planning a road trip, or perhaps an excercise in energy conservation in travel. Similarly, navigating mazes, parsing data and communications, and numerous other topics offer a wealth of exciting avenues for exploration to introduce, motivate, and engage students before beginning a lesson or module.

2.4 Discussion

Discussion in the classroom can take many forms. It might be planned or spontaneous, used at the beginning or end of a class period, be used to probe students comprehension of material or thoughts on where to go with future efforts in their work.

Planned Discussion

A planned discussion is one where the students come to class prepared to discuss some question or topic. For example, students might come to class prepared to discuss some open-ended aspect of a Thinkorithm, both to talk about their solutions and discuss any other questions they might have. By placing such discussions at the beginning of class, the tone for the rest of the period can be set, and any important information that the instructor feels the students might need can be shared in a timely manner.

Discussion might fall during the class period, providing a break to students' efforts. This allows students to possibly share areas where they are stuck or confused, and then get feedback from their peers (or the instructor) regarding possible directions to proceed.

Lastly, a discussion might come at the end of a period, leaving the option of not only discussing work done outside of class, but also its impact on their efforts in class that day. This gives the students to reflect as a group, and gives the instructor the opportunity to asses their progress and readiness for new material.

In this modern age, it is also possible to move discussion entirely out of the classroom into the virtual, online world. This is a challenging prospect to say the least; many of the cues that a good moderator relies on in guiding a discussion are no longer present, and a completely new set must be learned. The media, however, does present the opportunity for more students to interact (with both the instructor and each-other) than typically can be accomodated during a typical class period.[Kri99]

Spontaneous Discussion

Not all discussions need to be planned. Being flexible in the classroom is necessary when students are exploring new material. Because an instructor is being flexible according to student needs does not mean that the instructional techniques they normally use need to fundamentally change; instead, they simply change when they use them.

Spontaneous discussions are a balence of "just-in-case" and "just-in-time" instruction. They are "just-in-case" in that the instructor must be prepared to lead a discussion on the topic or topics of interest. Similarly, they take on a "just-in-time" role when they are applied in a manner timely to student frustration or confusion in the classroom. This kind of flexibility can be very rewarding in the classroom, typically requires less preparation than a lecture, but may not work well for those who have a hard time "letting go," or "just going with the flow." [Bal96]

2.5 Mini-Lecture

There are times when there might not be a better way to present material to a class than to simply lecture on the subject. In the TeamStorms approach, this is the least desirable method for presenting material;an instructor should be striving to create a constructivist learning environment in which students develop their own representations and understandings of knowledge and processes involved in solving new or difficult problems. As such, lecture should be used sparingly, and this is why it is referred to as *Mini-Lecture*, because lecture should be kept short and topical.[1]

The timing of these Mini-Lectures is up to the instructor; beginning lab with lecture is comfortable for students, as they are accustomed to passive learning in most classrooms, where they come in and knowledge is imparted onto them by their instructor. While this is familiar to the students, it is not recommended that students are allowed to come to rely on this as the way all labs will begin. Mixing up the laboratory, and sometimes allowing students to dive directly into the material without interference from the instructor is a good thing, and should not be ignored as a way to begin the day.

Delivering lectures in a spontaneous manner (similar to that described under Discussion) is a potential way to integrate lecture fluidly into the laboratory experience. When students are particularly stuck, or lab slows down for one reason or another, lecture might

[1] The author's father delivered countless 'mini-lectures' during his lifetim. They were referred to as such.

be a more effective way for dealing with the students' problems than a discussion. Keep in mind that problem or challenge for the students is not your goal ideal - a spontaneous Mini-Lecture should not "give away the punchline." Instead, it should provide hints and tips for bridging the gap between where the students are and where the instructor would like them to be.

2.6 Play

Play is the word TeamStorms uses for the classroom experience. Too often, class time is associated with repetitive tasks or (most often) lecture. Unfortunately, students who loose interest think it is the *material they are uninterested in.* It takes a very self-aware learner to differentiate between being uninterested in the material and the way the material is presented. An instructor implementing TeamStorms should treat class time, as much as possible, as the students' time to explore the material in question. Allow time for unstructured learning and exploration. Mix up the class schedule from week to week so students do not fall too much into a routine - schedule discussion or lecture at different points in the period so as to keep students from becoming bored with the instruction itself.

Many of the larger techniques an instructor might use during class-time have been covered - Discussion and Mini-Lecture being two of them. Many smaller points for interaction with students in the classroom can be made, a few of which are presented here:

- **Success Early, Success Often** Give students successes early in the course to build on; confidence and determination stemmed from success will overcome frustration in the learning process.
- **There are no Stupid Questions** Encourage questions at all times, and never condescend to a student for their apparent lack of knoweldge or understanding. At the same time, however...
- **Answer Questions with Questions** Unless the student's question is truly beyond their scope or expertise, get into the habit of making them *think*. This is sometimes referred to as the Socratic method, and approaches like Physics by Inquiry implement it quite rigorously.[McD96]
- **Build Respect** Neither students nor instructors like to be disrespected.
- **Use Analogies** Use analogies whenever possible to bring new, abstract material into the students' world.
- **Working Together is Not Cheating** Students stand to learn a great deal working together, as well as gain valuable group process skills to be applied later in their academic and professional careers.

3 Results

TeamStorms has been successfully applied to graduate courses in education at Indiana University Bloomington and Boston College, a high-school introductory programming course, and both introductory and sophomore-level courses for non-majors in the computer science department at Indiana University.

expand this section a bit - focus on the independent study work

Anecdotal evidence and informal interviews indicate that one in fifteen students have declared majors in the Department of Computer Science as a direct result of their experiences with these methods and the material. Women and continuing students have found this to be an appealing way to approach the discipline. We have had good success moving undergraduates into mentorship roles and independent study positions involving robotics work. Because TeamStorms is so young, further research regarding these results is warranted.

4 New or breakthrough aspect of work

TeamStorms draws from the rich body of literature on constructivist learning and multiple intelligences - the work of Dewey, Bloom, Minsky, Gardner, and numerous others. TeamStorms differs from the body of research pertaining to learning theory because it attempts to provide a template for instruction, a practical guide for applying the work of learning theorists to the classroom. It prescribes methods for situations an instructor might encounter in the classroom, and criteria advising when to apply those methods.

5 Conclusions

TeamStorms provides a framework for making abstract notions in robotics and programming concrete and tangible to students. Problem-based discovery methods are very supportive of many learning styles, and help make robotics and programming approachable to a diverse audience of learners. The LEGO® Mindstorms™ kit has played an important role as a motivational tool in this approach to teaching problem-solving and teamwork in the context of robotics and programming.

http://www.indiana.edu/ legobots/index.html

Acknowledgments

Many thanks to the continuing support of Profs. Jonathan Mills, Tom Keating, Charles Reigeluth, and the Departments of Computer Science at Indiana University.

References

[Bal96] Doug Baldwin. Discovery learning in computer science. *Proceedings of the twenty-seventh SIGCSE technical symposium on Computer science education*, pages 222–226, 1996.

[CCH+95] James Callahan, David A. Cox, Kenneth R. Hoffman, Donal O'Shea, Harriet Pollatsek, and Lester Senechal. *Calculus in Context*. W. H. Freeman, 1995.

[CHL96] Curtis Carver, Richard A. Howard, and William D. Lane. A methodology for active, student-controlled learning: motivating our weakest students. *Proceedings of the twenty-seventh SIGCSE technical symposium on Computer science education*, 1996.

[Kri99] Malini Krishnamurthi. Integrating e-mail in a programming class: implications for teaching programming. *Proceedings of the 1999 ACM symposium on Applied computing*, pages 126–131, 1999.

[McD96] L.C. McDermott. *Physics by Inquiry*. John Wiley and Sons, 1996.

[Nel99] Laurie Nelson. *Instructional Design Theory and Models*, volume 2, chapter Collaborative Problem Solving, pages 241–267. Lawrence Erlbaum Associates, 1999.

[Rei99] Charles M. Reigeluth. *Instructional Design Theory and Models*, volume 2, chapter What is Instructional-Design Theory and How Is It Changing?, pages 5–29. Lawrence Erlbaum Associates, 1999.