2023 Annual Conference & Exposition
Baltimore Convention Center, MD | June 25 - 28, 2023
The Harbor of Engineering Education for 130 Years
Paper ID #37351

Work in Progress: Robotics Programming Made Inclusive, Motivating, Enabling via Alternative Forms of Assessment

Lisa Milkowki, Seattle University

Lisa Milkowski is an Assistant Teaching Professor in the Department of Computer Science at Seattle University. She obtained her BS in Biomedical Engineering at Milwaukee School of Engineering (MSOE), and her MS and Ph.D. in Biomedical Engineering at Marquette University. She then held Assistant and Associate Professor positions at MSOE in the department of Electrical Engineering and Computer Science. She taught courses in a variety of areas including signal processing, medical imaging, computer science, and biomechanics. Her research interests included kinematic modeling, medical instrumentation, data analysis, and biomechanical modeling using precision 3D printing. She has served in various officer roles in the Biomedical Engineering Division of ASEE. After a move to Seattle, she began working at Seattle University teaching computer science and expanding opportunities for students in robotics.

Robotics programming made inclusive, motivating, and enabling via alternative forms of assessment WIP

Introduction

The opportunities for computer science (CS) graduates are strong in most regions of the country. Graduate certificate programs in CS have gained popularity with non-STEM bachelor's degree students. Many of these students go on to pursue an MS CS degree. Of the students entering the MS program from the certificate program, up to 60% have non-STEM undergraduate majors. The university's objective to increase graduate enrollment has driven the CS department to offer more courses to meet the educational needs of this population.

Researchers sought to determine whether a new robotics programming course could assist non-STEM undergraduate majors pursuing a graduate CS degree and underrepresented populations in transitioning to CS. The course development aimed to integrate more inclusive pedagogical practices. This paper presents the ongoing work in developing and evaluating the effectiveness of the new robotics programming course.

Literature Review

There are many K-12 initiatives involving robotics hardware and programming that encourage students to pursue STEM professions [1] - [2]. Additionally, there are undergraduate-level courses in robotics [3] – [7] typically designed to enhance motivation for students majoring in STEM professions. Barba et al. [8] present the design of two graduate courses for non-majors, adult learners, and non-traditional students. The courses use Pixelsense and Arduino to teach computational thinking, programming, and design skills. The authors specifically mention the importance of platform choice, assignment structure, maintaining student motivation, and the impact of self-guided final projects. Farah et al. [9] similarly address the needs of non-STEM majors by presenting work developing computational thinking via a single web application. This approach requires no software installation and minimizes the challenges of working with multiple applications including integrated design environments, digital education platforms, and file system management.

This paper presents a course building upon student competency in computational thinking acquired during prerequisite work. These students expand their learning and expertise to integrate various applications and technology stacks through robotics. Developing the ability to integrate contributes to both student satisfaction and professional competency.

Robot programming requires the integration of multiple technical subject areas including connecting to hardware, working within hardware limitations, basic architectures, open-source development environments, artificial intelligence, and cloud computing. Students enrolling in the robotics programming course are familiar with some, but not all these technical areas. Ambrose et al. [10] describe how students' prior knowledge can help or hinder learning. This is especially challenging in the context of multiple subject areas and inclusivity.

To address these challenges, Ambrose et al. [10] examine learning theories supported by empirical research evidence and strategies to promote goal-directed and intrinsic motivation. The first component involves building self-confidence in taking risks and persisting through multiple attempts in unfamiliar subject areas. Traditional exam assessment models underutilize hands-on practice in a classroom environment. By gaining experience before entering the professional world, students benefit and feel more comfortable trying new endeavors.

Another component is addressing and fostering motivation. Although students are ideally intrinsically motivated to learn, practical needs to perform well academically and conflicting motivations from other areas of their life can compete with their learning motivation. To counteract these tendencies and accentuate motivation, students should be provided with flexibility and control to make choices consistent with their course objectives. Flexibility and control allow students to choose activities or projects they identify with and value, and bridge to the prerequisite knowledge they are most comfortable connecting to. This is particularly effective in providing options to students with diverse learning styles and traditionally underrepresented students, who may be disadvantaged by the traditional exam form of assessment [11].

Proposed Solution

The proposed solution for enhancing the effectiveness and inclusivity of an introductory robotics programming course for underrepresented populations is described below [11].

- Control and flexibility in assessment
- Clear course objectives
- Learner-centered growth mindset
- Sense of belonging
- Multiple means of engagement
- Equitable participation and recognition of student differences
- Mitigating stereotype threat and affirming student identities
- Encouraging everyone to play a role in the learning process
- Continual course improvement through feedback

To address the lack of prerequisite knowledge in robotics hardware and programming among non-STEM students, the course was designed without robotic hardware prerequisites. Since certificate courses introduced students to Linux command line programming and C++, the robotics course utilized robotic hardware simulation instead. Simulations were driven by ROS open-source software relying on knowledge of C++ programming and Linux command line commands and pathways. Financial costs were kept low by utilizing open-source software and free online learning resources and textbooks.

To reward motivation, risk-taking, and persistence, a claw-back incentive was established. The claw-back incentive provides an exception to the final exam, which is granted and explained on the syllabus on the first day of class. The incentive for performing well on the robotic programming project will be revoked or 'clawed back' when the performance metrics are not met. Research by economist John A. List found it to be effective in motivating individuals [12], [13].

To enhance inclusion, the first day of class was dedicated to establishing a sense of belonging and a learner-centered growth mindset. Students were informed of the risks associated with offering the course and the opportunity it provided for learning and growth. The traits of highly successful people were discussed, and students were encouraged to incorporate these traits into their approach to the course. The importance of choice was emphasized in achieving success and happiness, and students were encouraged to take the same approach [14], [15].

Within the first two weeks, each student was asked to share their background with the class. The instructor modeled this by sharing their atypical background, and students were encouraged to share their unique backgrounds that likely did not originate from an initial STEM background. Participation was affirmed in class and through blogging, and students were expected to be supportive of their classmates in both forums. Any hurtful expressions were addressed, and students were encouraged to escalate them to campus faculty resource centers if necessary.

The course was structured into approachable modules, with shorter textbooks matched to the sequencing. The course started with an introduction to robot basics including sensing, actuation, planning, and control [16]. The course progressed into programming and architectures including reactive control, deliberative control, and hybrid architectures [17]. Lastly, project work explored topics such as robot operating systems (ROS) [18], [19], robotic simulators [20], and cloud robotics [21].

Overall, the proposed solution emphasizes the importance of designing inclusive practices that provide multiple means of engagement, while setting clear expectations and providing motivation for students to achieve their goals.

Overview of Course

The course had the following learning outcomes articulated on the syllabus [22]:

- Understand basic hardware and software components of a robotic system and variations in robot designs including cloud robotics systems.
- Be able to articulate limitations of robotics systems related to limitations of hardware capabilities.
- Understand the basic concepts of robotics related to software's role within a robotic system including robot architecture.
- Perform software development using a common robotics software development platform such as the Robotic Operating System (ROS) using widely available libraries of common robotic tasks and functions.
- Demonstrate the ability to implement software designs using a common robotics simulator or physical robot platform.

Assignments in the course include blog postings, attending a virtual or in-person professional conference and blogging about the experience, and a robotics programming project that meets specific criteria related to the course outcomes.

The project assignment was arranged to encourage continual progress and included:

1. Project proposal

2. Progress report
3. Demonstration
4. Technical documentation
5. Final presentation
6. Project slides

The final exam in the course is presented as a claw-back, meaning it is only necessary if a successful demonstration of the final project is not possible due to extenuating circumstances such as technical limitations. Limitations will be required to be documented in a Programming Project Report and Presentation. The final exam is not a substitute for the final project.

If a student uses the claw-back incentive and does not take the final exam, the project is worth 50% of their grade (figure 1). If they take the final exam, 25% of the final exam goes toward their final grade and 25% toward the project.

Figure 1: Weighting of assessments to compute the final grade

Homework and Assignments and Quizzes	25%
Individual or Team Robotics Programming Project	50% (25%)
Participation in class discussion/contribution	25%
Claw-back Final Exam	(25%)

Implementation

Assignments started with blog postings within our private Canvas course answering assigned questions. Questions were assigned randomly with students being assigned different questions. Example questions are "Does the guidance system of a ballistic missile exhibit servo or ballistic behavior?" and "The escape behavior diagrammed in the figure doesn't specify what happens if both the left and right bump switches activate simultaneously, as they would if the robot hits an object dead center. What would or should happen in this case? Draw an FSM diagram that incorporates the new functionality." [17].

Before a student blogs, they study the content and develop an understanding. If they are not successfully engaging with the content, they have the option to formulate a question about what it is that they don't understand. The deadlines for the blog postings keep the students progressing in the course like other assignments. The instructions students received for blog postings are shown in the appendix (Figure 3).

Assigned blogging continued with an expectation to attend a virtual or in-person professional conference and again blog on your experience. We were fortunate to have two local robotics conferences during the course offerings and attend those in person. One of these conferences featured robotics work at Amazon and robotic development within their warehouses. Alternatively, recorded professional conference presentations could also be watched if virtual or in-person options are not available.

Students pick their robotics programming project, but their project needs to meet criteria related to the course outcomes. The instructions were given to students for developing their project proposal as shown in the appendix (Figure 4).

Progress on the project is then expected and documented by the student in their progress report. The instructions students received for the progress report are again included in the appendix (Figure 5).

Project progress continued to the demonstration of successfully operating software in simulation or on a physical robot. The project demonstration was an informal presentation to the class. One of the goals of this assignment was to finalize the technical work of building the successfully operating software. Additionally, the objective was to highlight the successful integration of student-authored software elements with open-source library software elements.

The rubric used to assess this demonstration is combined with the assessment of the technical documentation. It is shown in the appendix (figure 6). The technical documentation is another project milestone. The document is due a few days after the informal demonstration to distribute the tasks and work for the students over different class sessions.

The project presentation, where students gave 12-minute presentations, marked one of the final steps in the project progression. Students who did not successfully demonstrate working software at the previous milestone had one more chance at this presentation to demonstrate the software. If they were still unsuccessful, they would have to take the final exam to show mastery of the course outcomes. The rubric for this presentation addresses presentation skills because other assignments were assessments of project technical work and documentation. The rubric is included in the appendix (figure 7).

Finally, students were required to upload their presentation slides, which would aid the instructor in the assessment. Slides and technical documentation are helpful documents to confirm the oral presentations and demonstrations align with the documented software code. A separate rubric was not used for the slide submission but was part of the paper presentation rubric.

Evaluation

Assessment of student learning was conducted through project demonstrations and presentations, with the possibility of incorporating peer evaluations in the future. The course's inclusivity, accessibility, and ability to support diverse learners were evaluated by analyzing the course evaluations. Figure 2 presents the course evaluation responses from students. Course completion rates and enrollment trends will be examined once enough students have participated.

Figure 2: Course evaluation responses of the students

3 - The course as a whole was well-organized.

Response Option	Weight	Frequency	Percent	Percent Responses
Strongly Agree	(5)	2	40.00%	
Agree	(4)	3	60.00%	
Neutral	(3)	0	0.00%	
Disagree	(2)	0	0.00%	
Strongly Disagree	(1)	0	0.00%	
				0 25 50 100

Response Rate	Mean
5/5 (100.00%)	4.40

5 - The instructor's attitude and teaching style encouraged my learning.

Response Option	Weight	Frequency	Percent	Percent Responses
Strongly Agree	(5)	4	80.00%	
Agree	(4)	1	20.00%	
Neutral	(3)	0	0.00%	
Disagree	(2)	0	0.00%	
Strongly Disagree	(1)	0	0.00%	
				0 25 50 100

Response Rate	Mean
5/5 (100.00%)	4.80

7 - The instructor appropriately assessed learned skills through exams, reports or other materials.

Response Option	Weight	Frequency	Percent	Percent Responses
Strongly Agree	(5)	2	40.00%	
Agree	(4)	3	60.00%	
Neutral	(3)	0	0.00%	
Disagree	(2)	0	0.00%	
Strongly Disagree	(1)	0	0.00%	
				0 25 50 100

Response Rate	Mean
5/5 (100.00%)	4.40

- I felt handling the discussion questions each week, while we had them, was a powerful way to tackle the material.

8 - I gave my best effort to achieve the course objectives, i.e. attendance, assignments, reading and s

Response Option	Weight	Frequency	Percent	Percent Responses
Strongly Agree	(5)	4	100.00%	
Agree	(4)	0	0.00%	
Neutral	(3)	0	0.00%	
Disagree	(2)	0	0.00%	
Strongly Disagree	(1)	0	0.00%	
				0 25 50 100

Response Rate	Mean
4/5 (80.00%)	5.00

- Being launched into the deep end with the project was one of the more gratifying and challenging tasks I've had at SU.

Conclusion

At the time of writing, the number of students surveyed was limited, and observations are noted until more data is collected from a larger sample size.

1) The course's ability to motivate students and foster interest in robotics programming was evident.

2) The real-life examples presented during the conference were inspiring and motivating for students.

3) Doing abstraction of software units in other courses such as an object-oriented design course may make learning how to program in this multi-disciplinary setting of robotic systems more approachable because that abstraction can be extended to hardware units.

4) An emphasis on learning through integration and the reuse of other ROS libraries was a helpful step in breaking down the steep entry to some success before programming new components.

5) Technical difficulties with software installation and running graphics intensive simulations on student laptops were addressed by providing university workstation computers or computer systems for student use. Alternatively, a requirement could be put in place requiring the student to provide a laptop with a sufficient graphics card and computing power.

6) The instructor's effort required to organize the new approach is comparable to that of traditional assessment approach courses. For blogging assignments, it could become difficult to read and monitor all the blog posts with large enrollment. This could be managed with a grader's assistance. Project proposals need to be approved by the instructor and tracked for progress from the proposal to the demonstration for each student/team. However, this is exchanged for no long hours writing and grading their midterms or final exams but instead listening to their final presentations over a couple of class periods. Additionally, cheating issues are unlikely to arise when orally presenting and being expected to document libraries or code you are expected to reuse.

As a work-in-progress, more metrics examining student motivation and inclusivity will be pursued. Data from student course evaluations and surveys were analyzed to evaluate course inclusivity and motivation. Course completion rates and enrollment trends will be examined once enough data are available for comparison. Enrollment trends and patterns among different student populations will also be investigated when more students from underrepresented populations have enrolled in the course.

References

[1] N. Eteokleous and E. Nisiforou, *Designing, Constructing, and Programming Robots for Learning*. Hershey, PA: IGI Global, Information Science Reference, 2022.

[2] A. Master, S.Cheryan, A. Moscatelli and A. Meltzoff, "Programming experience promotes higher STEM motivation among first-grade girls," Journal of Experimental Child Psychology, vol. 160:92-106, 2017.

[3] P. Mosley, Y. Liu, S. Hargrove, and J. Doswell, "A Pre-Engineering Program Using Robots to Attract Underrepresented High School and Community College Students" Journal of STEM Education, Vol 11.5/6 pp.44- 2010.

[4] C. Luo, J. Wang, W. Zhao, and L. Wang, "Multi-Lab-Driven Learning Method Used for Robotics ROS System Development," in Proceedings of the 2017 ASEE Annual Conference & Exposition, Columbus, Ohio. 10.18260/1-2—28692.

[5] A. Yousuf, C. Lehman, M. Mustafa and M. Hayder, (2015, June), "Introducing Kinematics with Robot Operating System (ROS)" *in Proceedings 2015 ASEE Annual Conference & Exposition, Seattle, Washington.* 10.18260/p.24361

[6] S. Wilkerson, S. Gadsden, A. Lee, R. Vandemark, E. Hill and A. Gadsden, "Board 64: ROS as an Undergraduate Project-based Learning Enabler" *in Proceedings of ASEE Annual Conference & Exposition, Salt Lake City, Utah*. 10.18260/1-2—30078

[7] Y. Chang, Y. Wang, and Z. Zhang, "A project-based platform for students' Robot Operation System (ROS) programming experience" *in Proceedings 2022 ASEE Annual Conference & Exposition, Minneapolis, MN.* Available: https://peer.asee.org/41272

[8] E. Barba, and S. Chancellor, "Tangible Media Approaches to Introductory Computer Science" *in Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*, 2015 ACM, pp. 207–12, https://doi.org/10.1145/2729094.2742612

[9] J. C. Farah, A. Moro, K. Bergram, A. K. Purohit, D. Gillet and A. Holzer, "Bringing Computational Thinking to non-STEM Undergraduates through an Integrated Notebook Application." *European Conference on Technology Enhanced Learning* (2020).

[10] S. Ambrose, M. Bridges, M. DiPeitro, M. Lovett, M. Norman, *How Learning Works: 7 Research-Based Principles for Smart Teaching*. San Francisco, CA: John Wiley & Sons Inc., 2010.

[11] T. Addy, D. Dube, K. Mitchell, and M. SoRelle, *What Inclusive Instructors Do: Principles and Practices for Excellence in College Teaching.* Bloomfield: Stylus Publishing LLC, 2021.

[12] G. Harrison, J.A. List, "Field Experiments", *Journal of economic literature*, vol.42.4, pp.1009-1055, 2004

[13] E. Fehr and J. A. List. "The Hidden Costs and Returns of Incentives-Trust and Trustworthiness among CEOs." *Journal of the European Economic Association*, vol. 2, no. 5, pp. 743–71, 2004.

[14] T. Ferriss, *Tools of Titans : The Tactics, Routines, and Habits of Billionaires, Icons, and World-Class Performers*. United States: Billionaire Mind Publishing, 2017.

[15] S. Covey, "The 7 Habits of Highly Effective People." *Dallas business journal* vol. 23.6, 1999.

[16] M. Mataric, *The Robotics Primer*. Cambridge, Mass: MIT Press, 2007.

[17] J. Jones and D. Roth, *Robot Programming: A Practical Guide to Behavior-Based Robotics*. New York: McGraw-Hill, 2004.

[18] J. O'Kane, *A Gentle Introduction to ROS*. Columbia, SC: Jason M. O'Kane, 2014.

[19] Robotic Operating System. Available: ROS.org

[20] Gazebo simulation reference. Available: https://roboticsimulationservices.com/ros-gazebo-everything-you-need-to-know/

[121] AWS RoboMaker Robotics Curriculum. Available: https://github.com/aws-robotics/aws-robomaker-robotics-curriculum

[22] Richard M. Felder, Donald R. Woods, James E. Stice, Armando Rugarcia, "The Future of Engineering Education II. Teaching Methods that Work." *Chemical Engineering Education*, vol. 34(1), pp.26-39 (2000).

Appendix

Figure 3: Blog posting instructions
Write an answer or response to each question you are assigned. Please number or label your responses so that we all can find responses to the various questions. Please also restate the question in your answer. You may skip one of the questions if you would prefer to write a specific or general question from the chapter. Everyone comments on the answer from one other person from the class. Your comment may include that you found a different answer helpful. However, your response will add something to the comments or responses. Stating a like to someone else's answer to a question is not enough. Your comment could be a general or specific question. These answers and comments will be the starting point for our next class period. They will also help the instructor tailor future coursework to best aid everyone's learning. Thanks for your willingness to share your thoughts!

Figure 4: Instructions for the project proposal
The goal of the robotics programming project is that you perform software development using a common robotics software development environment such as ROS. You are expected to use widely available libraries for common robotic tasks and functions. Remember however that your project is an *extension* of existing libraries and functions. It cannot simply be a demonstration of someone else's authored work. For example, downloading someone else's work from Github and demonstrating it is NOT enough. You must add some unique functionality besides just utilizing the libraries and functions. For example, you may use two different libraries. One that determines the location of a target and another that determines how to navigate to the target. Your project and program would do the work of combining the work of the two (and the logic) to get these two libraries to accomplish an overall task.

Try to be specific in your proposal. Don't worry if things need to change later. It is a proposal and some changes are likely to happen with forecasting.

Required Elements:

- Use of at least one widely available library or function
- Use of ROS or equivalent
- Use of a simulator such as Gazebo or physical robot to demonstrate your project
- Synthesis of new functionality beyond libraries or functions

Please summarize your project proposal in a Word or PDF document with the following sections.

Introduction

- Description of the Project being proposed
- Description of libraries along with links /references
- Description of new functionality beyond the libraries /functions

Description of Methods of Development. For example:

- Download libraries and write a test program to confirm its works independently
- Write a program to do simple communication with the library
- Write a program to integrate my logic and communication with the library

Optional: Results at this point

Appendix: Summary of software used with version numbers. Examples: VMWare, Ubuntu, ROS 18.04, Gazebo, etc.

Optional code snippets and figures are highly encouraged for clarity throughout the document.

I anticipate this document being 500 - 1000 words.

Figure 5: Instructions for the progress report

The goal of this progress report is that you show evidence of progress toward completing your Robot Project as measured by the required elements.

Please see the Robotics Project Proposal instructions if you need to review the goals of the robotics programming project.

Try to be specific in your progress report. You may 'add-on' to your project proposal with more information added. Likely some sections will now have more detail; you can begin filling the appendices and results tables, etc. The rubric used for your 'Final Project Submission' is listed below. Making reasonable progress toward the final project submission and presentation is the subject/criteria of this assignment.

Required Elements:

- Use of at least one widely available ROS library or function
- Use of ROS or equivalent
- Use of a simulator such as Gazebo or physical robot to demonstrate your project
- Synthesis of new functionality beyond libraries or functions
- (Optional) Successful demonstration of your operating robotics project

Please summarize your project proposal in a Word or PDF document with the following sections.

Introduction (More detail added for the libraries and new functionality descriptions).

- Description of the Project proposed
- Description of libraries along with links /references
- Description of new functionality beyond the libraries /functions

Description of Methods of Development. (This section may have been a 'list' in the proposal. Now it likely is filling out more to a detailed paragraph description of the methods.

Results at this point. This section was optional in the proposal but now you should have some preliminary results to present.

Appendix: Summary of software used with version numbers. Examples: VMWare, Ubuntu, ROS 18.04, Gazebo, etc.

Code snippets and figures are highly encouraged for clarity throughout the document.

I anticipate this document being 1000 - 1500 words.

Figure 6: Final Robotics Project Rubric

Final Robotics Project (1)

Criteria	Ratings		Pts
Successful operation and demonstration of your robotics project (Meaning your project works and can be demonstrated). Your project may be demonstrated on a robotics simulator such as Gazebo or on a physical robot such ad the Turtlebot being assembled in the Robotics Lab.	5 pts Full Marks	0 pts No Marks	5 pts
Complete technical DOCUMENTATION and clear technical communication including: Documented code (inclusion of code) clear descriptions, detailed descriptions appropriate for other members of class to understand your work, organized documents and presentations with all required sections, professional conduct	5 pts Full Marks	0 pts No Marks	5 pts
Use of appropriate ROS open source libraries and other existing libraries, algorithms, functions. Minimal requirement is use of at least one existing library such as ROS SLAM, MoveIt, etc.	5 pts Full Marks	0 pts No Marks	5 pts
Software development of unique or novel elements of your project that are authored by you. This work may be some form of novel extension of existing libraries or functions. For example, downloading someone else's work from Github and demonstrating it is NOT sufficient. You must add some unique functionality besides just utilizing the libraries and functions. For example, you may use two different libraries. One that determines the location of a target and another that determines how to navigate to the target. Your project and program would do the work of combining the work of the two (and the logic) to get these two libraries to accomplish an overall task.	5 pts Full Marks	0 pts No Marks	5 pts
			Total Points: 20

Figure 7: Robotics Formal Presentation Rubric

Paper Presentation Criteria

Criteria	Ratings			Pts
Introduction -- strong opening, including sufficient context to understand paper's research objective Includes paper objective and background information matched to audience	2 pts Comprehensive but concise	1 pts Missing information or too long	0 pts Missing or incomplete	2 pts
Organization and content -- well organized and all expected components included	2 pts Structure enhances understanding	1 pts Adequate organization	0 pts Disorganized or confusing	2 pts
Closing	1 pts Effective high level summary	0.5 pts Rushed or repetitive	0 pts Missing or incomplete	1 pts
Presentation materials: clarity of slides, appropriate number of slides, smooth transistions during talk	1 pts Full Marks	0 pts No Marks		1 pts
Length of presentation within 1 minute of target time	1 pts Full Marks	0.5 pts Just a little too short or too long	0 pts No Marks	1 pts
Technical paper content represented clearly and accurately	1 pts clear, complete, accurate	0.5 pts aspects not represented or inaccurate	0 pts Missing or incomplete	1 pts
Style of Delivery: Articulation, speaking confidently, appropriate speed, good expression. Good use of gestures and looking at audience or camera Articulates clearly. Speaks confidently (no filler words or long pauses; loud enough [in person] looks at entire audience and only faces screen when indicating particular items of interest; [virtual] mostly looks at camera rather than down at slides or notes Move a little, but avoid distracting motions like twisting or	2 pts Effective	1 pts delivery missing couple of features	0 pts very difficult for audience to understand	2 pts