Robots and Art:
Interactive Art and Robotics Education Program in the Humanities

Elizabeth Ann Jochum
Department of Communication and Psychology
Aalborg University (AAU)
Aalborg, Denmark
jochum@hum.aau.dk

Lance Putnam
Department of Architecture, Design and Media Technology
Aalborg University (AAU)
Aalborg, Denmark
lp@create.aau.dk

***Abstract*—We describe the design of an undergraduate course in art and robotics that aims to integrate basic concepts of computer science, robotics and art installation for undergraduate students within the problem-based learning model. Our methodology aims to bridge the gap that separates humanities from computer science and engineering education to prepare students to address real world problems in robotics, including human-robotic interaction and HCI. Given the proliferation of interactive, systems-based art works and the continued interest in human-centered factors in robotics research (such as aesthetics, culture and perception), we believe this is an important area for education and research.**

Keywords—robotics; interactive art; education; pedagogy; human-robot interaction; HCI

I. Introduction

The problem-based learning (PBL) model at Aalborg University (AAU) provides a unique framework for developing a transdisciplinary course that prepares students to combine critical thinking and problem-solving skills with hands-on experiments and practice. The Art and Technology (ArT) undergraduate program at AAU is unique in that it offers a cross-disciplinary background in both new media art and pertinent technical subjects such as electronics, programming and rapid prototyping. Unlike traditional fine arts programs, ArT provides students with the competencies required to deeply engage with the latest technologies and to translate creative theories and approaches into practical results.

We were motivated to develop a course that exemplifies the educational vision of ArT and to provide a clear progression and integration of previous ArT courses. Multimedia Programming-Autonomous Art (MMP-AA) integrates art and technology by focusing specifically on the theoretical and practical aspects of robotic art. The course places equal emphasis on both aesthetic and technical concerns so that students may develop competencies in the creation of aesthetically engaging autonomous art works.

While there do exist some undergraduate programs that combine art and computer science education [1], many of these curriculums are centered on electronic and/or digital media arts and do not directly incorporate robotics. Robotics education remains out of reach for many students not enrolled in traditional computer science or engineering programs. Therefore, a second aim of MMP-AA was to generate interest and enthusiasm for robotics research among humanities students. The semester theme of “Narratives and Interaction” provided the context for the course and we encouraged students to incorportate this theme in their projects, for example by combining robotic art with storytelling and interactivity. Students were asked to develop original research projects that combined mobile robotics with the creation of an original artifact, performance or installation.

We incorporated several modes of evaluation into the course, including student entrance and exit surveys, video recording, and project documentation. Here, we present the data including course development and curriculum, analysis of content and student projects, evaluation, and plans for future research.

II. The Art and Technology Context

A. *Transdisciplinarity*

One of the pedagogical challenges at ArT is to select subjects that apply to a wide range of media and at the same time have direct applications within those media. In other words, students should not only be familiar with the languages and research methods across disciplines but should also be able to integrate diverse fields of knowledge to solve real world problems. The boundaries that have traditionally separated humanities research from computational research are increasingly blurred, and students should be prepared for this new landscape. Transdisciplinary thinking requires one to make abstractions on top of domain-specific knowledge. A significant obstacle to transdisciplinary teaching is to choose both the most effective abstractions and the clearest language to communicate such concepts. This is the challenge we address in MMP-AA.

B. *Multiple Languages*

The ArT curriculum involves a unique mix of theory and practice, and students take courses across a wide range of subjects including art history, sculpture, dynamic art, interactive systems programming, play and event, and entrepreneurship. While the program offers a wide range of courses across topics, few courses formally combine aesthetic theory and practice with programming and technology. The languages of art and programming do not readily translate across disciplines, and it is usually left to the students to combine these two fields in their individual projects. The course theme of "Languages of Motion" was an effort to bring computational motion and aesthetic motion into closer alignment. Furthermore, many of the technologies taught in ArT involve software programs for screen-based media. By focusing on robotic art, we hoped to expand the students' awareness to include physics-based scenarios, encouraging students to experiment with new approaches to automated motion such as choreography.

C. Demographics

Thirteen students enrolled in MMP-AA. There were five male students and eight female students, ranging in age from 22-29 years old. All of the students were third year (fifth semester) ArT bachelor students, and had varying levels of programming experience outside of the ArT curriculum. Most students had basic programming skills including C++, HTML/CSS, and Java. Prior to MMP-AA, students had completed courses in basic electronics, materials (including structure and composition), and digital representations (including laser cutting and 3D printing). One student had some prior experience working with Lego Mindstorms, but beyond that none of the students had previous experience working with robots. Most of the students had seen some robotic art works, but their familiarity with historical and contemporary robotic art and robotics research was limited.

III. Pedagogical Approach

This section outlines our pedagogical approach, including the teaching formats, curriculum and learning objectives.

A. Co-Teaching

Our teaching is based on a collaborative model where teaching, assignments and evaluations are developed and implemented collaboratively at the faculty level. Our appointments are at the Faculty of Humanities and the Faculty of Engineering and Science respectively, which further strengthens the transdisciplinary foundation of the course and demonstrates to the students ways to foster fruitful collaboration between humanities, computer science and engineering. We also decided that both professors should be present at all lessons in order to facilitate communication across disciplines.

B. Lecture/Workshop Format

We deliberately structured the course to balance the time between lectures and hands-on workshops as well as aesthetic and technical considerations. The course consisted of eight lessons in total with each lesson lasting two hours. Lessons typically followed the structure of a lecture on a specific topic followed immediately by a hands-on learning exercise or discussion. Wherever possible, individual lessons incorporated both technical and aesthetic topics.

C. Curriculum

The overall curriculum aimed to teach students how to design, program and execute a computer-controlled work of art based on computational models and theories in robotic art. The lesson topics were

- Origin and development of robotic art
- Robot communications
- Languages of motion I (periodic motion and random walks)
- Languages of motion II (kinesics, flocking, emergent behavior)
- Markov chains and "Acting for Robots"
- Workshop on designing and constructing robot bodies and mechanisms with a visiting robot artist

The remaining two lessons were in-class presentations of the midterms and final projects, where the students were asked to present their functioning prototypes and answer questions about their projects.

Students were provided with robots to experiment with (the mobile Arduino robot and Sphero mobile robot), but were also given the opportunity to develop their own design or robotic prototypes. As the course is an upper-level undergraduate course, a prerequisite for enrollment is imperative and object-oriented programming (e.g., C++ or Java).

D. Project-Based Assignments

Following the AAU PBL model, we used project-based assignments to encourage students to engage in open-ended, play-based experimentation and inquiry. We deliberately refrained from placing too many constraints on the midterm and final projects, but rather encouraged students to be guided by their own curiosity given the constraints of the relatively simple robotic platforms. Our hope was that this would result in works that were relevant to the students' experience, skill level and general artistic interest.

A necessary component for developing interactive, robotic art works is to investigate, anticipate or understand human response and reactions to the exhibited art works [2]. These topics are relevant to the study of robot design and HRI research [3]. In their projects, students were expected to apply theoretical foundations from art and performance and to explore the aesthetic potential of motions. From this we can learn how to approach concepts such as autonomy and interactivity on an experiential and aesthetic level.

There were two assignments in the course: 1) a midterm sketch/study and one-page written summary outlining the research project and 2) the completion of a group-based mini-project incorporating computer-controlled robotics. The mini-project was presented in class, and the functioning prototype was accompanied by a written report and oral presentation summarizing the project, method, approach and conclusions.

Students were allowed to form their own project groups and we made no effort to balance the groups in terms of skill sets, e.g., programming. Rather, we preferred the students form groups based on shared interest in a topic or specific robotic platform. We speculate that this approach worked well because of multiple factors: all students had basic programming skills, the incorporation of group-based class exercises, and students were from the same study program (ArT). If the student population was more diverse, it may have been necessary to structure the group formation more. However, the group-based class exercises may have the additional benefit of getting students acquainted with one another.

IV. Implementation

In this section, we present a detailed description of how certain aspects of the course were implemented. We discuss the selected teaching platforms, programming languages, and algorithms and describe the robot artist workshop and student projects.

A. Theme: Languages of Motion

The underlying theme of MMP-AA was “Languages of Motion.” Our goal was to introduce elementary algorithms of motion and how those motions, individually or combined into choreography, can create an aesthetic response in the viewer. One goal of robotic or interactive systems-based art is to evoke emotional responses through planned motions executed by otherwise inanimate objects. Other media such as computer animation and sound have their own specific languages of motion, and we believe there are general principles within languages that can be applied directly to robotics. For example, random paths are useful for exploration and for masking artificial movements and periodic functions can be composed to synthesize complex, structured patterns. Understanding the link between the languages of motion and aesthetic and cognitive responses may open up new ways of thinking about design and interactivity for robotics research [4].

B. Platforms

As the intention of the course was to focus on robot communications and languages of motion, we wanted to use ready-made mobile robots that supported open, standard protocols. We ruled out use of drones and other 3D-navigable robots early on as we felt these would present too many technical challenges as well as safety issues for an introductory course. The robot platforms chosen were the Orbotix Sphero 2.0 [5] (Figure 1) and Arduino Robot [6] (Figure 2). The Sphero is a Bluetooth-controlled, hermetically-sealed ball that can be commanded to move and turn relative to an initial reference frame. It also has a controllable colored light and can stream numerous sensor data back to the client including its position, accelerometer, gyro and IMU readings. The Arduino Robot is essentially a programmable two-wheeled cart. It also provides buttons and a dial for user input and a display screen and speaker for feedback. A major advantage of the Arduino Robot over the Sphero is that additional sensors, such as a video camera, can be easily attached to the robot. The Sphero, on the other hand, can be used on a wider variety of terrains, including water, and may be easier to use as an actuator.

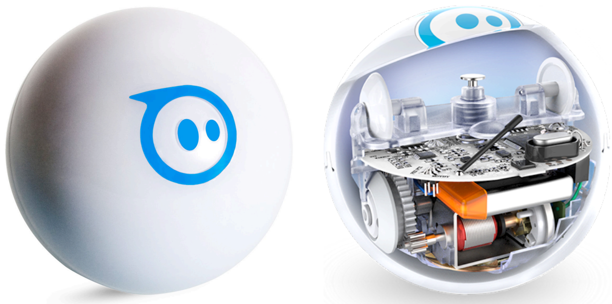

Figure 1. The Orbotix Sphero 2.0 remote-controlled mobile robot. Shown are its outer appearance (left) and inner workings (right). (Images from http://www.gosphero.com.)

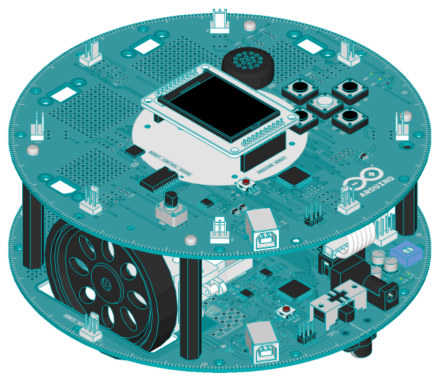

Figure 2. Illustration of the Arduino Robot: a programmable two-wheeled cart robot.

C. Programming

It was important to use the same programming language for both robot platforms to maximize the possibility of code reuse. We selected C++ as it is used by the Arduino platform and much of the robotics community. For the Sphero, we used the AlloSystem multimedia toolkit [7] and custom-made Bluetooth client and Sphero packet generating and parsing classes. We considered using the Robot Operating System (ROS) [8] but at the moment only Linux/Ubuntu is officially supported and most of our students use either Windows or OS X.

The algorithms we selected were meant to teach the students about elementary types of motions and how to combine those motions. For the elementary motions, we chose circular motion, random walks, and flocking [9] as the basis for periodic, random, and force-based motions, respectively. To combine motions two main principles were introduced: motion superposition, i.e. weighted vector summation of elementary motions, and time-based sequencing. Sequencing, in turn, was divided into two main approaches: linear and random. Linear sequencing consisted of a series of control commands separated by delta times while random sequencing was implemented via Markov chains with fixed delta times between state transitions.

In-class assignments were conducted using the Sphero and generally followed a progression from simple to more complex tasks. For example, in one lesson, the first task was to change the color of the Sphero's light. This progressed to moving the Sphero in a line and then stopping, and then onto circular and random motions. The last exercise was to simultaneously control the motion and color of the Sphero according to some pattern.

D. *Workshop with Guest Artist*

One special feature of the course was a one-day workshop taught by a visiting robot artist. The artist also gave a corresponding lecture on his own art works and design methodology. In the workshop, students constructed simple mechanisms (gears, belts, and pulleys) with flexible and cheap materials such as cardboard boxes, straws and rubber bands. In the second part of the workshop students were taught how to combine the mechanisms with Arduino controllers. Unfortunately, we did not allow enough time for the students to implement more advanced mechanisms, but each student succeeded at designing and constructing a functioning prototype. After working with rigid bodies with limited degrees of freedom and movement primitives (i.e., the Sphero and Arduino robots), the students enjoyed the chance to work with flexible materials and to design their own robots. Unfortunately, the workshop came too late in the semester for students to incorporate this knowledge into their projects. In hindsight, we believe the students would have benefited from the workshop earlier in the semester before they commenced work on their final projects. This would have introduced the possibility of building their own robots or combining the Arduino or Sphero with custom-built robots or parts.

E. *Project Descriptions*

Thirteen students worked in five groups. The projects reflected a wide range of topics, some more scientific and others more artistic in their approach and methodology.

1. *Color/Gesture Mapping*. This project aimed to map color changes to the Sphero based on human-robot interaction, using the robot's orientation and altered motion trajectories (human input) to control the color of the Sphero.

2. *Sphero Dancers*. This project abstracted choreographic structures from human dancers to generate motion trajectories for three Sphero "dancers" to create dramatic tension and evoke an emotional response in the viewers.

3. *FlirtyBot*. Using an Arduino Robot, this project built a social robot with a distinct personality, using simple interactions such as dialogue, sound, and simple movement patterns in response to input from the human user.
4. *Mind-controlled Sphero*. This project combined the Sphero platform with the Emotiv EPOC biopotential neuron headset [10] to simulate telekinesis. The custom software (authored by the students) enabled users to control the motion of the Sphero in real time using only facial expressions.
5. *"Kiwi" Interactive Narrative*. This robotic performance combined preprogrammed and tele-operated Spheros with live shadow puppetry, a musical score and narration based on the Kiwi bird, a flightless bird from New Zealand. The students developed a storyboard and applied principles of narratives and interaction to generate an original performance (Figure 3).

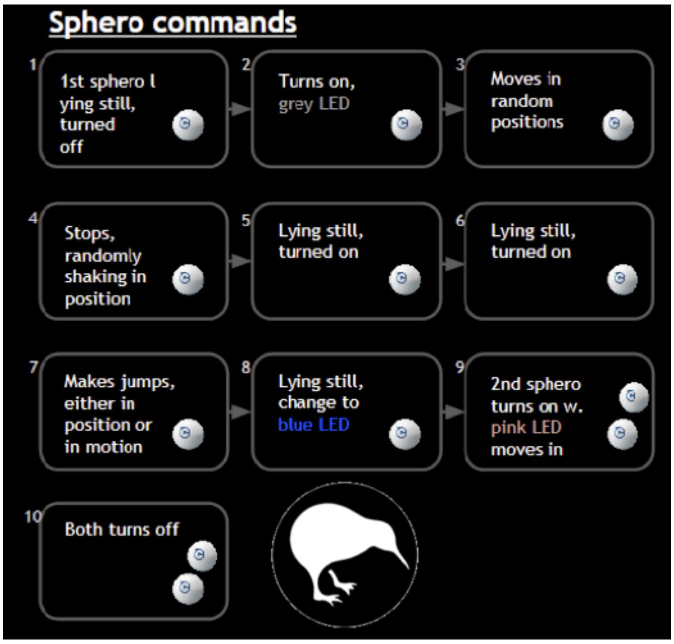

Figure 3. Storyboard of the *"Kiwi" Interactive Narrative* student project. This was a live performance that combined preprogrammed and tele-operated Spheros with live shadow puppetry, a musical score, and narration based on the flightless bird from New Zealand.

V. ANALYSIS AND REFLECTIONS

This section presents analysis and reflections of the course based on our observations and experiences in the classroom, evaluation of student projects, and student feedback.

A. *Entrance and Exit Surveys*

At the beginning of the course, students were asked to complete an entrance survey that included questions about prior programming experience and experience with robots and robotic art, as well as their expectations for the course. All thirteen students completed the intake survey. On the last day students were asked to complete an exit survey, which included questions on the course content, format, projects and challenges they faced. Eight students completed the exit survey.

Based on the entrance surveys, many students were excited about the possibility of moving from screen-based media to physical systems ("reaching beyond the screen"), and some thought that working with robots would deepen their understanding and knowledge of programming. Nearly all of the students expressed enthusiasm about the "hands-on" nature of the course, and were eager to apply their skills to more advanced, interactive art works that involved human-robot interaction.

The exit surveys show that students' initial expectations were largely met, and the students generally agreed that the course struck a good balance between theory and practice. The students appreciated the flexibility to develop their own projects as well as the opportunity to apply theory to practice in the in-class workshops and assignments. Students responded positively to the challenge of working in the physical world, including discovering the limits and challenges of working in physics-based scenarios.

Working with physical systems challenged the students in new and unexpected ways:

> "Robots sometimes react to the physical world in unpredictable ways. They might also be preprogrammed with a behavior that clashes with something we want to make them do."
>
> "A lot of the stuff we have been programming have had screen-based outputs. It was nice and quite interesting to be able to move into the physical world, and discover the limits thereof. Suddenly we had to deal with a whole new set of issues and problems along the way."
>
> "Considering physics, robots can be quite unreliable."

Many of the groups made an effort to apply theories of narrative and interaction to their projects:

> "In conclusion of the design process, many lessons have been learnt, both practical and aesthetic in creating robotic based art. The group has discovered the power of motion in narrative storytelling and how robots can seem more lifelike and organic with focus on movement rather than on form. […] This movement can be implemented like an actor in a theatrical performance."

B. *Platforms*

Out of the five student projects, four projects involved use of the Sphero and only one project used the Arduino Robot. One explanation for this preference may be because the Sphero was used as the primary teaching platform, but also possibly due to its simplistic, yet open-ended nature.

One of the problems we encountered with the Sphero is that it is difficult to get it to move accurately along a predefined trajectory over time. This stems partially from its lack of ability to report its absolute position and orientation. Our solution was to run a calibration program before each performance and align the Spheros through visual inspection. Other factors such as slipping, changes in momentum, or dropped control packets can also introduce errors in the motion trajectory and cannot be prevented through calibration. A workaround to these problems, which we did not test, may be to use some form of feedback control.

C. *Algorithms*

Overall, the students responded well to the algorithms that were taught. Most notably, several students expressed a deep interest in the theory of Markov chains and requested a more in-depth study of them.

The application of the presented algorithms to the project work varied. We found that students tended to pick one algorithm and explore its possibilities rather than combining multiple algorithms. For example, one project exclusively used circular motions, another used primarily random walks, and a third utilized a Markov chain consisting of move/turn commands. A possible explanation for this, supported by the project reports, is that the students encountered unexpected technical challenges along way and simply did not have time to progress beyond the basic algorithms. None of the projects made use of flocking which could be due either to the small number of project groups or the demand of the algorithm for more advanced control of multiple robots.

D. *Co-teaching*

Students were generally enthusiastic about the co-teaching model, and suggested that it was helpful to have both professors on-hand to answer questions and to maintain the balance between artistic and technical discourses. Personally, we feel that our distinct areas of expertise were complementary and that our mutual interest and engagement in the topic of autonomous art was a good model for the students.

VI. Future Work

Overall the students and professors were satisfied with the course, and many students expressed interest in continuing work on their projects. Given their interest and the strong artistic and technical merit of the projects, we plan to develop a second course that builds on this framework. Our objective is to develop the skills and competencies the students acquired in MMP-AA by expanding on the knowledge and practice of robotic art.

In the next course, we will introduce basic feedback control using video or other sensor data to address some of the problems with obtaining precise trajectories. We plan to introduce new algorithms that can be used to generate motion such as gradient fields, cellular automata, and chaotic or other non-linear equations. We are also interested in conducting a hands-on workshop in 3D-printed bodies and mechanisms that will build on the students' previous coursework in rapid prototyping. Overall, we hope that students will be able to refine their previous projects and at the same time experiment with designing their own robots and news forms of expressive motion.

ACKNOWLEDGMENT

We thank Mikael Vetner, Falk Heinrich, and Ståle Stenslie for their commitment to this course and to research-based teaching. We thank the members of the RELATE research group for their early advice and support of this project.

References

[1] H.J. Kim et al. “Artbotics: community-based collaborative art and technology education,” *ACM SIGGRAPH 2007 educators program*. ACM, New York, NY, USA, 2007.

[2] O. Bown et al., “The Machine as Autnomous Performer,” in *Interactive Expereince in the Digital Age*. L Candy and S. Ferguson, Eds. Springer, 2014, pp. 75-90.

[3] K. Dautenhahn, “Methodology & Themes of Human-Robot Interaction: A Growing Research Field,” *International Journal of Advanced Robotic Systems*, vol. 4, no. 1, pp. 103-108, 2007.

[4] G. Hoffman and W. Ju. “Designing Robots with Movement in Mind,” *Journal of Human-Robot Interaction*, vol 3, no. 1, pp. 89-122, 2007.

[5] Go Sphero, (2014). [Online]
Available: http://www.gosphero.com/sphero-2-0/

[6] Arduino Robot, (2014). [Online]
Available: http:// http://arduino.cc/en/Main/Robot

[7] Git Hub, Inc. (2014). AlloSphere-Research-Group/AlloSystem. [Online] Available: https://github.com/AlloSphere-Research-Group/AlloSystem

[8] M. Quigley et al. "ROS: an open-source Robot Operating System," *IRCA Workshop on Open Source Software*, vol. 3, no. 3.2, 2009.

[9] C. W. Reynolds. “Flocks, herds, and schools: A distributed behavioral model,” *Computer Graphics*, vol. 21, no. 4, pp. 25-34, 1987.

[10] Emotiv, (2014). Emotiv EPOC/EPOC+ [Online] Available: https://emotiv.com/epoc.php