2019-2020 — Autonomous Robotics — Proposed Syllabus

Background

- Students at Brookline High School from all backgrounds should be exposed to computing before they graduate. After introductory computing courses, robotics courses provide a pathway to further studies.
- Autonomy in robotics (being self-directed) implies computing. All robotics courses should stress *autonomous* robotics, because that is the future of robotics — for example, self-driving cars. Therefore, robotics courses are also computing courses.
- A robot is anything with sensors and actuators. Robotics (and computing) courses should encompass all types of robots: driving robots, assistive robots, prosthetic robots, wearable robots, sculptural robots, walking robots, grasping robots, flying robots, *etc.*.
- Robotics (and computing) courses should include a consideration of ethics and the choices posed in the design, implementation, and deployment of robots and the software controlling them.
- Robotics (and computing) courses should incorporate creativity, one of the *7 Big Ideas of Computer Science* (http://j.mp/7-big-ideas). Making computational artifacts, including robots and software, is inherently a creative endeavor.
- The study of robotics is inherently project-based and the projects can be student-directed, once students master basic skills in computing and mechanics. In this approach, task-oriented projects can include autonomous driving (wayfinding, maze solving, mapping, tailing), prosthetic devices, wearables, kinetic sculptures, legged robots, aerial robots, and Botball.
- Botball® is an autonomous robotics program with both regional and global competitions. Competitions are good for two reasons: they establish deadlines and they provide a task to accomplish every year that incorporates important aspects of autonomous robotics (odometry & route planning, vision, mechanical design, contingency and error recovery, strategy, ...)

Course syllabus

Students will be able to independently use their learning to…

use sensors	• touch sensors • distance sensors • reflectance sensors • cameras	design robots	• design rolling, walking, assistive, wearable, sculptural, flying robots
use actuators	• LEDs • drive motors • servo motors	create artifacts	• create conceptual artifacts (specifications, designs, evaluations) • create computational artifacts (code) • create physical artifacts (robots)
write code	• use block-based languages (Snap!, MIT App Inventor) • use text-based languages (Python, Robot-C, Forth) • error recovery	complete tasks	• basic mobility • line following and maze solving • object retrieval and sorting • grasping and manipulating • real-world interaction
consider ethics	• articulate technological impact • make ethical design choices • cite sources for remixed work	apply strategy	• plan task completion • use feedback to adjust strategy • respond to real-world failures
communicate	• reflect on work of self and others • review work of self and others	explore advances	• use artificial intelligence • use multi-robot swarms

Optional: compete in Botball	https://www.kipr.org/botball/what-is-botball/aligned-standards-national-impact — critical thinking, analytical skills, adaptive learning / flexibility, decision making, computational thinking, creativity / innovation, collaboration, problem solving, communication.

Student practices will include...

From PBL:
- solve challenging & authentic problems
- sustain inquiry in their problem-solving
- direct their own problem-solving process
- iteratively critique & revise their design & implementation (imagine - design - build - test)
- create artifacts for their authentic audience
- reflect on their designs & artifacts and on the process

From DLCS:
- creating
- connecting
- abstracting
- analyzing
- communicating
- collaborating
- researching

Student subject matter knowledge will include...

Computing and Society
- Understand safety and security concepts, security and recovery strategies, and how to deal with cyberbullying and peer pressure in a social computing setting. (9-12.CAS.a – Safety and Security)
- Understand, analyze impact and intent of, and apply technology laws, license agreements and permissions. (9-12.CAS.b – Ethics and Laws)
- Recognize, analyze, and evaluate the impact of technology, assistive technology, technology proficiencies, and cybercrime in people's lives, commerce, and society. (9-12.CAS.c – Interpersonal and Societal Impact)

Digital Tools and Collaboration
- Selection and use of digital tools or resources and computing devices to create an artifact, solve a problem, communicate, publish online or accomplish a real-world task. (9-12.DTC.a – Digital Tools, 9-12.DTC.b – Collaboration and Communication, 9-12.CS.a – Computing Devices)
- Use of advance research skills including advanced searches, digital source evaluation, synthesis of information and appropriate digital citation. (9-12.DTC.c – Research)
- Understand how computing device components work. Use of troubleshooting strategies to solve routine hardware and software problems. (9-12.CS.a – Computing Devices, 9-12.CS.b – Human and Computer Partnerships)
- Understand how networks communicate, their vulnerabilities and issues that may impact their functionality. Evaluate the benefits of using a service with respect to function and quality. (9-12.CS.c – Networks, 9-12.CS.d – Services)

Computational Thinking
- Creation of new representations, through generalization and decomposition. Write and debug algorithms in a structured language. (9-12.CT.a – Abstraction, 9-12.CT.b – Algorithms)
- Understand how different data representation affects storage and quality. Create, modify, and manipulate data structures, data sets, and data visualizations. (9-12.CT.c – Data)
- Decompose tasks/problems into sub-problems to plan solutions. (9-12.CT.d – Programming and Development)

- Creation of programs using an iterative design process to create an artifact or solve a problem. (9-12.CT.d – Programming and Development)
- Creation of models and simulations to formulate, test, analyze, and refine a hypothesis. (9-12.CT.e – Modeling and Simulation)

This subject matter knowledge is from the *2016 Massachusetts Digital Literacy and Computer Science (DLCS) Curriculum Framework*. Additional subject matter knowledge will include strands and topics from the *2016 Massachusetts Science and Technology / Engineering Curriculum Framework* (including HS-ETS-1 – engineering design, HS-ETS-2 – materials, tools, and manufacturing, and HS-ETS-3 – technological systems).