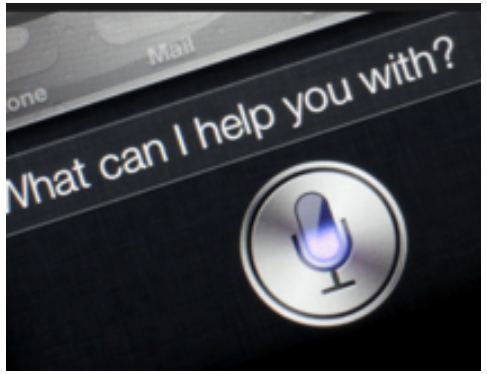
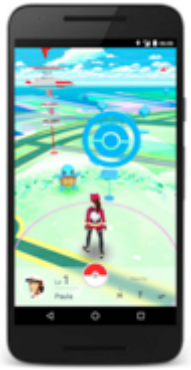
Brain Maker: Creating Your Future

Robot + AR: Bringing Computational Thinking to Mainstream Education

The Algorithmic Economy

- Computer has permeated every aspects of our life
- AI and AR have moved out of research labs into mainstream applications





Teaching Computer Science To K-12

Has Become A Necessity

It's now commonly accepted that, just like Math and Language Arts, Computer Science should be taught during K-12 (Adding a "C" in front of "STEM")

"In fifteen years we'll be teaching programming just like reading and writing . . . and wondering why we didn't do it sooner."

— Mark Zuckerberg

"Don't just play on your phone, program it."

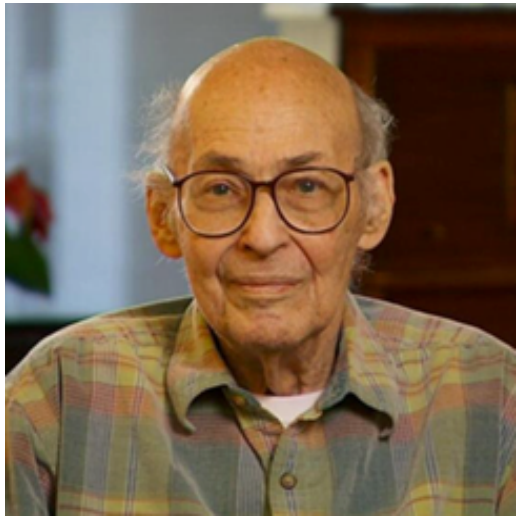
— President Barack Obama

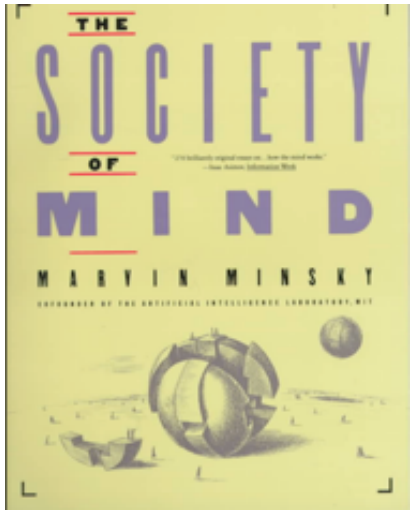
Computer Science Is More Than Coding

It is a new way of thinking

"Computer Science is not only about computers themselves; more generally, it provides us with a whole new world of ways to understand complex processes"

- Marvin Minsky





How To Teach “Computational Thinking”

The Challenges Of Computer Science Education

- Concepts too abstract for students to understand, and challenging for teachers to explain
- Curriculum too boring to attract students attention and interest
- While syntax of computer languages can be taught easily, teaching problem solving skill is much harder

Fast Growing Market

Opportunities and Challenges

Despite the challenges, Computer Science education has become the fastest growing segment of the educational market. But still a Blue Ocean:

- Challenges： Need deep innovation
- Opportunities： Can establish brand and platform

Brain Maker

A Constructionism Approach To Computer Science Education

By integrating with Robotic, AR, and gaming technologies, Brain Maker allows students to learn computational thinking and AI concepts intuitively by creating a brain for a robot. Key differentiation：

- Make Abstract Concept More Intuitive
- Make Boring Curriculum More Fun
- Lower Entry Barrier for Teacher and Students

Curriculum focuses on training students problem solving skills and creativity.

Seymour Papert & Constructionism

- Seymour Papert, with his 40+ years of research, has laid the foundation of how best to teach “Computational Thinking”
- Constructionism holds that learning can happen most effectively when people are active in making tangible objects in the real world

The “turtle” was a physical robot that could be programmed by Logo commands and draw geometrical shapes.

Integration With Robotics

Intuition and Fun

By Integrating with robotics technology, Brain Maker curriculum maps abstract concepts into the physical world to enable students to learn computational thinking in a fun and intuitive way

Integration with AR and Gaming Technologies

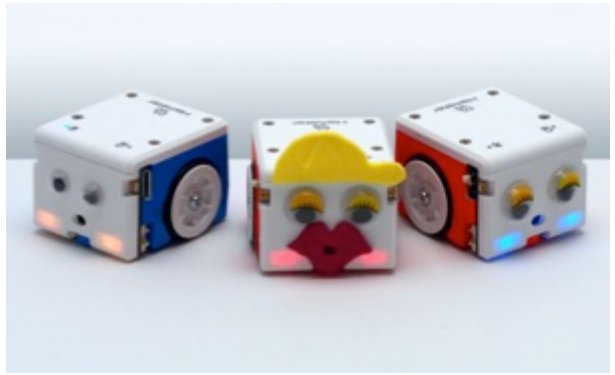
Motivation and Imagination

Taking advantages of AR technologies, Brain Maker cleverly integrates physical robots with the virtual world to allow students to explore interactivity in both the virtual and physical worlds. This not only makes the curriculum more fun, but also helps with the imagination of the students

Vertically Integrated Solution

hardware + software + curriculum

Hardware



AFFORDABLE: Small, low cost, easy to use

Software

FREE and **OPEN** to form standard and to drive content development

Curriculum

EXTENSIVE: from university level to 3rd grade level;

- 1st party contents for quality and branding;
- 3rd-party contents for sustainable profitability

Cross-platform Supports

Supports All Common Hardware and Popular Languages

Phone & Tablets With Scratch

Scratch on Mac or PC

For more advanced class using Scratch

Python on Mac or PC

For more advanced courses

Components Of A Digital Brain

Machine Learning:
Reinforcement Learning
Neural network

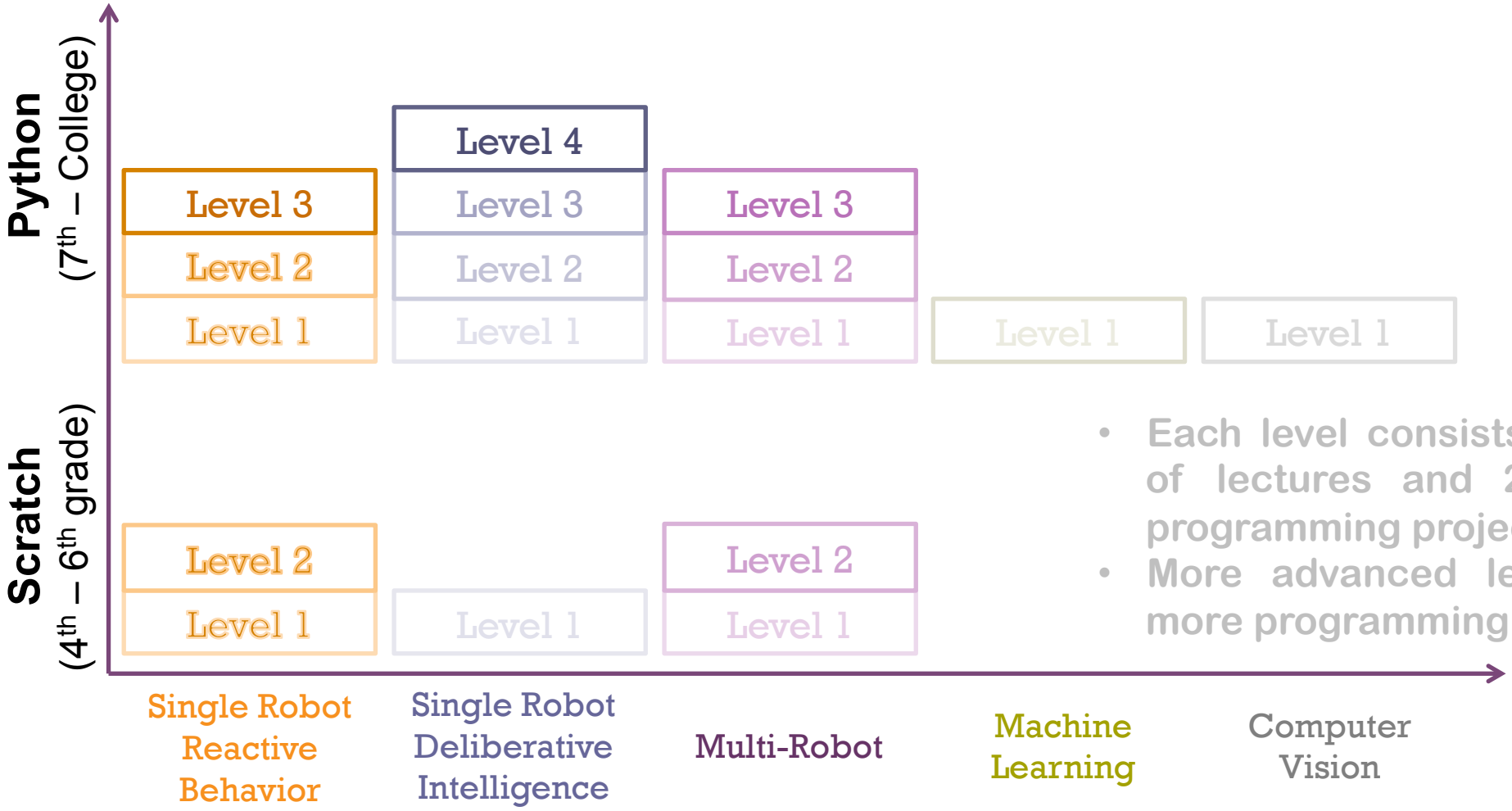
Multiple Agents:
Motion Coordination
(Coupled vs. Decoupled)
Swarm Robots

Deliberative Intelligence:
Modeling
Planning
Localization

Reactive Intelligence:
Closed-loop Control
Event-driven Programming
Finite State Machine

Brain Maker Comprehensive Curriculum

Easy to start: Hard to master



- Each level consists of 10 hours of lectures and 20+ hours of programming projects.
- More advanced levels involves more programming projects

Brain Maker Curriculum: Beginner

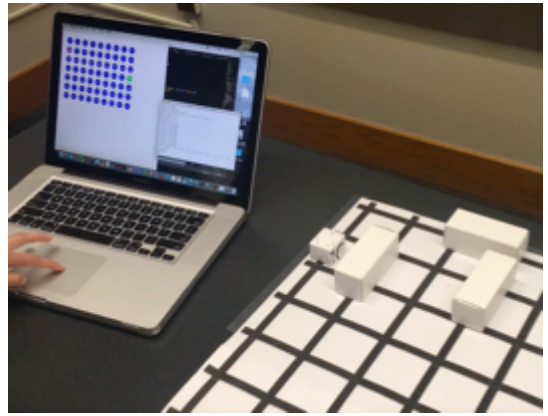
Intuitive, Fun, Easy To Star

- For Elementary School Students with no prior programming experience
- 3 Semesters（Each Semester with 20-25 hours of teaching materials + Homework）
- Curriculum includes (but not limited to)：
  - Basic logic（If-then-else, While, For loop）
  - Variables and lists
  - Robot actuators and sensors
  - Closed-loop control
  - Reactive intelligence
- Curriculum focuses on teaching problem solving skills and computational thinking
- Curriculum emphasizes on being intuitive and fun
- Using visual programming environment (Scratch) , students can get robot moving within 5 minutes

Brain Maker Curriculum: Intermediate

Computational Thinking and Algorithm

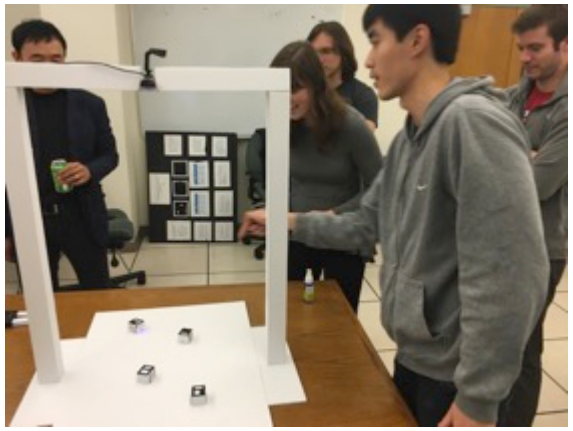
- For Middle School Students with basic prior programming background
- 4 Semesters（Each Semester with 20-25 hours of teaching materials + Homework）
- Curriculum includes (but not limited to)：
  - Data Structure and various data types
  - Functions
  - Algorithm (e.g. Maze solver)
  - Introduction to Processes and Threads
  - Concept of States and State Machine
  - Introduction to OOP
  - Introduction to GUI
- Curriculum focuses on algorithm design and implementation as well as unique challenges of programming a physical device
- Curriculum emphasizes on being intuitive and fun
- Using Python (with template provided)



Brain Maker Curriculum: Advanced

Artificial Intelligence Concepts

- For High School and College Students with substantial prior programming background
- 6 Semesters（Each Semester with 20-25 hour of teaching materials + Homework）
- Curriculum includes (but not limited to)：
  - Graph and Search
  - Robot Motion Planning
  - SLAM Concept and Particle Filter Algorithm
  - Multi-robot Coordination
  - Introduction to Computer Vision
- Curriculum focuses on various basic concepts of Artificial Intelligence
- Curriculum emphasizes on combining theory with practice (implementation)
- Using Python (with additional libraries provided)



Brain Maker Encourages Discovery Learning

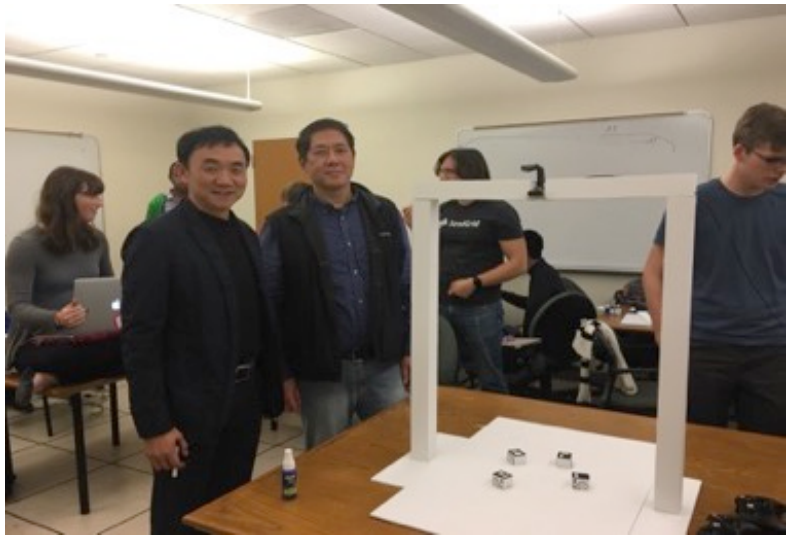
A Better Way To Learn Computer Science

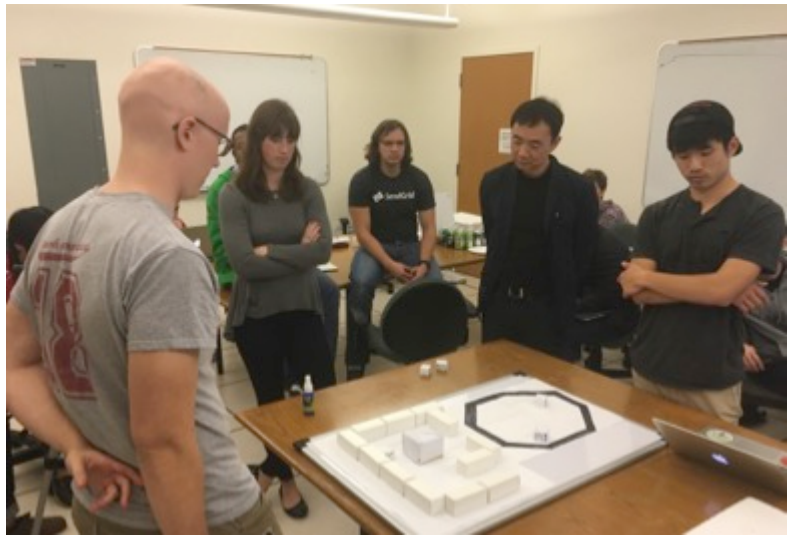
> *Many children are held back in their learning because they have a model of learning in which you have either 'got it' or 'got it wrong.' But when you program a computer you almost never get it right the first time. Learning to be a master programmer is learning to become highly skilled at isolating and correcting bugs ... The question to ask about the program is not whether it is right or wrong, but if it is fixable. If this way of looking at intellectual products were generalized to how the larger culture thinks about knowledge and its acquisition we might all be less intimidated by our fears of 'being wrong.'*

- Visual Feedbacks
- Quick Iteration
- Team Learning

Case Study： Advanced Curriculum

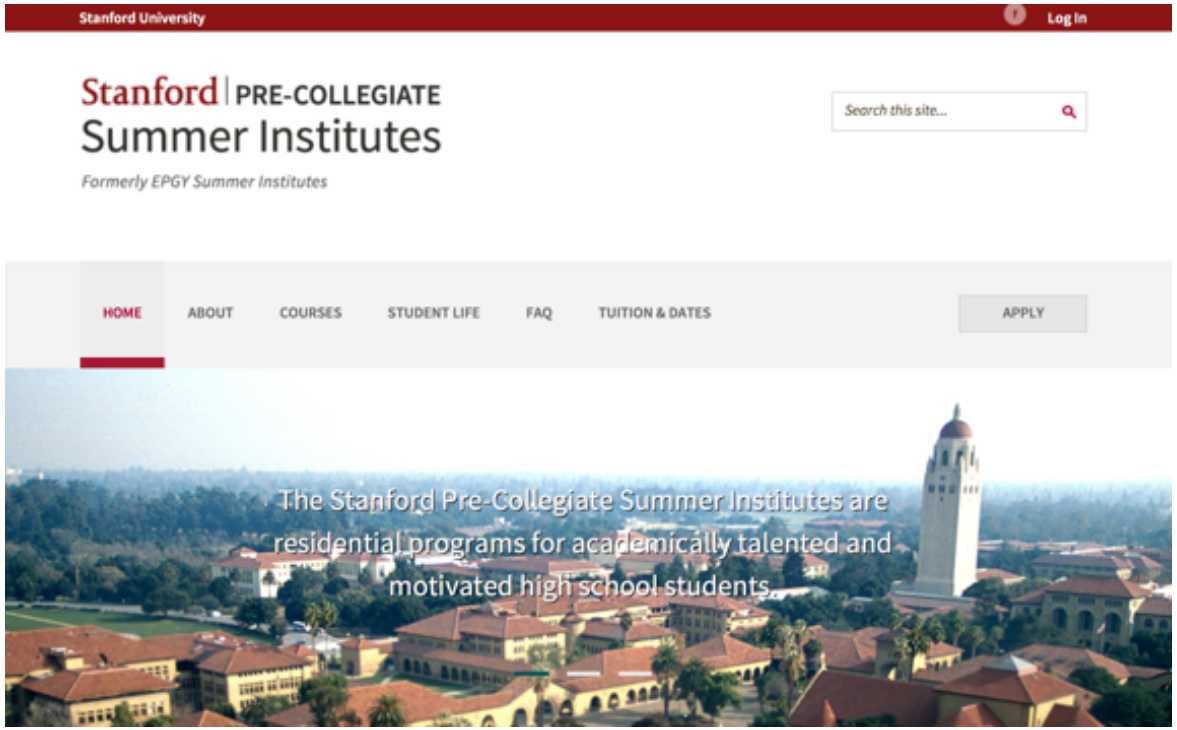
Stanford University/Computer Science Department





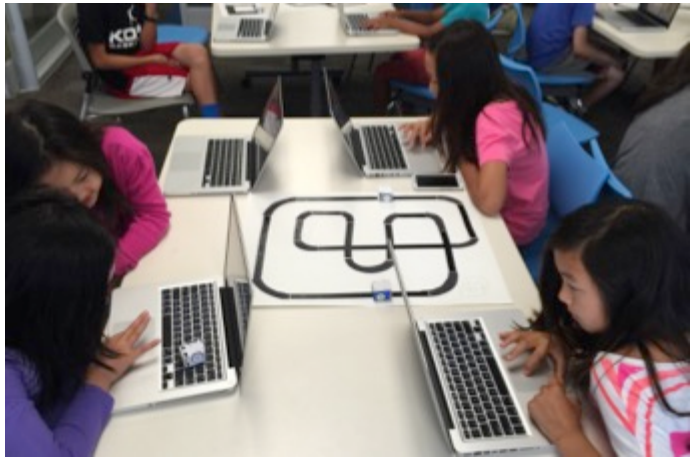
Case Study: Intermediate Curriculum

Stanford Pre-collegiate Summer Institutes



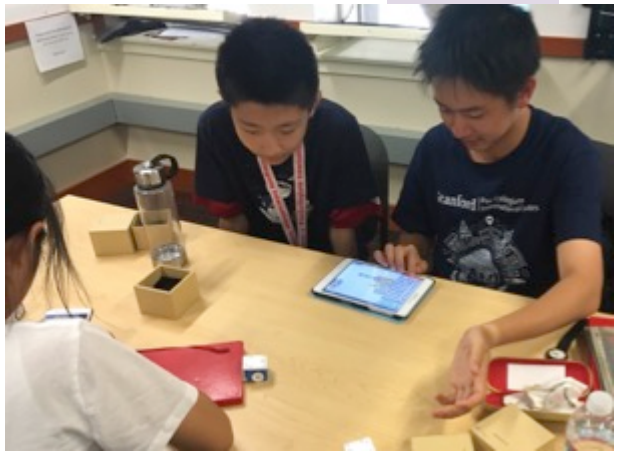
Case Study: Introductory Curriculum

Silicon Valley Summer Camp



Case Study: International Seminar

Stanford | Pre-Collegiate International Institutes

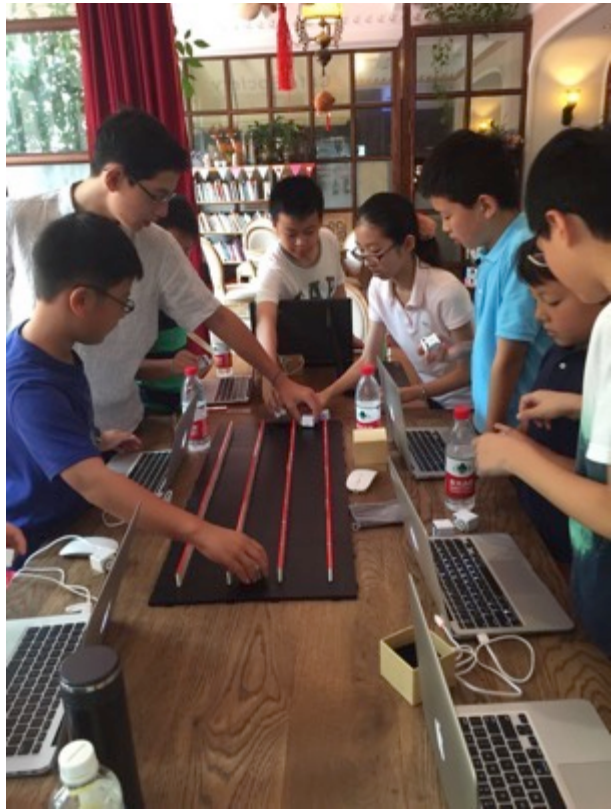


Case Study: Korea

Robot was adopted country-wide

Case Study: China

Shanghai Summer Camp



Case Study: China

Shanghai PingHe Bilingual School

Student Feedbacks

Stanford Pre-collegiate Summer Institute

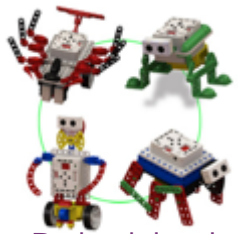
S-593474	Being introduced to event-driven programming, and being forced to think about the consequences of the code on a physical robot. Coming from a Java background, I was able to experience a new programming paradigm and broaden my viewpoint.
S-640732	I learned a lot about computational thinking in general and especially in computer science. This course taught me a lot about programming and Python and what it required to become a successful programer. This course challenged me a lot which I really enjoyed because it made me a smarter and more knowledgeable programer.
S-672867	I have never learned python or worked with robots before. I enjoyed learning both of these skills at SPCS.
S-732167	As a student who hasn't had any opportunity to learn about computer science, this course wasn't only a great introduction but also a briliant way to motivate myself in future. With the robot provided to us, I'm already prepared to play with my code at my house when I return to my area. As professor David Zhu often told us, this truly was a great experience because I experienced how to actually apply the code that I built unlike in usual CS classes where people only code.
S-833018	The class did not hold anything back and throughly challenged me. It did not provide solutions or cheat code and pushed me to discover things and gain knowledge myself. It was very riveting in that it directly linked programming with something physically dynamic and also provided bigger scale concepts and relations to the real world.
S-879985	1/2 theory, 1/2 practice was definitely useful for learning and then actually practicing. It was meaningful to be able to see directly how the code impacted the real world through the robot, as well as experiencing the difficulties of trying to accurately control an imperfect system even with perfect code.
S-969855	We learned an incredible amount in a very short period of time. It was also the first time that I learned about the theory of robotics which was fascinating. I think learning an overview of general topics and implementing them in the TA sessions was extremely useful.
S-976763	We learned a lot through the course and I liked the schedule, even though I would like a bit of teaching before the lab starts and a break between the 3 hour lab.

Stanford | Pre-Collegiate Summer Institutes

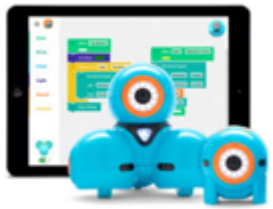
Competitive Analysis

Hardware vs. Software (and Curriculum)

Hardware (Complexity and Function)



Roboblock



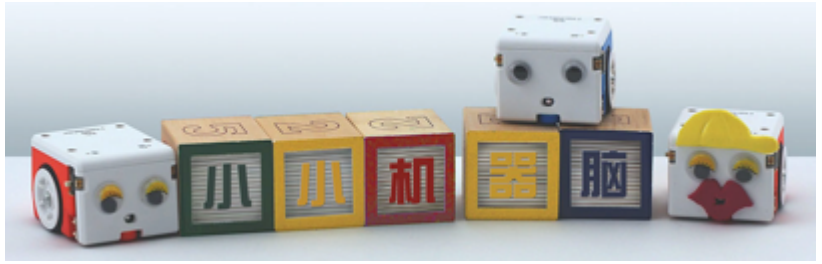
Dash&Dot

Lego-EV

Nao

By cleverly integrated with robotic, AR, and gaming technologies, Brain Maker is the only integrated platform that offers a broad range of curriculum with very simple robot hardware

Ozobot



Scratch Jr.

Tynker

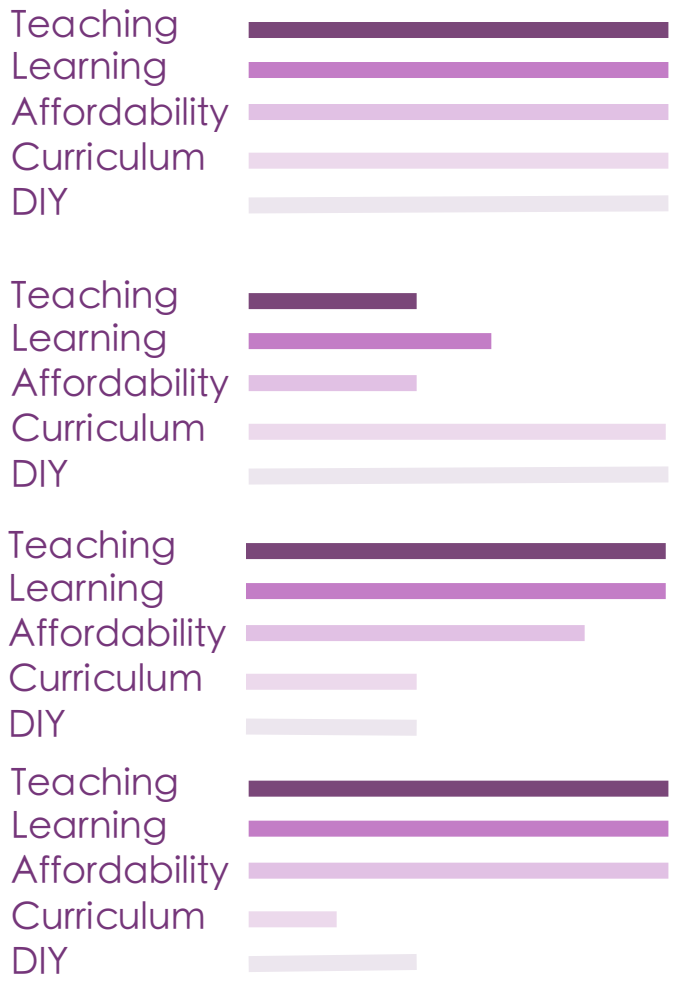
Swift Playground

UDACITY

Software (Depth Of Curriculum)

Competitive Analysis

Ease of Deployment vs. Depth of Curriculum



- Ease Of Deployment：
  - **Easy to Teach**： Standard programming environment (languages), complete curriculum
  - **Easy to Learn**： Fun and Intuitive。
  - **Easy to Afford**： Low cost (including maintenance cost)
- Depth Of Curriculum：
  - **Comprehensive Curriculum**: From Elementary School (starting 3rd grade) to College Level
  - **Expendable Framework**: Our curriculum allow easy customization and expansion

Business Model

Hardware + Software, Online + Offline

- Hardware + Software: avoid the issues of software/content piracy and/or cheap hardware clones
- Offline + Online: Offline for brand creation and market education; Online for low cost scalable deployment
- Our business model:
  - Off-line: Licensing Brain Maker (a vertically integrate solution with online support) to Schools
  - Online: Offer online curriculum directly to students

Founding Team

- Founded by Dr. Zhu (PhD, Stanford University) and Dr. Kim (PhD, KAIST)
- Both Successful Serial Entrepreneurs
- Experienced with integrated hardware / software product development
- Deep domain knowledge in robotics and AI
- Deep root in education

Co-Founder: Dr. David Zhu

Stanford Ph.D. Robotics and AI

- Computer Science Ph.D. from Stanford University, with a focus on Artificial Intelligence and Robotics
- 20+ years of entrepreneurial experience in robotics and gaming, including:
  - Nomadic Technologies : Pioneered the use of robots in teaching and research
  - Motion Factory: Pioneered the concept of “Digital Actors” by applying robotics and AI to gaming

Artificial intelligence put to practical use

Computer science class applies robots to everyday tasks

By Carrie Chang

No, they aren't walking garbage cans.

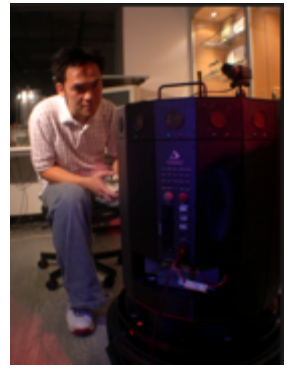
But these 3-foot cylindrical creatures might someday be the ones to take out your trash. Or bag your groceries to the front door. Or maybe even fry up a tasty omelette and deliver it to your bed in the morning.

Meet Karel the Robot in 3-D. Unlike its on-screen counterpart, however, these machines can do a lot more than pick up beepers and run through mazes. These "agents," the subjects of a computer science course being offered this quarter, come equipped with sensors that allow them to assess their environments and alter their operations accordingly.

"Artificial intelligence for the past few decades has been concerned primarily with very brainy

Please see ROBOTS, page 2

Robotics entrepreneur David Zhu, left, and Computer Science Prof. Mike Genesereth pose with Nomad and Vagabond, two robots to be used in an innovative class offered this fall.



斯坦福就学期间创办的机器人公司Nomadic Technologies

Co-founder: Dr. Kyoung Jin Kim

Recognized Educational Robotic Expert In Korea

- Founder of Robomation, pioneered the concept of “thin-client” robots
- 15+ years of experience in educational robots
- Designer of famous Korean educational robot “Albert”
- Recognized expert in educational robotics in Korea

Famous educational robot Albert

Advisors

Professor Latombe
Stanford University

- Professor (emeritus) Stanford University
- Director Stanford Robotics Lab
- Chairman Stanford Computer Science Department
- Pioneer in AI and Robotics
- Author of the “Robot Motion Planning” book

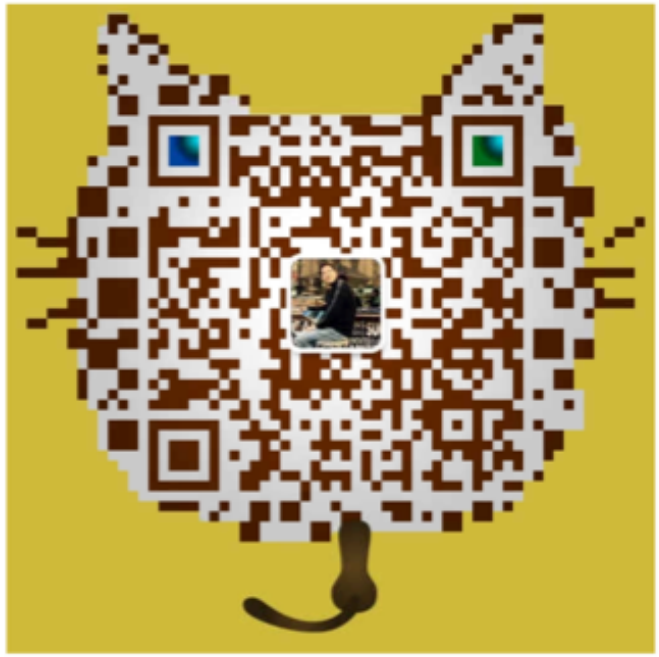
Dr. Kyong-Sok Chang
CEO Wonik Robotics

- Stanford PhD
- Founder and CEO of SimLab, a company providing software platform for advanced research and teaching in robotics
- 10+ years educational robot development experience

Kre8 Technology Advantages

- Product: Deep innovation, vertically integrated solution, curriculum focus
- Branding: Stanford University and Stanford Pre-collegiate Summer Institutes
- Business Model: Integrated Hardware + Software, Offline + Online
- Founding Team: Experience, Deep Domain Knowledge, Rooted in Education

Join Us To Create The “Future Of Learning”



Wechat: robot-toys
eMail: info@kre8tech.com