Pair Programming in K-8

A powerful tool for Computer Science engagement & learning

What We Will Do Today

- Explore why Pair Programming can be an effective tool to motivate student interest in computer science
- Understand best practices for Pair Programming based on research
- Experience it ourselves! Break into pairs to program in Scratch:

    1st: as pair programmers creating a conversation between 2 sprites

    2nd: as “buddy programmers” animating a scene
- Discuss at the end of each project what we learned about the process.

Goals

- Experience pair and buddy programming first-hand & understand best-practices
- Have fun learning to code with the support of others!

What is Pair Programming?

2 students work on 1 device, alternating roles:

- Driver - controls mouse /keyboard & does actual coding
- Navigator - "...aids driver in thinking through issues, searching for errors, discussing alternatives..."[1]

2

[1]Salge, C. A. d. L. and Berente, N. (2016); [2]Code.org (2014)

Why Pair Programming?

- Increases engagement, confidence & perseverance[1] in computer science courses & majors
- Fosters collaboration & metacognition[2] as students describe their thinking to a partner
- Improves the quality of code students produce[1]
- Reflects the real-world collaboration required in computer science occupations
- A necessity in classrooms without 1-1 devices
- Provides an additional source of knowledge and support in the classroom

[1]McDowell, C., Werner, L., Bullock, H., and Fernald, J. (2002); [2]Werner and Denning, 2009

Pair Programming Supports Diversity in CS

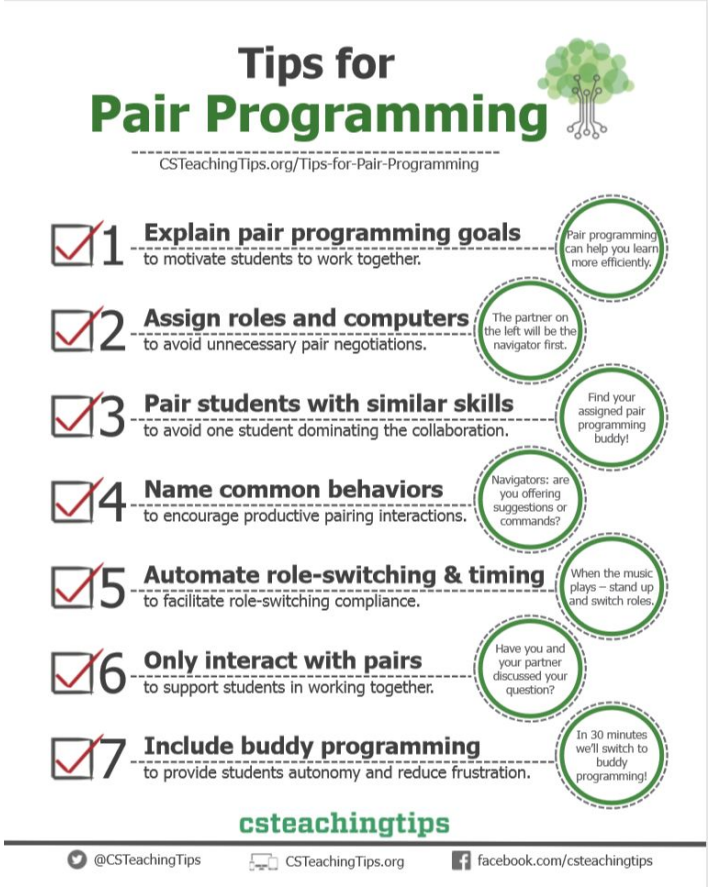
Research[1] at UC Santa Cruz on pair-programming in introductory Computer Science classes showed students who pair-programmed were more likely to complete the class and go on to major in CS than students who did not pair-program. The effect on female students was greater than on males.

Harvey Mudd College uses pair and buddy programming as well as other supports for students who enter college with little programming experience. In 2018, 56% of Harvey Mudd students earning a Bachelor's degree in Computer Science were female[2], compared to just 19% of all U.S. Bachelor in Computer Science degrees in 2017[3].

[1] McDowell, C., Werner, L.L, Bullock, H., and Fernald, J. (2002); Werner, L.L., Hanks, B., & McDowell, C. (2004)
[2]"Harvey Mudd Graduates Highest-ever Percentage of Women Physics and Computer Science Majors" (2018); Weisul, K. (2017)
[3]Women and Information Technology, By the Numbers. (2018);

Best Practices

Tips from [Colleen Lewis](), Associate Professor of Computer Science at Harvey Mudd College



Lewis, C. (n.d.)

CS Teaching Tips: Tips for Pair Programmers

Lewis, C. (2016) CS Teaching Tips: Tips for pair programming

Strategies for Younger Students with Diverse Needs[1]

- Arrange your classroom to facilitate collaboration
- Communicate the importance of *each student's* input & ideas
- Model & encourage collaboration & problem solving skills using the Collaborative Discussion Framework[2] for seeking & giving help
- Convey that working well together is as important as the code they produce. Reflect as a group on the collaboration process, not just CS content.
- Be flexible; some students may not be able to tolerate strict pair programming roles/timing
- Consider cross-grade collaborations, similar to "reading buddies"

[1]Peer collaboration during K-12 Computer Science instruction (n.d.)
[2]Park, M., & Lash, T. (2014)

K-12 CS Framework: Collaboration is a Core Computing Practice[1]

Washington State K-8 CS Standards Addressed with Pair Programming[2]

Identifier	Level 1B: 3–5
1B-AP-16	Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development. (P. 2.2)
1B-AP-17	Describe choices made during program development using code comments, presentations, and demonstrations. (P. 7.2)
1B-IC-20	Seek diverse perspectives for the purpose of improving computational artifacts. (P. 1.1)

Identifier	Level 2: 6–8
2-AP-15	Seek and incorporate feedback from team members and users to refine a solution that meets user needs. (P. 2.3, P. 1.1)
2-AP-18	Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts. (P. 2.2)

[1]Peer collaboration during K-12 Computer Science instruction (ND). Creative Technology Research Lab, University of Illinois.
[2]Washington State Computer Science K–12 Learning Standards (Attribution: CC BY-NC-SA 4.0)

Next Steps

1. Respond to the Scratch Experience Survey
2. Go to assigned group area (Beginner, Competent, Expert) & find a partner
3. Open the Pair Programming Workshop Links document & follow instructions for opening & naming a Scratch project.
4. Work with your partner to create a Scratch project with two characters having a conversation on a topic of your choice. *Ideas: why you became teachers, goals for your teaching next year, what you love most and find the most difficult in teaching, exciting things you've learned at NCCE*

Don’t Forget Pair Programming Best Practices!

- Use only 1 device to write code per pair
- Person on left starts as the Navigator
- Switch roles every 5 min (at timer)
- Driver writes code
- Navigator oversees logic & bug detection
- Share your thinking aloud with your partner
- Use comments in your code to document your thinking!

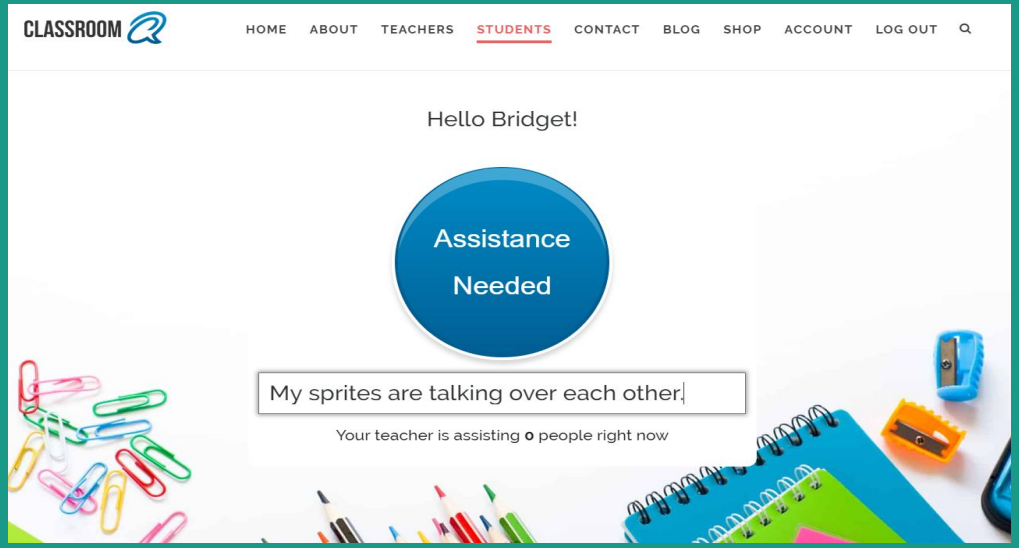
If You Need Help

- Try using the Collaborative Discussion Framework[1] to prompt for solutions
- Go to the "Scratch Help" section in the Pair Programming Workshop Links doc

Still need help? No problem! See the next slide...

[1]Park and Lash, 2014

If as a team you are stuck, use ClassroomQ to add your names to a queue for help.



Expert participants will be asked to help teams when workshop presenters are helping others (remember: coding is collaborative😊).

Pair Programming Discussion & Project Sharing

What did you like/not like about pair-programming?

Did anything surprise you?

Would you do anything differently?

If you'd like to share your project, email a link to me at: greenbergb@digitaledleadership.org

2nd Project: Buddy Programming

1. Get back together with your partner, but this time you will code separately on your own devices.
2. Follow instructions for opening & naming a Scratch project in the Pair Programming Workshop Links document.
3. Create a project in Scratch where you describe a favorite subject, activity, or hobby and include an animation sequence.
4. Every 5 minutes (at the timer) turn to your partner and discuss any problems you are having with your code and try to help each other find solutions. No problems to discuss? Share something you've figured out instead!

Buddy Programming Discussion

What did you like/not like about buddy-programming?

Did anything surprise you?

Would you do anything differently?

Final Thoughts?

Links

Workshop Links

Pair Programming Resources

Scratch Resources

References - 1

Harvey Mudd graduates highest-ever percentage of women Physics and Computer Science majors (2018). Retrieved from: https://www.hmc.edu/about-hmc/2018/05/15/harvey-mudd-graduates-highest-ever-percentage-of-women-physics-and-computer-science-majors/

McDowell, C., Werner, L., Bullock, H., and Fernald, J. (2002). The effects of pair-programming on performance in an introductory programming course. Presented at The 33rd ACM Technical Symposium on Computer Science Education, February 7-March 3, 2002. Retrieved from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.89.6834&rep=rep1&type=pdf

Pair programming (2014). Code.org. Retrieved from: https://www.youtube.com/watch?v=vgkahOzFH2Q

Park, M., & Lash, T. (2014) Collaborative discussion framework. Retrieved from: https://docs.google.com/document/d/1rx7d9iUr_sLra2ylcH4Z-h6NqWjAeoAmesDqJbxCrJw/edit

References - 2

Peer collaboration during K-12 Computer Science instruction (ND). *Creative Technology Research Lab*, University of Illinois. Retrieved from: https://ctrl.education.illinois.edu/TACTICal/Collaboration

Salge, C. A. d. L. and Berente, N. (2016). "Pair programming vs. solo programming: What do we know after 15 years of research?" *49th Hawaii International Conference on System Sciences* (HICSS), Koloa, HI, 2016, pp. 5398-5406.
doi: 10.1109/HICSS.2016.667

Washington State Computer Science K–12 Learning Standards (2018). Washington State Office of Superintendent of Public Instruction. Retrieved from:
https://www.k12.wa.us/student-success/resources-subject-area/computer-science/computer-science-k-12-learning-standards

Weisul, K. (2017). Half of this college's STEM graduates are women. Here's what it did differently. *Inc.com* Retrieved from:
https://www.inc.com/kimberly-weisul/how-harvey-mudd-college-achieved-gender-parity-computer-science-engineering-physics.html

References - 3

Werner, L.L., Hanks, B., & McDowell, C. (2004). Pair-programming helps female computer science students. *Journal of Educational Resources*. Retrieved from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.208.8243&rep=rep1&type=pdf

Werner, L.L. , Denning, J. (2009). Pair programming in middle school: What does it look like? *Journal of Research on Technology in Education*, 42(1), 29–49. Retrieved from: https://files.eric.ed.gov/fulltext/EJ856931.pdf

Women and information technology: By the numbers (2019). National Center for Women in Information Technology. Retrieved from: https://www.ncwit.org/sites/default/files/resources/btn_05092019_web.pdf