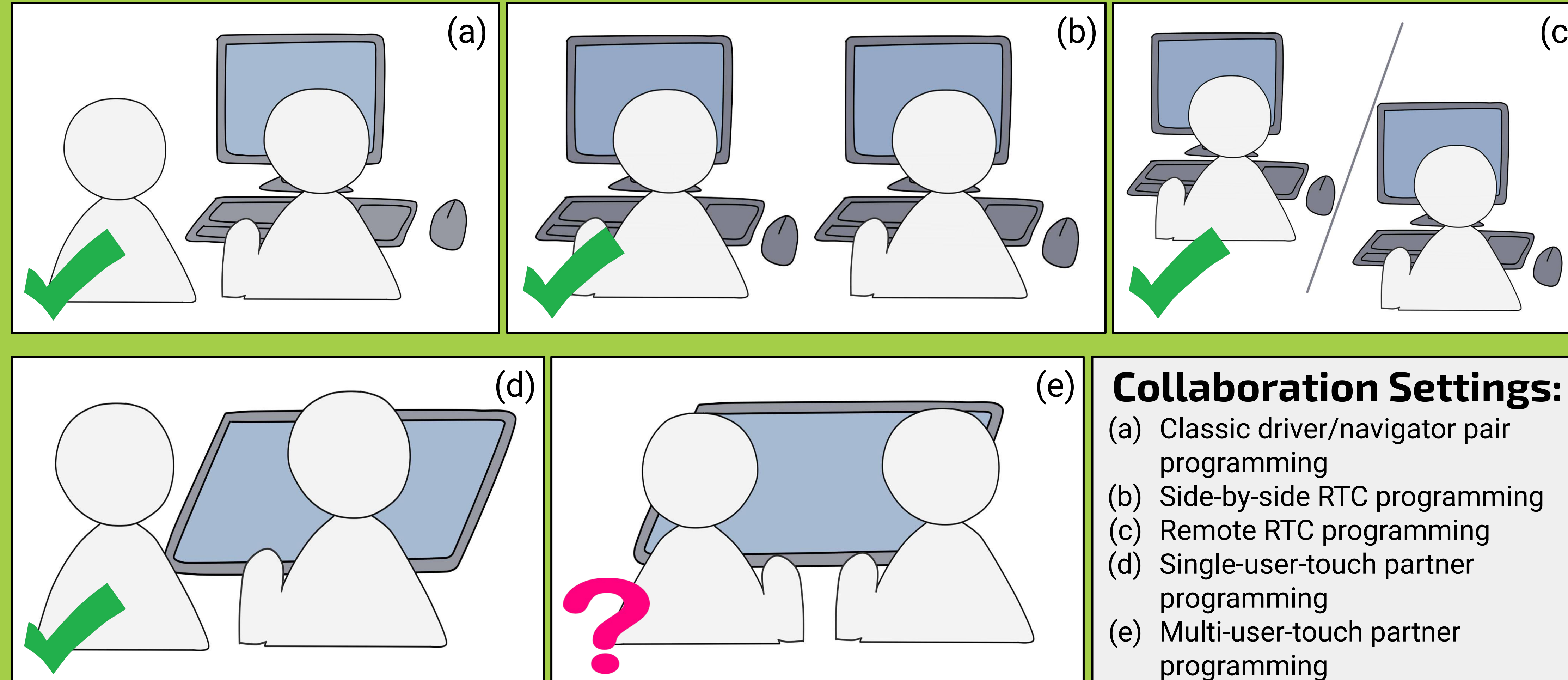


Introduction

App Inventor is a web-based blocks programming environment that allows people to build mobile apps.

Since collaboration in a blocks programming environment is not well understood, we designed experiments to evaluate different collaboration settings to determine the benefits and drawbacks of each. Our findings will inform the development of a multi-user-touch system.

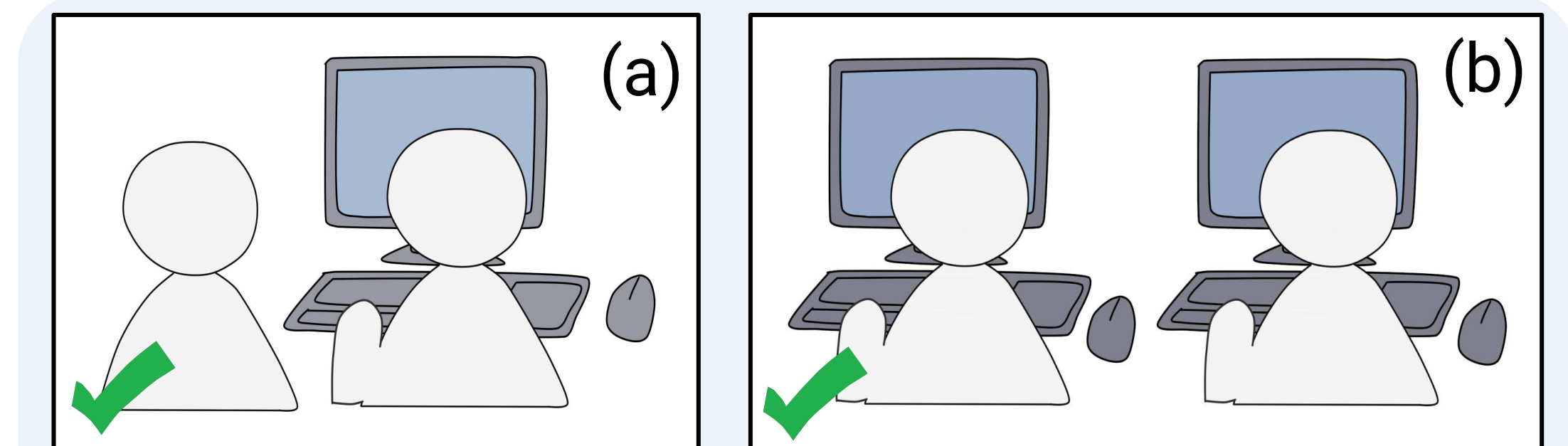


Collaboration Settings:

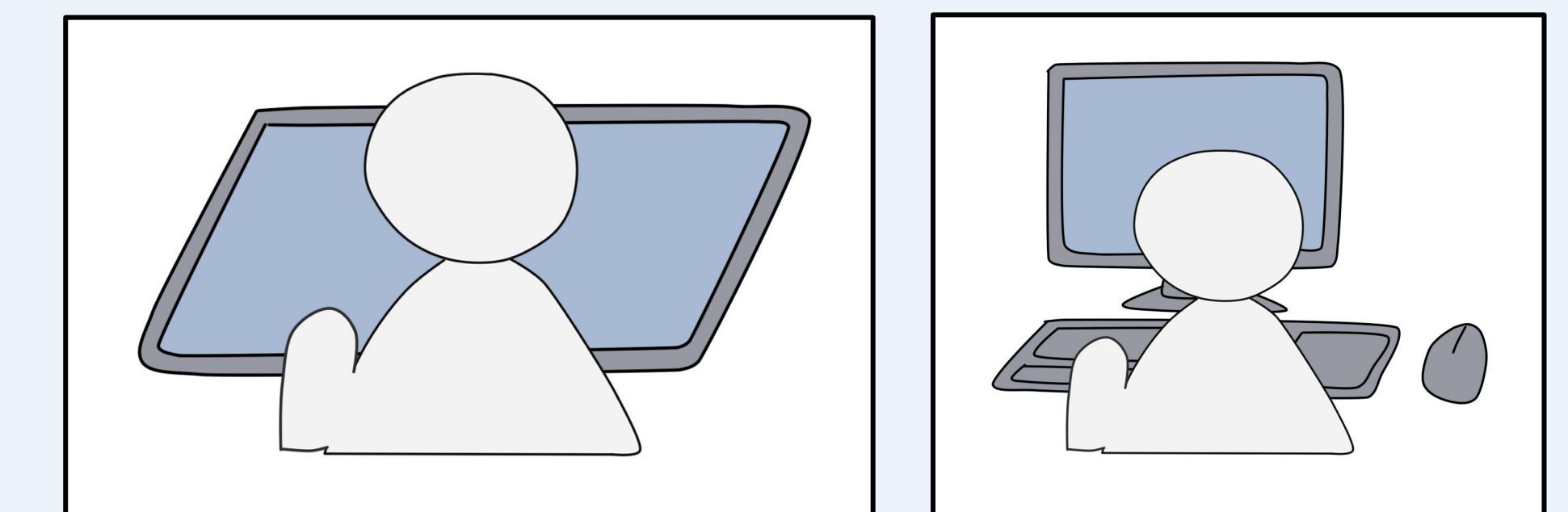
- (a) Classic driver/navigator pair programming
- (b) Side-by-side RTC programming
- (c) Remote RTC programming
- (d) Single-user-touch partner programming
- (e) Multi-user-touch partner programming

Experiment Timeline

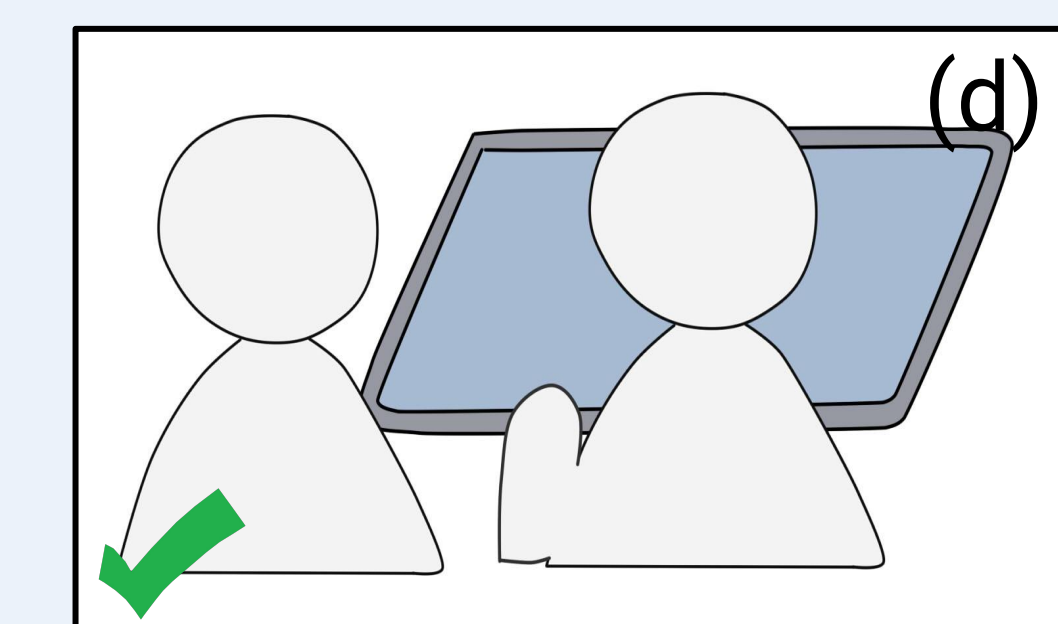
1. We will look at two collaboration settings in App Inventor.



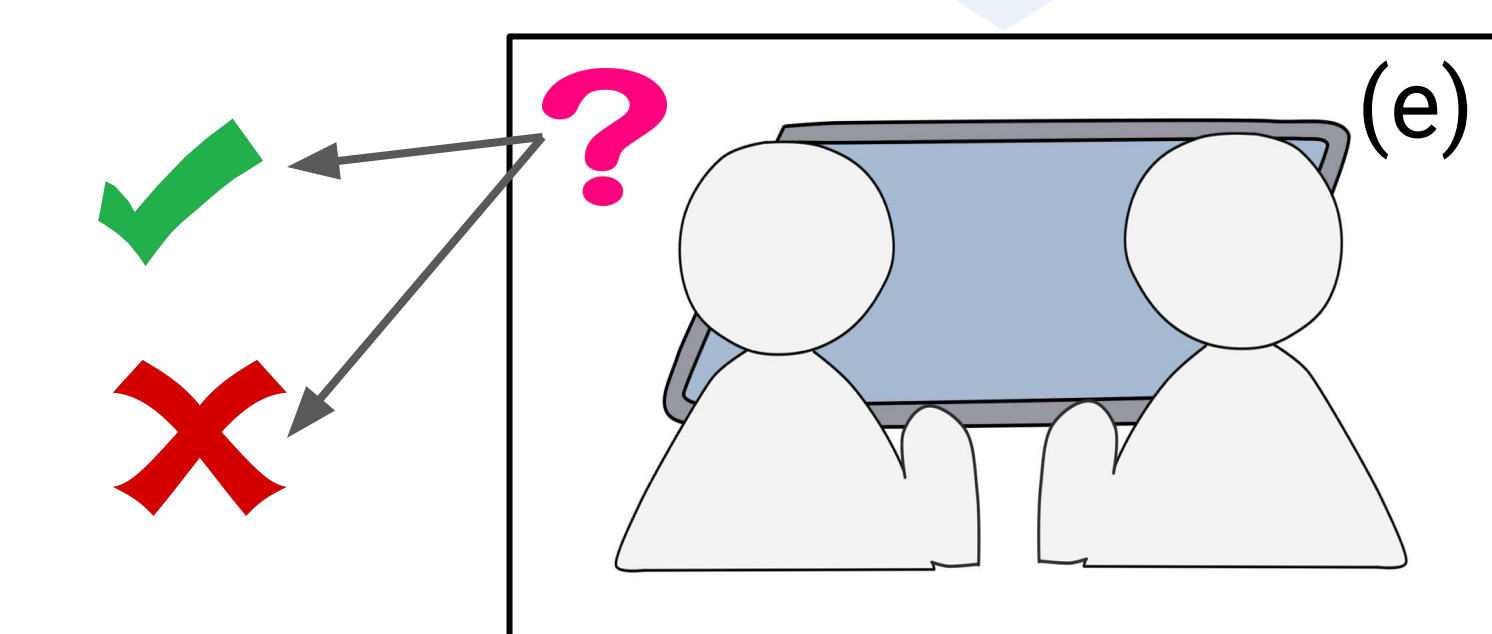
1. We will look at one user with and without a touchscreen to compare user behavior.



1. If users like touchscreen, we will investigate single-user-touch partner programming.



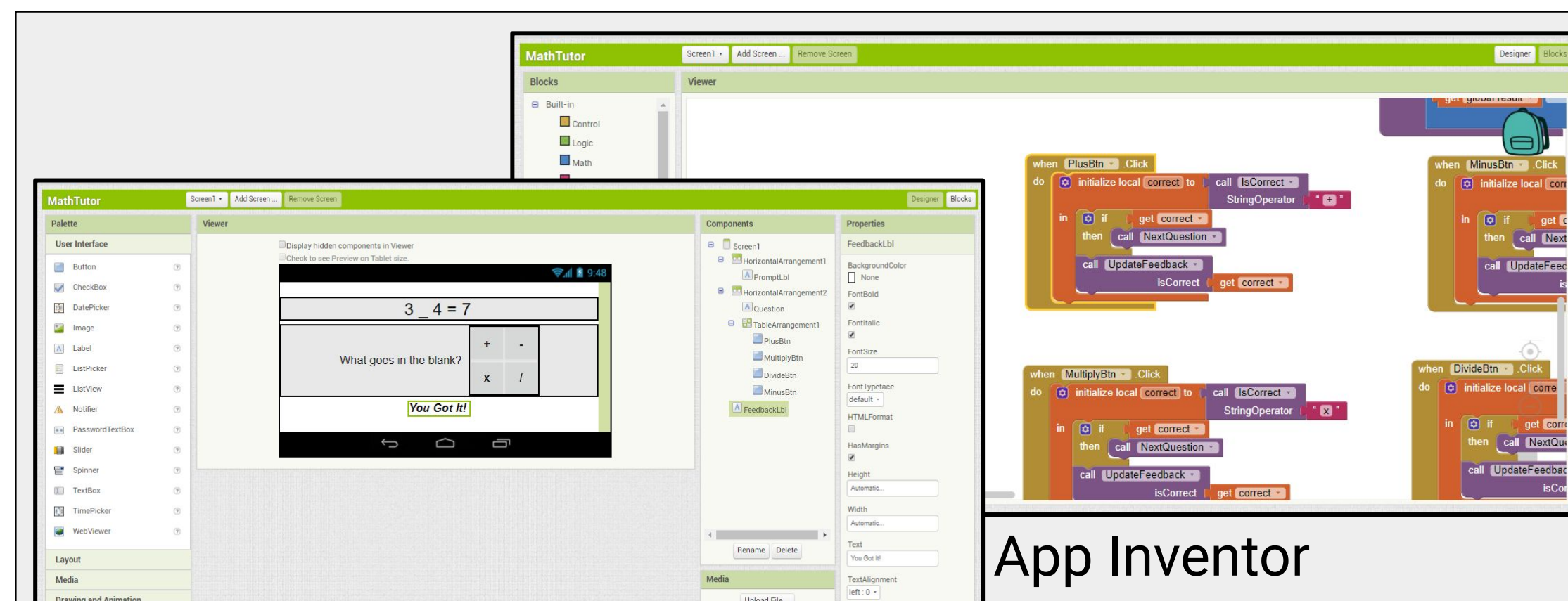
We will work to develop a multi-user-touch system for App Inventor if these experiments suggest the system would be beneficial.



Pair Collaboration

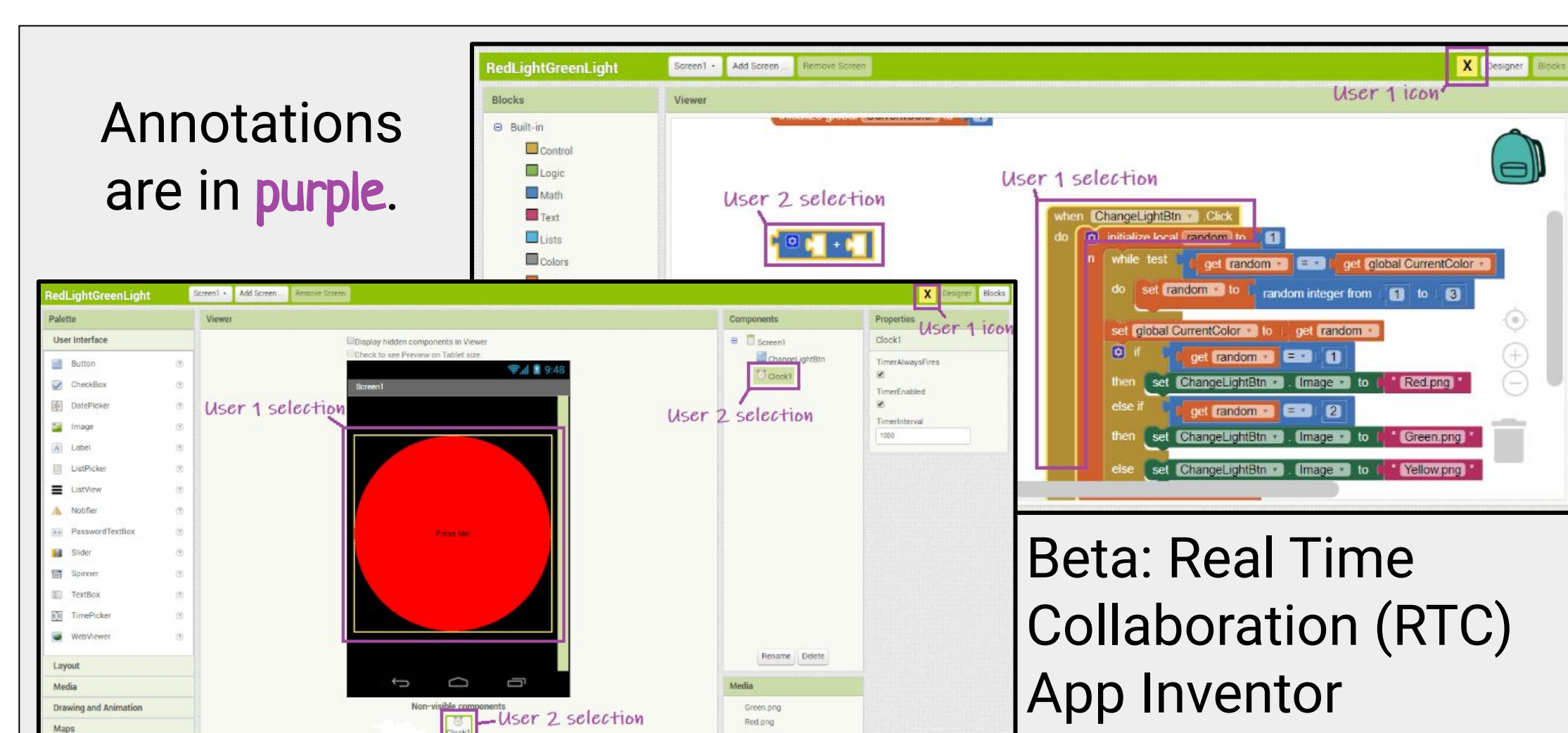
Pairs can collaborate in App Inventor in two ways:

Single screen collaboration:



One user controls the computer mouse or a touchscreen, where control might be more fluid.

Multi screen collaboration:

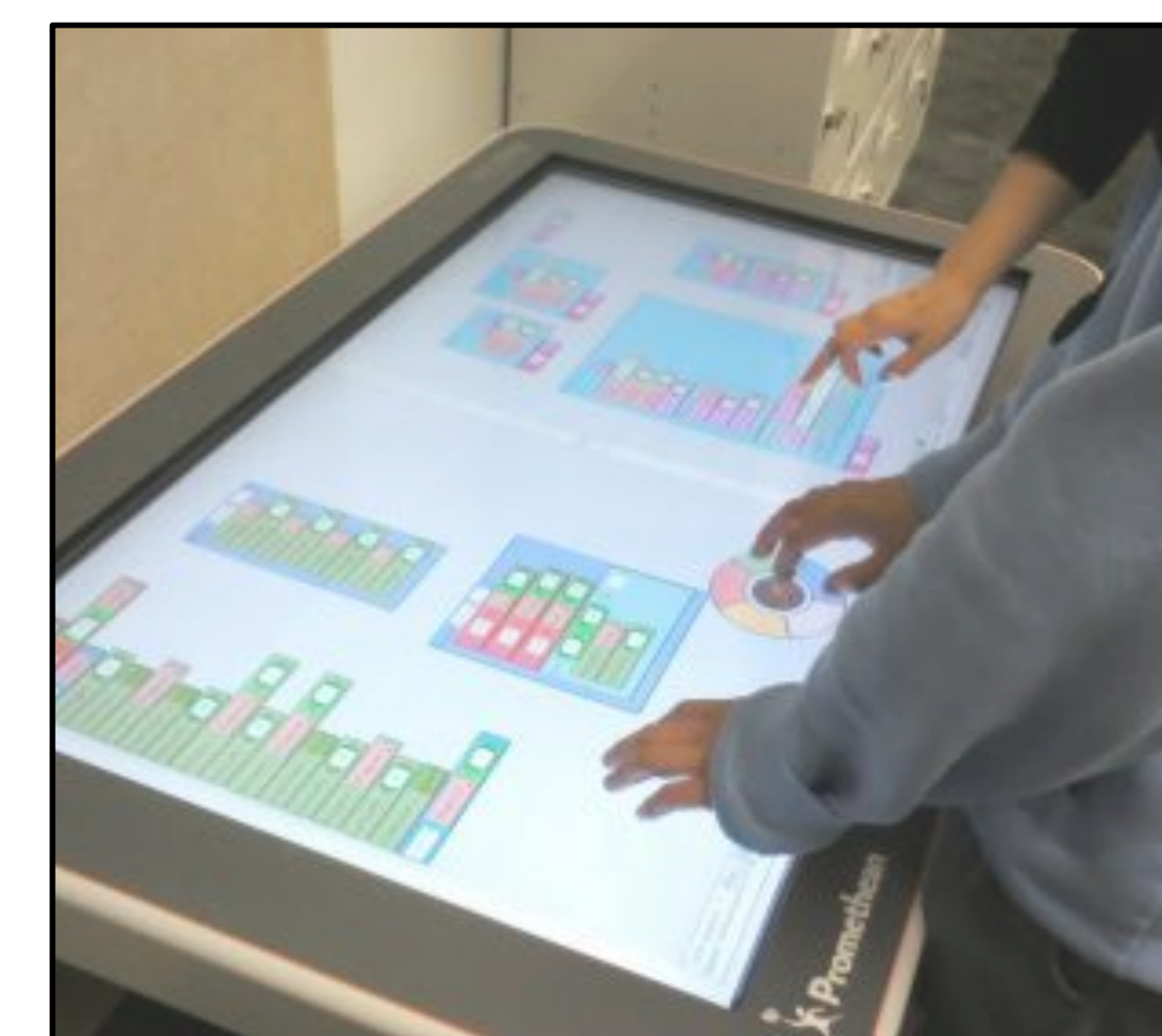


This is like Google Docs for Blocks. Two users can create components and move blocks on their own screens at the same time.

Supporting Two Users on a Touchscreen

We wish to investigate the setting where two people can touch the same screen simultaneously. Right now, Blockly and App Inventor do not recognize more than one user interaction at a time.

If we want to modify App Inventor to support touch by two users, we must first design new behaviors for its interface. When two users interact with one screen, there can be conflicts. Sometimes, the application needs to distinguish between the two users; some solutions exist but are not suitable for App Inventor.



Tabletop Grace (1) gives users separate spaces to work in.



(left to right) Carpus (2), Multitaction with fiducial markers, and DiamondTouch can distinguish users using specialized equipments.

For example, Tabletop Grace is a multi-user blocks programming environment that partitions its interface into different work areas for each user. Partitioning works for collaboration, but it requires large surfaces that everyday users don't have (3). Researchers have also built external devices to distinguish users, but these are very specialized and not available to the general public.

Acknowledgements

Researchers were funded by the Science Center 2018 Summer Research Program through T.T. and W.F. Chao Summer Scholars Program in Natural Sciences (Xu) and Wellesley Fund for Student Faculty Research (Tinney). We are thankful for the time and guidance of many people. In particular, we thank our advisor, Lyn Turbak; Blockly's lead developer, Rachel Fenichel; Mike Tissenbaum and Mark Sherman on App Inventor's team; and Orit Shaer and Clarissa Verish from the Wellesley HCI Lab.

References

- (1) Ben Selwyn-Smith, Michal Homer and Craig Anslow, "Towards collaborative block-based programming on digital tabletops," 2017 IEEE Blocks and Beyond Workshop (B&B), Raleigh, NC, 2017, pp. 57-60. doi: 10.1109/BLOCKS.2017.8120412
- (2) Raf Ramakers, Davy Vanacken, Kris Luyten, Karin Coninx, and Johannes Schöning. 2012. Carpus: a non-intrusive user identification technique for interactive surfaces. In Proceedings of the 25th annual ACM symposium on User interface software and technology (UIST '12). ACM, New York, NY, USA, 35-44. doi: 10.1145/2380116.2380123
- (3) Stacey D. Scott, M. Sheelagh T. Carpendale, and Kori M. Inkpen. 2004. Territoriality in collaborative tabletop workplaces. In Proceedings of the 2004 ACM conference on Computer supported cooperative work (CSCW '04). ACM, New York, NY, USA, 294-303. doi: http://dx.doi.org/10.1145/1031607.1031655