Programming on a tangible object or simulation to learn elementary CS concepts?

Grégoire Fessard, Ilaria Renna, Patrick Wang
Institut Supérieur d'Électronique de Paris
Human-Machine Interaction Research Group

patrick.wang@isep.fr
https://patrickwang.fr
@Gkz_

INTRODUCTION

Physical computing is the "design of tangible and interactive objects using programmable hardware" [3]. Research has shown that physical computing increased learners' motivation and creativity [7], facilitated the inclusion of underrepresented populations in CS [6], and resulted in significant learning gains [1]. However, we could not find studies comparing learning gains when a learner programs a tangible object or an exact equivalent digital simulation of it.

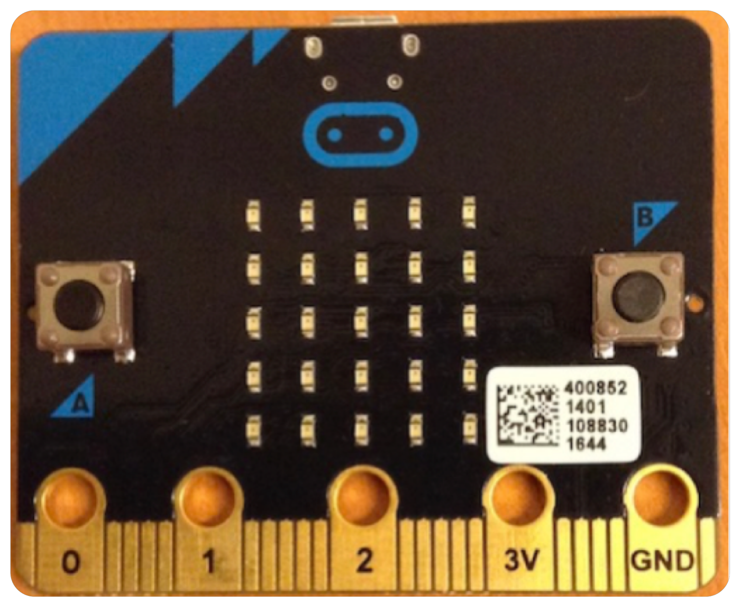
Research question:
Are there differences in learning gains when programming a tangible object or an exact equivalent digital simulation?

==> Focus on variables, conditionals, and loops <==

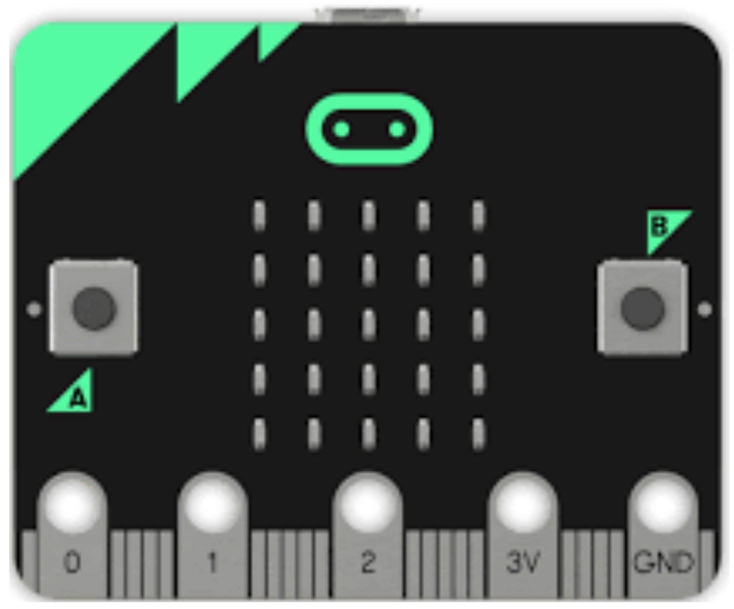
PROGRAMMABLE OBJECT

The BBC micro:bit is a pocket-sized computer [2]. It is already used to introduce CS and programming in various contexts [5,7]. The micro:bit can be programmed using blocks (see demo) to manage components such as a 5x5 LED grid or 2 push buttons.

The micro:bit comes both as a programmable **tangible object** or a **digital simulation**. Both accept block-based programs and will have the same results for a same program.



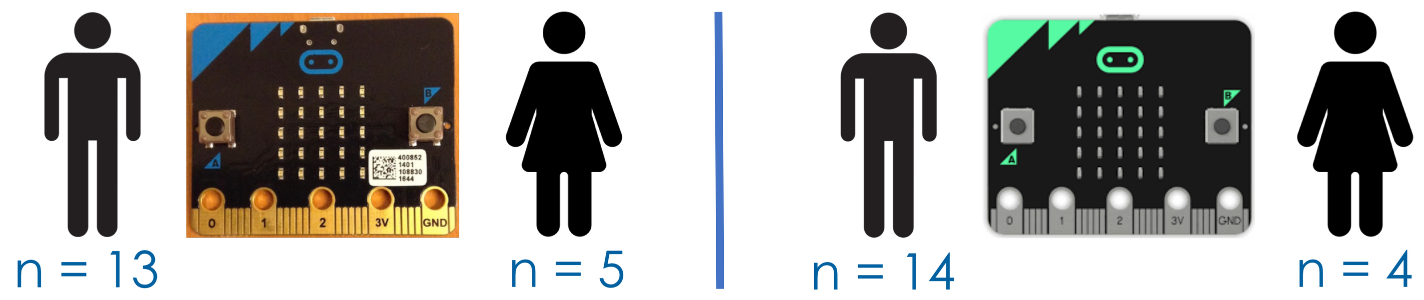
Tangible micro:bit card



Simulated micro:bit card

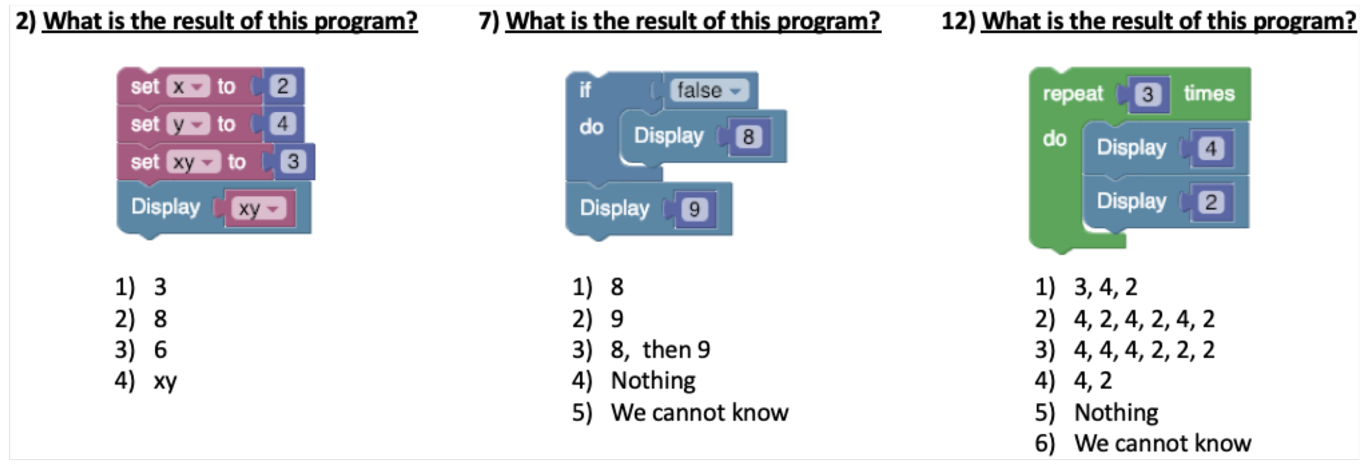
METHODOLOGY

The 1-day experiment involved 36 participants aged 14-17 with little or no prior knowledge of programming. Participants were separated *a priori* into 2 groups of 18 based on their gender, age, and prior knowledge.



Pre- and post-test

The pre- and post-test consists of 14 MCQ: 4 on variables, 5 on conditionals, and 5 on loops. Each question concerns a single programming concept so that we can evaluate the learning gains on each concept. For each concept, questions were ordered to be increasingly more difficult. Incorrect answers were based on students' misconceptions reported in the literature [4].



2) What is the result of this program?

```
set x to 2
set y to 4
set xy to 3
Display xy
```

1) 3
2) 8
3) 6
4) xy

7) What is the result of this program?

```
if false
do Display 8
Display 9
```

1) 8
2) 9
3) 8, then 9
4) Nothing
5) We cannot know

12) What is the result of this program?

```
repeat 3 times
do Display 4
   Display 2
```

1) 3, 4, 2
2) 4, 2, 4, 2, 4, 2
3) 4, 4, 4, 2, 2, 2
4) 4, 2
5) Nothing
6) We cannot know

Programming exercises

Participants worked on 15 exercises in total: 1 to introduce the programming environment, 4 on variables, 3 on conditionals, 4 on loops, and 3 on mixed concepts. For each concept, exercises were ordered to be increasingly more difficult, with lesser and lesser guided wordings. Participants were prompted to test their solutions as often as possible. See demo for examples of programming exercises.

RESULTS

Validation of the protocol

We first checked that the *a priori* repartition did not introduce any initial bias in the experiment. We compared mean pre-test scores between the two groups and found that there was **no significant difference in terms of prior programming knowledge** (t = 1,499, p = 0.145).
The programming exercises were effective as **both groups significantly improved their overall scores** from pre-test to post-test, with significant learning gains on conditionals and loops but not on variables.

Comparison of learning gains

Results related to learning gains for both group and between the two groups

Group	Overall	Variables	Conditional structures	Iterative structures
Tangible	t = 8.574 **p << 0.01**	t = 0.325 p = 0.375	t = 6.556 **p << 0.01**	t = 6.174 **p << 0.01**
Simulation	t = 5.242 **p << 0.01**	t = 1.046 p = 0.155	t = 6.761 **p << 0.01**	t = 3.915 **p << 0.01**
X-comparison	t = 0.687 p = 0.497	N/A	t = 1.425 p = 0.163	t = 0.586 p = 0.562

No significant difference in learning gains was found between the two groups (overall and by concept). There is no analysis for variables because no learning gain was found during the experiment for both groups.

CONCLUSIONS

While both groups improved their post-test scores, no significant difference in learning gains was found when programming a tangible object or an equivalent digital simulation. These results raise new questions:

- What would be the results with more physical interaction with the tangible object?
- Are there different programming strategies when working with a tangible or digital object?

REFERENCES
[1] Cliburn, D. C. (2006). Experiences with the LEGO Mindstorms throughout the undergraduate computer science curriculum. In *Proceedings. Frontiers in Education. 36th Annual Conference* (pp. 1-6). IEEE.
[2] https://microbit.org
[3] Przybylla, M., & Romeike, R. (2014). Key competences with physical computing.
[4] Qian, Y., & Lehman, J. (2017). Students' misconceptions and other difficulties in introductory programming: A literature review. *ACM Transactions on Computing Education (TOCE)*, *18*(1), 1.
[5] Schmidt, A. (2016). Increasing computer literacy with the BBC micro: bit. *IEEE Pervasive Computing*, *15*(2), 5-7.
[6] Sentance, S., & Schwiderski-Grosche, S. (2012). Challenge and creativity: using. NET gadgeteer in schools. In *Proceedings of the 7th Workshop in Primary and Secondary Computing Education* (pp. 90-100). ACM.
[7] Sentance, S., Waite, J., Hodges, S., MacLeod, E., & Yeomans, L. (2017). Creating Cool Stuff: Pupils' Experience of the BBC micro: bit. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (pp. 531-536).

École d'ingénieurs du numérique