



TurtleTable: Learn the Basics of Computer Algorithms with Tangible Interactions

Iza Marfisi-Schottman^(✉), Sébastien George, and Marc Leconte

Le Mans Université, EA 4023, LIUM, 72085 Le Mans, France
{iza.marfisi, sebastien.george,
marc.leconte}@univ-lemans.fr

Abstract. Computer programming has become a basic skill and is now part of the curriculum taught in middle school. In France, this educational reform was very sudden and many teachers are not yet trained adequately to teach this subject, leaving them in search of guidelines and tools to help them. It is in this context that we propose TurtleTable, an educational game, who has two main originalities. First of all, this game is not based on writing a computer program, like most of the other tools, but rather on executing a given program, step by step. These two approaches are complementary and essential to apprehend the logic behind computer programming. Secondly, TurtleTable is played in groups of three, by manipulating Tangible objects on an Interactive Tabletop (TIT). The players have to collaboratively move the objects on a grid, by following the instructions of the program on the screen. The game immediately indicates which instructions are executed correctly, allowing the learners to correct their mistakes. TurtleTable was tested by 59 middle school students. We developed four different interactions for TurtleTable (TIT, tabletop, tablet and computer) to compare their effects on motivation, collaboration and learning. The preliminary analysis show that the students really appreciated the game, especially the TIT version, that encouraged collaboration and gave them the impression they learned more.

Keywords: Tangible object · Interactive tabletop · Serious games · Learning games · Algorithm · Computer programming

1 Tools for Teaching the Basics of Computer Programming

Digital tools are an important part of our modern society and computer science is now considered as a basic skill that needs to be taught as early as possible, along with math and science. In many countries, such as France, the new official curriculum adopted in September 2016, that includes computer programming, was very sudden, leaving many teachers unprepared. Most teachers do not have any training on these new skills and are in need of guidelines and tools to help them. Many passionate teachers and associations have created blogs with examples of activities [1]. Most of these activities are “unplugged”, meaning they can be done without a computer or specific digital equipment. Among these activities, one seems to be particularly appreciated: the *stupid robot* game. This game has many variations for 2 to 4 players but basically one player

writes a program with instruction cards (e.g. move forward, turn left, turn right) and the other player executes it like a computer by moving and turning on a checkered floor mat of tiles. Teachers can also use free digital tools to help their students learn how to program with drawings [2], diagrams [3], blocs [4] or even with real computer languages [5]. These digital tools have the advantage of correcting the students directly and offering many exercises with various difficulty levels. These digital games are based on activities that consist in writing a program to move a robot (or an animal) in an environment. These activities are often presented in a game-like environment, and challenge the students with tasks that progressively get harder (e.g. move the robot to collect all the candy, only use three lines of code). The basic concepts of computer programming (i.e. variables, conditions, loops and functions) are gradually introduced as the levels of the games get harder. This type of activity, based on the construction of a program, is essential to understand algorithms, but often leads learners to execute their code compulsively and irrationally, until it works. Indeed, this type of game doesn't encourage the players to simulate the program in their head, before pressing the execute button, to foresee how it's going to behave. Yet, this is a key competency to master computer programming. Many unplugged activities, such as the example given above, help develop this competency by working on the execution of a program step by step. However, these activities require the constant presence of the teacher to explain the activity and to make sure the students do not make mistakes.

In order to fill the gap, we propose TurtleTable, a digital game that teaches students how to execute a program, just like a machine. The goal is to help them understand how a computer program is executed and understand the logic of algorithms. In the third part of this paper, we detail the innovative Human Computer Interactions (HCI) with tangible objects and interactive tabletops, we designed for TurtleTable, in order to enhance motivation, collaboration and learning. We then discuss the results of a test lead with 59 middle school students, from 14 to 16 years old. In the last section, we provide insight on the improvements that could be brought to TurtleTable.

2 TurtleTable: Learn How to Execute a Computer Program

TurtleTable is complementary to other digital tools that teach computer programming. First of all, the game is based on executing an algorithm, step by step, whereas the other tools are based on building algorithms. TurtleTable is also meant to foster collaborative learning whereas more classical computer applications are typically played alone. In this section, we will present the TurtleTable's game, the programming concepts introduced in each level, and the functionalities that allow real-time corrections and scoring. TurtleTable was named after *turtle graphics*, a tool that was used to introduce computer programming, with the LOGO language [6] since the 1960s. It has since inspired generations of programmers and countless educational games. These games come in all shapes: card games, digital games, simulators, kits to build your own robot... and now this one, on an interactive table.

2.1 Game Objective

The objective the TurtleTable game is simple: execute the program on the left of the screen (Fig. 1, panel A) by moving the objects on the grid (Fig. 1, panel B). The program, written in a pseudo-language, is composed of basic instructions (e.g. *move obj_1 3 steps forward*, *turn obj_2 45° to the right*), and can contain variables, conditions, and loops. If the instruction is executed correctly (i.e., the object was placed on the right spot or turned correctly), the instruction is colored in green and the students can execute the next instruction. In addition, the movements of the objects are materialized by a line on the grid, that creates a drawing at the end of the level. On the top of the interface (Fig. 1, panel C), the players can access the 20 levels and see their scores.

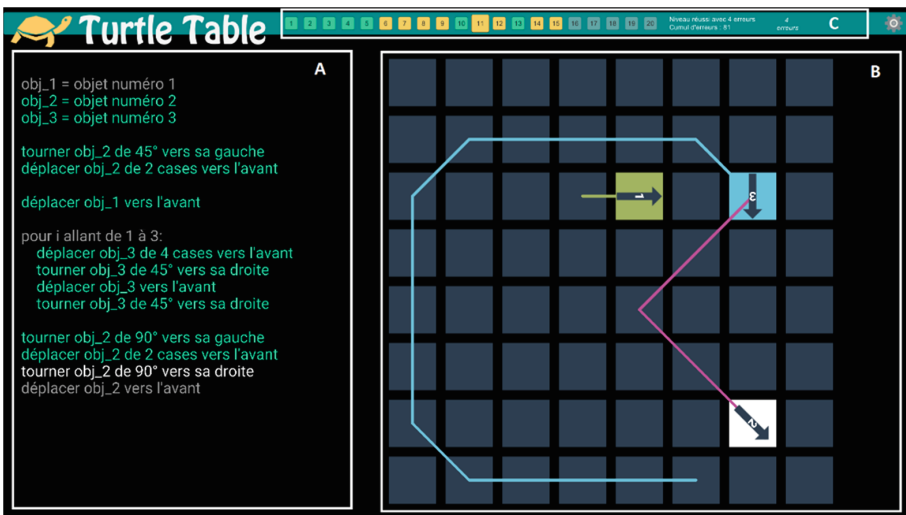


Fig. 1. Interface of TurtleTable, level 11

2.2 Introducing Basic Computer Programming Concepts

TurtleTable has 20 levels that progressively introduce the basic concepts of computer programming. Level 4 introduces the notion of variables ($a = 4$), Level 8 introduces the repeat loop (*for i from 1 to 3 do {...}*), level 12 introduces the simple conditional instructions (*if i == 2 {...}*), level 14 introduces the conditional instructions (*if a < b {...} else {...}*) and finally, level 15 introduces the repeat until loop (*repeat {...} until a > 3*). These concepts were found in the official guidelines for teaching computer programming in middle school. The explanations to introduce these concepts and the order in which to present them were inspired by the tutorials in the CODE games [4] that are widely used in American schools. As the players get further in the game, these concepts accumulate. For example, level 16 has a conditional instruction in a loop and level 20 has a loop in another loop. The levels therefore get more complicated and challenging.

2.3 Real Time Correction and Scoring

If the students don't move an object correctly, the image of this object shakes and they have to execute the instruction again, until they get it right. TurtleTable offers several visual aids to help the learners execute the algorithm correctly: the current instruction is colored in white, the instruction executed correctly in green and the instruction executed with an error appears in red. All of these visual aids are optional and can be tuned off, making the levels substantially more challenging, especially those with loops.

TurtleTable keeps track of the number of errors for each level and shows them on the top right of the game interface (Fig. 1, panel C). If the level is finished without any errors, the level's button turns green. If the level is finished with one or more errors, it is colored in orange. The objective is to encourage players to take the time to think and discuss with each other, before moving the objects. Once all the instructions are done correctly, the next level automatically shows up. When the 20 levels are finished, the game shows a personalized message with the total number of errors, urging the players to improve their score by doing the levels again.

Now that we have described the TurtleTable game, the next section will present the innovative interactions it offers to increase motivation, collaboration and learning.

3 Tangible Objects with Interactive Tabletops

Our goal is to provide a tool that teachers can use in their classes, to introduce the basic concepts of computer programming to middle school students. Since programming is completely new to the students and most of the teachers, we wanted to create a fun and social activity that would encourage everyone to participation and collaborate. Research in Human Computer Interactions (HCI) offers empirical evidence that the use of Tangible objects on Interactives Tabletops (TITs) fosters student engagement and may enhance learning through collaborative learning methods [7, 8]. The use of TITs appears especially effective for children, who are used to learning by manipulating objects. We therefore choose to use this type of interaction for TurtleTable. In the next section, we provide details on why and how these interactions were designed.

3.1 Tangible Objects

Several studies have proven the fact that tangible objects help students collaborate and appropriate concepts, while making problem solving more fun and pleasant [9, 10]. In addition to the attractiveness of this new kind of interaction, the main advantage of tangible objects is that they are an invitation to action and physically engage the body in the activity. Toy manufacturers already make games for learning computer programming that are composed of software, installed on a computer, and tangible objects, that communicate with this software. These objects are either used for the construction of the program [11–13], or for the construction of the robot that will execute the program [14]. For TurtleTable, we decided to use tangible objects to execute a given program (Fig. 2). Apart from the first two levels, all the others require the use of two or three objects. The objective was for each player to participate by moving his/her object.

The objects were custom-made with a 3D printer. The top is hollowed out so that the players can see the screen underneath. The objects have exactly the same shape and therefore can be interchangeable and turned. After testing the game with a small group of students, we realized it was more intuitive to add a sticker that indicates a number and the front of the object that would correspond with the number and front of the virtual object it is on.

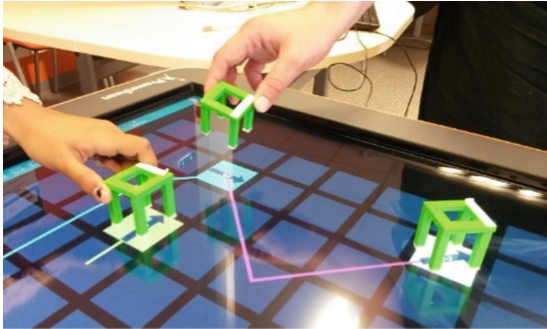


Fig. 2. Tangible objects manipulated to play TurtleTable

3.2 Interactive Tabletops

The use of an interactive tabletop also facilitates collaboration between team members [15]. The large work space allows students to visualize the interface and interact with it equally, unlike computers or tablets where, most of the time, one person interacts and the others observe. The interactive tabletop used for TurtleTable is a *ActiveTable* produced by Promethean. This model does not have RFID captors or a camera under the screen, that can detect *ARTags*. It is equipped with laser beams that swipe the surface of the screen. A touch on the screen is detected when a finger or an object cuts the lasers. We therefore designed the tangible objects with “legs” that could be detected by the table and wrote a protocol to recognize it, by detecting the space between the four legs, so that it would not be mistaken for fingers.

4 Tests Lead with Middle School Students

4.1 Objectives

The game objective in TurtleTable (i.e. correctly executing an algorithm) is based on unplugged activities that are largely used in class and that have proven their effectiveness for teaching computer programming. The main benefit of TurtleTable is the fact that it corrects the students automatically and therefore allows the teachers to spend more time helping those in need. Consequently, it does not seem very interesting to compare classes that use TurtleTable with classes that do not. The benefits of the innovative TIT interactions, however, remains uncertain. The aim of this first

exploratory study is therefore to measure if the use of TITs has a positive effect on **motivation, collaboration** and **learning**. TIT interactions offer two original features: the tangible objects and the large workspace, offered by the interactive tabletop. In order to measure the effect of each of these features, we decided to set up two control groups: one without tangible objects, but with the big screen, and another without tangible objects or the big screen. In order to have a neutral control panel, we also wanted a group to play the game on a computer, with basic mouse and keyboard interactions. We therefore developed four versions TurtleTable (Fig. 3).



Fig. 3. Middle School students playing with the four versions of TurtleTable

- The **TIT** version is played with Tangible objects on an Interactive Tabletop. The players move and turn the object on the grid by placing the tangible object on top of it and doing the desired movement.
- The **Tabletop** version is played on an interactive tabletop, without tangible objects. The objects are moved with basic tactile interactions: click to select and then swipe to move forwards or backwards. We developed a two finger compass-like rotation movement to turn the objects.
- The **Tablet** version is played on a tablet. The tactile interactions are the same as the tabletop version.
- The **Computer** version is played on a computer. To move an object, the player must first click on it with the mouse and then press the UP arrow to move forward or the DOWN arrow to move backward. The LEFT and RIGHT arrows are used to turn the object.

4.2 Data Collection and Experimental Protocol

In order to measure the effect of the of TITs on motivation, collaboration and learning, we collected different types of data. First, we designed an individual **pre and post-test** to measure the knowledge of the subjects regarding the basic concepts of computer programming (variables, loops and conditions). This test is composed of 10 exercises that are similar to TurtleTable's levels: the goal is to execute a program by drawing the path that the objects do on a grid with a pencil. In order to evaluate if the subjects are capable of using the right syntax and writing optimized code, the two final exercises of the test consist in writing the algorithm for a given drawing on a grid. All the **usage tracks** of TurtleTable are also collected. This provides precise indications on the number of errors and the time it took to complete a level. We also **filmed all the sessions** in order to analyze the interactions between the subjects. In order to understand these interactions, we also set up **focus groups** during which we discussed these interactions. Finally, we designed a **final survey** to collect the subjects' insight on TurtleTable and its HCI.

A first experimentation was led with 59 students of the La Salle middle school (Laval, France). There were 30 girls and 29 boys, from 14 to 16 years old. These students came to our research center for half a day in December 2017. This field trip was presented by their teachers as a first initiation to computer programming. The large majority of these students had no experience in computer programming or algorithms: only 5 subjects were familiar with scratch programming, as they went to a computer science workshop during their lunch break.

The students were separated in four groups and assigned different versions of TurtleTable: 16 in the TIT group, 16 in the Tabletop group, 15 in the Tablet group and 12 in the Computer group. Each group was composed of 4 teams of three to four students. We asked the teachers to prepare the groups and the teams so that the level of the students would be evened out. The experimentation was planned as such:

1. The subjects had 15 min to answer the individual pre-test.
2. The teams had 60 min to play with the version of TurtleTable they were assigned. We asked them to go at least up to the level 15, in order for them to have seen all the programming concepts. During this session, we explicitly asked the teachers not to help their students.
3. The subjects had 15 min to answer the individual post-test.
4. The teams had 20 min to test the other versions of TurtleTable. During this session, several teams were interviewed in focus groups.
5. The subjects had 20 min to individually answer the final survey.

4.3 Results

First of all, **TurtleTable was greatly appreciated by the students and the teachers**. Almost all the groups managed to finish the 20 levels of the game in the allotted time and were competing to get the lowest number of errors by redoing all the levels. The analysis of the post-test shows that more than half of the students mastered the two first concepts presented in the game: variables (6/8) and repeat loops (12/16). Considering only a few students had motions of algorithms, this is a good result for their first hour

of programming class. In the next sections, we discuss the differences between the different HCI versions of the game, in terms of motivation, collaboration and learning.

Motivation

In the final survey, the subjects indicated the version of the game they preferred (Fig. 4). Almost **2/3 preferred the TIT version**, 1/5 preferred the Tablet version and very few preferred the Computer of the Tabletop versions. These results are confirmed by the words they used to qualify the versions of TurtleTable (word clouds in Fig. 4) and the discussions lead during the focus groups. The TIT version is mostly qualified as “fun” and “interesting”, the Tablet version is viewed as “interesting” and “educational”, the Computer version as “amusing” by half of subjects but “boring” by the other half and finally, the Tabletop version is seen as “difficult”.

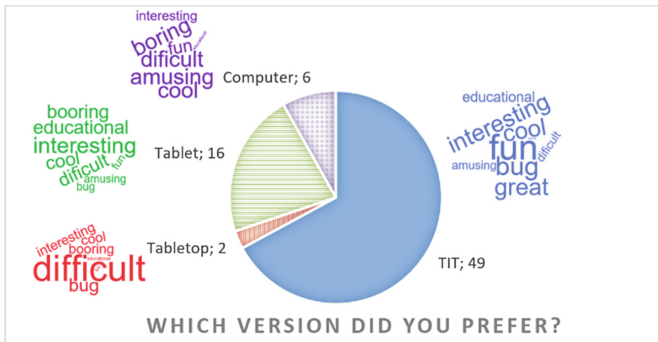


Fig. 4. Subjects preferred version of TurtleTable

Collaboration

The analysis of the videos shows that the teams using TITs and Tabletops had **much more interaction with each other** than the other groups. They talked more to each other and were all looking at the screen. In addition, the use of TITs facilitated collaboration: at least **two or three members of the team were engaged in the activity** by moving an object or helping the others move the objects. This is confirmed by the words used by the subjects in the final survey such as “convivial” and “group work”, that only appear for the TIT version. With Tabletops, only one or two team members were engaged. With Tablets and Computers, the players would take turns playing the levels and the others would often look away while they were waiting for their turn.

Learning

Each pre and post-test was analyzed to determine the instructions that had been done correctly. The mean scores out of 100 for the pre-test was: TIT-14.00, Tabletop-15.42, Tablet-13.86 and Computer-15.54. For the post-test: TIT-28.38, Tabletop-27.85, Tablet-26.80 and Computer-26.85. The ANOVA test showed there was no significant differences between progression of the four groups. However, if we analyze the knowledge gain in each concept separately (bottom of Fig. 5), the subjects who used TITs seemed to have significantly better results for the “Repeat loop”. The analysis of

the questionnaires also revealed that more **subjects using TITs and Tabletops said they had learned a lot in the survey** (top of Fig. 5). In addition, the subjects qualified the TIT version with adjectives such as “educational” and “instructive”, three times more than the other versions.

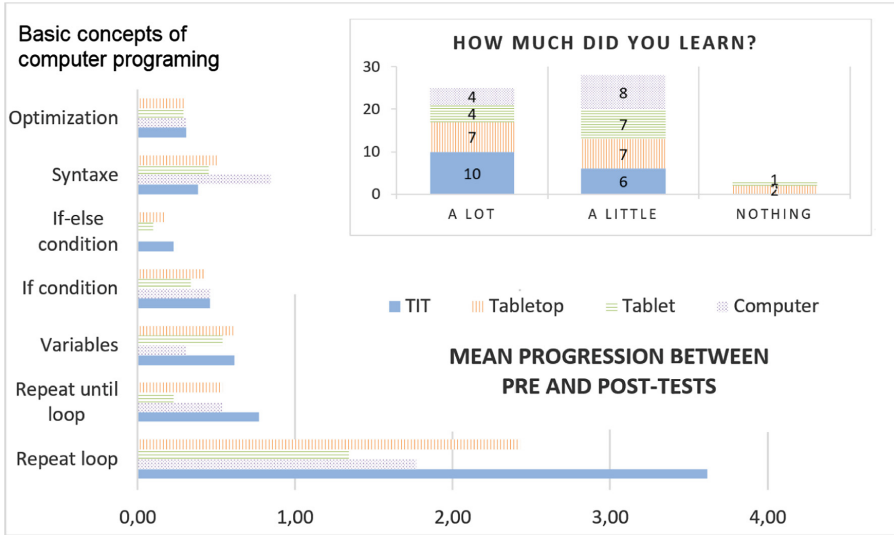


Fig. 5. Objective and subjective data to measure knowledge gain

5 Conclusion, Limitations and Perspectives

In this paper, we propose TurtleTable, a game that introduces the basic concepts of computer programming. The game is made to be used in class, in teams of three to four. It offers an original approach to apprehend the logic behind computer programming: the players have to execute a program by moving objects on a grid. The game offers real-time correction and a scoring mechanism that encourage players to take time to discuss the next move with the others. In order to maximize motivation, collaboration and learning, we designed TIT interactions: the students play by manipulating Tangible objects on an Interactive Tabletop. The first test, lead with 59 middle school students, was a success. In less than one hour, more than half of the students, with no prior knowledge in algorithms, had mastered the concepts of variables and the repeat loops. We developed four different versions of HCI (TIT, Tabletop, Tablet and Computer) to compare their effect on motivation, collaboration and learning. Even though the preliminary analysis did not show a significant increase in learning with TITs, this interaction definitely facilitated collaboration among the group and was a factor of great motivation.

Further analysis of the data collected during the test needs to be done. For example, we did not take into account the fact that some students were sitting and others standing when using the TIT and the Tabletop version of TurtleTable. In addition, we observed

interesting behavior that needs to be analyzed. For example, with the TIT version, the students would rotate their bodies to help them turn the object in the right direction. We are currently looking into standardized interaction analysis to annotate the hours of videos collected during the teste. Finally, in order to test the effect of the size of the workspace, we intent on developing another version of TurtleTable that works with tangible objects on a tablet. We are looking into the possibility of painting the objects with conductive paint so that they are recognized by the tablet.

References

1. One of the main French website with various unplugged activities to help teachers teach computer science. <https://pixees.fr/informatique-en-primaire-les-activites-debranches/>. Accessed 17 July 2018
2. LightBot. <https://lightbot.com/>. Accessed 17 July 2018
3. RobotProg. <http://www.physicsbox.com/indexrobotprogen.html>. Accessed 17 July 2018
4. Scratch. <https://scratch.mit.edu/>, <https://code.org/learn>. Accessed 17 July 2018
5. Logo Turtle. <https://www.tortue-logo.fr/en/logo-turtle>. Accessed 17 July 2018
6. Turtle Academy. <http://turtleacademy.com/>. Accessed 17 July 2018
7. Kubicki, S., Pasco, D., Arnaud, I.: Using a serious game with a tangible tabletop interface to promote student engagement in a first grade classroom: a comparative evaluation study. *Int. J. Inf. Technol.* **4**, 381–389 (2015)
8. Schubert, M., Serna, A., George, S.: Using collaborative activities on tabletops to enhance learning and knowledge transfer. In: *Proceeding of the International Conference on Advanced Learning Technologies (ICALT)*, Rome, Italy, pp. 610–612 (2012)
9. Schneider, B., Jermann, P., Zufferey, G., Dillenbourg, P.: Benefits of a tangible interface for collaborative learning and interaction. *IEEE Trans. Learn. Technol.* **4**(3), 222–232 (2011)
10. Price, S., Pontual Falcão, T.: Where the attention is: discovery learning in novel tangible environments. *Interact. Comput.* **23**(5), 499–512 (2011)
11. Scharf, F., Winkler, T., Herczeg, M.: Tangicons: algorithmic reasoning in a collaborative game for children in kindergarten and first class. In: *Proceedings of the International Conference on Interaction Design and Children*, NY, USA, pp. 242–249 (2008)
12. Bers, M.U., Horn, M.S.: Tangible programming in early childhood: revisiting developmental assumptions through new technologies: Childhood in a digital world. In: Berson, I.R., Berson, M.J. (eds.) *High-tech tots: childhood in a digital world* Greenwich, Osmo (2009)
13. Magnenat, S., Ben-Ari, M., Klinger, S., Sumner, R.W.: Enhancing robot programming with visual feedback and augmented reality. In: *Proceedings of the Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE)*, Vilnius, Lithuania, pp. 153–158 (2015)
14. Lego Mindstorms. <http://www.lego.com/en-us/mindstorms>. Accessed 17 July 2018
15. Shaer, O., et al.: The design, development, and deployment of a tabletop interface for collaborative exploration of genomic data. *Int. J. Hum Comput Stud.* **70**(10), 746–764 (2012)