



Contents lists available at ScienceDirect

SoftwareX

journal homepage: [www.elsevier.com/locate/softx](http://www.elsevier.com/locate/softx)

Original software publication

# Little AI: Playing a constructivist robot

Olivier L. Georgeon\*

Université Claude Bernard Lyon, LIRIS UMR5205, F-69622, France

National Research Nuclear University MEPhI (Moscow Engineering Physics Institute) Moscow, Russian Federation



## ARTICLE INFO

### Article history:

Received 10 March 2017

Received in revised form 29 June 2017

Accepted 29 June 2017

### Keywords:

Pedagogical game

Artificial intelligence

Developmental learning

Constructivist learning

## ABSTRACT

Little AI is a pedagogical game aimed at presenting the founding concepts of constructivist learning and developmental Artificial Intelligence. It primarily targets students in computer science and cognitive science but it can also interest the general public curious about these topics. It requires no particular scientific background; even children can find it entertaining. Professors can use it as a pedagogical resource in class or in online courses. The player presses buttons to control a simulated “baby robot”. The player cannot see the robot and its environment, and initially ignores the effects of the commands. The only information received by the player is feedback from the player's commands. The player must learn, at the same time, the functioning of the robot's body and the structure of the environment from patterns in the stream of commands and feedback. We argue that this situation is analogous to how infants engage in early-stage developmental learning (e.g., Piaget (1937), [1]).

© 2017 Published by Elsevier B.V.

## Code metadata

Current code version	v1.3
Permanent link to code/repository used of this code version	<a href="https://github.com/ElsevierSoftwareX/SOFTX-D-17-00023">https://github.com/ElsevierSoftwareX/SOFTX-D-17-00023</a>
Legal Code License	Creative Commons Zero (CC0)
Code versioning system used	git
Software code languages, tools, and services used	Swift 3
Compilation requirements, operating environments & dependencies	macOS 10, xCode 8
If available Link to developer documentation/manual	<a href="http://little-ai.com/">http://little-ai.com/</a>
Support email for questions	<a href="mailto:olivier.georgeon@gmail.com">olivier.georgeon@gmail.com</a>

## Software metadata

Current software version	1.3
Permanent link to executables of this version	<a href="https://itunes.apple.com/us/app/id1114007742">https://itunes.apple.com/us/app/id1114007742</a>
Legal Software License	Creative Commons Zero (CC0)
Computing platforms/Operating Systems	iOS 9.
Installation requirements & dependencies	
If available, link to user manual - if formally published include a reference to the publication in the reference list	<a href="http://little-ai.com/">http://little-ai.com/</a>
Support email for questions	<a href="mailto:olivier.georgeon@gmail.com">olivier.georgeon@gmail.com</a>

\* Correspondence to: Université Claude Bernard Lyon, LIRIS UMR5205, F-69622, France.

E-mail address: [olivier.georgeon@liris.cnrs.fr](mailto:olivier.georgeon@liris.cnrs.fr).

## 1. Motivation and significance

In their famous introductory book to Artificial Intelligence, Russell and Norvig state “the problem of AI is to build agents that receive percepts from the environment and perform actions” [2]. This statement expresses the *realist paradigm of IA*. In this paradigm, the AI algorithm refers to input data as “percepts” as if input data directly represented a pre-given reality.

While the realist paradigm has been successfully applied to solving predefined problems, constructivist epistemology suggests another approach to account for autonomous agents (animals or robots) actively learning from experience in an unknown environment (e.g., [1]). In the *constructivist paradigm*, the agent’s input data constitutes the result of an action initiated by the agent rather than a percept (e.g., [3,4]). In a given state of the environment, the input data may vary according to the agent’s action, and thus does not constitute a direct representation of an agent-independent presupposed reality.

Designing a constructivist AI system is challenging because the constructivist paradigm appears counter intuitive; it is difficult to understand how input data can be something other than the agent’s perception. By playing Little AI, designers better understand how to construct knowledge about the world from regularities of actions and feedback. This first-hand experience helps them imagine robots that can self-develop in the absence of ontological data about themselves and their environment [5].

Little AI relates to puzzle games where the player must learn about a hidden environment through trial and error such as Lightbulb (e.g., [6]). It, however, differs from such games by the fact that the players must also learn the meaning of their own actions (in Lightbulb, the players know that they are turning switches on and off). In Little AI, the player must simultaneously infer the structure of the environment and the functioning of the robot. Another difference is that Little AI displays a history of interactions (called the trace) that helps find the regularities of interaction visually, which is innovative for this kind of games. Additionally, each type of interaction emits a specific sound, also offering audible feedback assisting the player to detect regularities.

Little AI also relates to *programming games* such as Robocode [7] by the fact that Little AI provides a set of challenging tasks for AI designers. Similar to these games, Little AI does not implement AI or machine-learning algorithms. Algorithms that could play Little AI must be implemented externally as discussed in Section 5; examples of such algorithms are referenced in Section 4. Little AI differs from programming games by the fact that the player does not actually develop code while playing. Instead, programming occurs through the trace, which acts as a series of commands that the player can re-execute.

Little AI contributes to scientific discovery by popularizing tasks that are studied in developmental psychology and constructivist artificial intelligence. It constitutes a repository of tasks that can federate research that will grow as more challenging tasks are designed and studied by the scientific community.

## 2. Software description

To play the game, the player presses the commands at the bottom of the Game Screen (Fig. 1, left). This generates new interactions in the trace as examined in the illustrative example of Section 3. Interactions have values (positive or negative) representing how much the robot “likes” or “dislikes” the interaction. The score is the sum of the values of the last ten interactions. The level is completed when the score reaches 10. The player can move to the next or to the previous level (if unlocked) by sliding the robot to the left or to the right. The player can press the *Level* link at the top left to access the Level Screen.

The Level Screen (Fig. 1, right) shows the audio control button, the level buttons, and the donate buttons. Levels are organized in groups with regard to the kind of research questions they illustrate and the teams that study them. The first level of each group is unlocked. The next levels are unlocked when the previous levels have been completed. Level numbers are generally in the form  $g.nn$  where  $g$  is the group number and  $nn$  is the level number in group  $g$  (in the game,  $g$  is omitted when  $g = 0$ ). The player can slide the Level Screen horizontally to access other groups of levels, or slide downwards to resume playing the current level.

## 3. Illustrative example

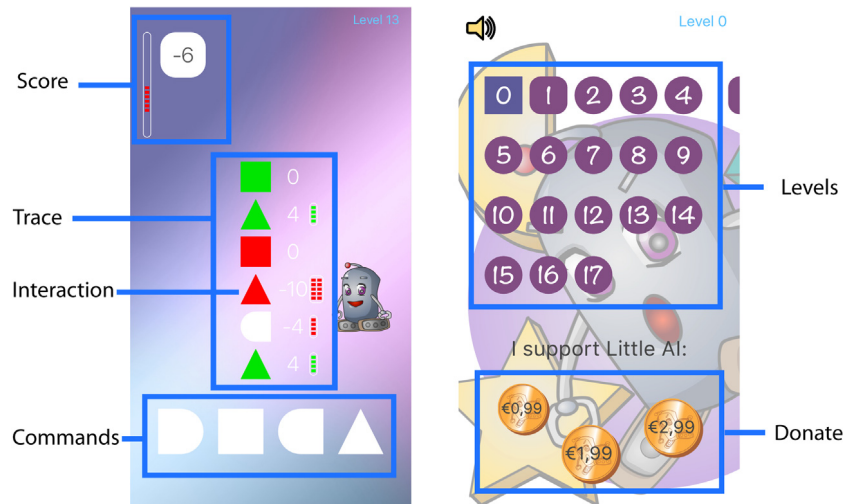
The four screenshots in Fig. 2 illustrate how to complete Level 0.00. Fig. 2a: the player presses the white square command, producing an interaction of value 0 represented by the small orange square. Fig. 2b: the player presses the white round command, producing an interaction of value 1 represented by the small green circle. The trace is scrolled, shifting the orange square one step upwards. The score now equals  $0 + 1 = 1$ . Fig. 2c: the player presses the white round command nine more times, obtaining 10 consecutive interactions of value 1. The small orange square has been scrolled up off the screen. The score reaches 10; meaning that Level 0.00 has been completed. The player is invited to open the *replay window*. Fig. 2d: the replay window shows a 3D simulation of the Little AI robot interacting with tiles. When the player presses the white square command, the robot goes backwards bumping into the orange sharp-edged tile, which the robot “dislikes” (score: 0). When the player presses the white round command, the robot goes forwards bumping into the green soft-edged tile, which the robot “likes” (score: +1).

In the higher levels, the patterns of interaction become more complex; commands may generate different results in different contexts, and the player must anticipate the results in order to complete the level. Notably, the simulation displayed in the replay window constitutes only one of many possible interpretations of the causes that could generate the patterns of interactions experienced by the player. The fact that this representation often differs from what the player had imagined illustrates how cognitive beings construct hypothetical representations of the world that may or may not correspond to the “real world”, assuming that the real world even exists.

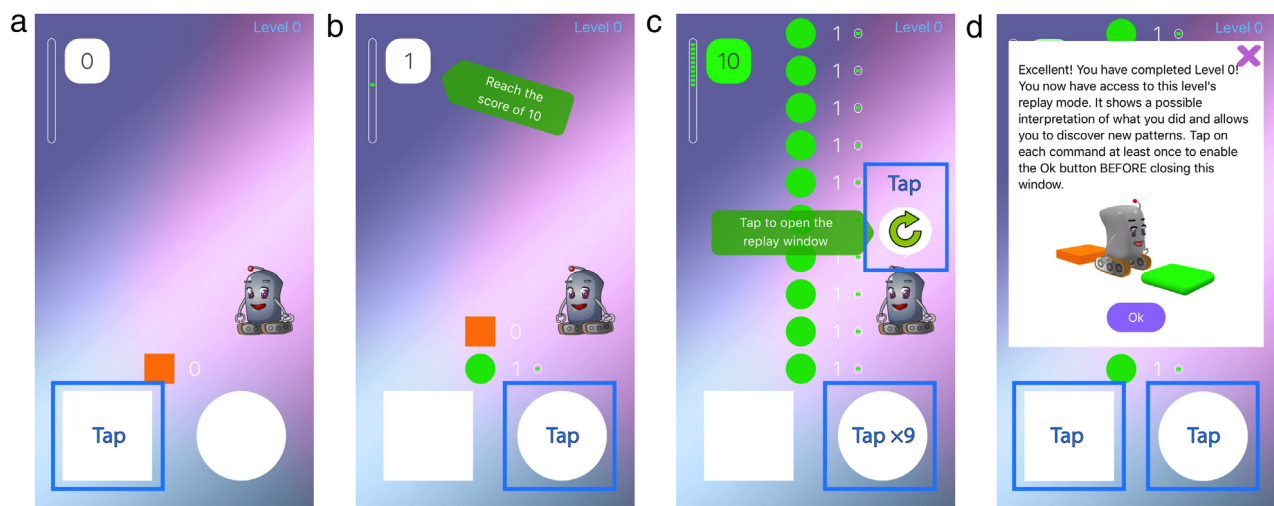
In terms of the internal design of Little AI, each level is implemented as a simple algorithm that takes the player’s command  $c$  in input and returns result  $r$ . Internally, an interaction  $i$  is a tuple  $i = \langle c, r \rangle$ . On the display, an interaction’s shape represents  $c$  and an interaction’s colour represents  $r$ . The values of interactions are hard coded for each level. The 3D simulations are not used “behind the scene” to compute the level but are only launched when the *replay window* is open. In the future, this will allow designing different 3D simulations to illustrate the same level.

## 4. Impact

Little AI is available for free on the Apple app store. It has been downloaded and installed more than a hundred times since July 2016. Our group at the University of Lyon is using it in class for teaching artificial intelligence to master degree students. Georgeon, Barbier-Gondras, & Morgan [8] also proposed it as a pedagogical resource in the *Implementation of Developmental Learning Massive Open Online Course (IDEAL MOOC)*. Players have reported that, “as the difficulty of the game increased, they start to understand the real issues in detecting rewarding patterns of interaction”, and that “they were imagining new kinds of algorithms as they were playing the game”.



**Fig. 1.** Little AI's user interface. Left: Game Screen. Each time the player presses a command, a new interaction appears at the bottom of the trace and the trace is scrolled one step upwards. Interactions have a numerical value displayed on their right in digits and as a bar-graph (green for positive, red for negative). Right: Menu Screen. Level 0 is completed (square button). Level 1 is the next to be played (rounded rectangle button). Levels 2 to 17 are still locked (round buttons). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



**Fig. 2.** Playing Level 0.00. (a) Tap the white square. (b) Tap the white circle. (c) Tap the white circle 9 more times and then tap the Replay button. (d) Watch the robot while tapping the commands.

We do not plan on turning Little AI into a commercial product. Players are only invited to make a donation through in-app-purchase to support the maintenance costs. We are advertising Little AI through various channels to increase its visibility in the scientific community and in the general public.

Levels 0.01 to 0.04 illustrate a task first introduced by Singh, Jaakkola, and Jordan [9] as a simple example of a Partially Observable Markov Decision Process (POMDP). These authors designed an agent that can perform this task using non-Markov reinforcement learning without state estimation. Georgeon, Ritter, and Haynes [10] proposed an alternative approach using a constructivist schema mechanism. Georgeon, Casado, and Matignon [11] examine the differences between reinforcement-learning and constructivist-learning algorithms that can complete this task.

Level 0.05 to 0.10 illustrate the *string problem* first introduced by Georgeon and Hassas [12]. Later, Georgeon, Bernard, and Cordier [13] proposed a new approach to addressing this task with an agent capable of constructing a proto-ontology of the environment from regularities of interaction.

Levels 0.11 to 0.17 illustrate the *small loop problem* first introduced by Georgeon and Ritter [14]. Georgeon, Wolf, and Gay [15] implemented an *e-puck* robot that can play this task. Georgeon and Marshall [16] argued that this task could be used as a test to highlight the emergence of sense-making in artificial agents.

Group 1 illustrates collaborative tasks, such as the *Frog game* introduced by Meyer, Bekkering, Paulus, and Hunnius [17] to assess joint action coordination in young children, and the *Russian elevator* task introduced by Samsonovich, Tolstikhina, and Bortnikov [18] to assess emotional artificial intelligence. Levels in this group are collaborative multi-player levels that can be played in duo over the internet. We are working with Samsonovich and his colleagues to develop more levels and invent a constructivist adaptation of the Russian Elevator task.

Group 2 explores the problem of learning to interact with simple mobile objects. This problem involves constructing and updating a rudimentary spatial memory. We are still testing Levels 2.00 to 2.02, and working on creating more levels in this group with

the goal of defining a generic task to assess the emergence of spatial awareness in artificial agents.

## 5. Conclusions

We believe that Little AI will popularize the constructivist paradigm of AI and constructivist epistemology in general. Little AI follows an incremental scientific methodology consisting of defining tasks that illustrate important theoretical questions, and, in parallel, challenging the scientific community to design agents that can solve these tasks. The methodology is incremental in the sense that new agents should also be able to learn previous tasks, thus leading to designing increasingly intelligent AIs. AI designers may replicate these tasks in their favourite Integrated Development Environment (IDE) that they use to program their AI. In the future, we wish to work with other researchers to create new groups of levels that demonstrate the tasks that they are studying.

## Acknowledgements

We gratefully thank Michel Chourot for his graphic design, Jonathan Morgan for his French to English translations, and Joe Wilson for his “Baby Girl” audio creation. Little AI was developed with the support of the Service for Innovation of Pedagogy (ICAP, NUMEDUC) at Université Claude Bernard Lyon.

## References

- [1] Piaget J. *The construction of reality in the child*. New York: Basic Books; 1937.
- [2] Russell S, Norvig P. *Artificial intelligence, a modern approach*. Englewood Cliffs NJ: Prentice Hall; 2003.
- [3] Riegler A. The radical constructivist dynamics of cognition. In: Wallace B, editor. *The mind, the body and the world: Psychology after cognitivism?* London: Imprint; 2017. p. 91–115.
- [4] Georgeon O, Cordier A. Inverting the interaction cycle to model embodied agents. In: Samsonovich A, Robertson P, editors. *Biologically inspired cognitive architectures*. BICA 2014; proceedings of the 5th annual international conference on biologically inspired cognitive architectures, vol. 41. Elsevier; Procedia Computer Science; 2015. p. 243–8.
- [5] Georgeon O, Boltuc P. Circular constitution of observation in the absence of ontological data. *Construct Found* 2016;12(1):17–9.
- [6] Lightbulb <http://www.ninjadoodle.com/lightbulb> [accessed Jun 28 2017].
- [7] Robocode <http://robocode.sourceforge.net/> [accessed Jun 28 2017].
- [8] Georgeon O, Barbier-Gondras C, Morgan J. Developmental-AI MOOC assessment. In: Khalil M, Ebner M, Kopp M, Lorenz A, Kalz M, editors. *European MOOC summit*. EMOOCs 2016: Proceedings of the 4th European Stakeholder summit on experience and best practices in and around MOOCs; 2016. p. 539–543.
- [9] Singh S, Jaakkola T, Jordan M. Learning without state-estimation in partially observable Markovian decision processes. In: Cohen WW, editor. *Machine learning*. ML 1994; Proceedings of the eleventh international conference on machine learning. Elsevier; 1994. p. 284–92.
- [10] Georgeon O, Ritter F, Haynes S. Modeling bottom-up learning from activity in soar. In: *Behavior representation in modeling and simulation*. BRiMS 2009: proceedings of the 18th annual conference on behavior representation in modeling and simulation. BRiMS Committee; 2009. p. 65–72.
- [11] Georgeon O, Casado R, Matignon L. Modeling biological agents beyond the reinforcement-learning paradigm. In: Georgeon O, editor. *Biologically inspired cognitive architectures*. BICA 2015: Proceedings of the 6th annual international conference on biologically inspired cognitive architectures, vol. 71. Elsevier; Procedia Computer Science; 2015. p. 11–6.
- [12] Georgeon O, Hassas S. Single agents can be constructivist too. *Construct Found* 2013;9(1):40–2.
- [13] Georgeon O, Bernard F, Cordier A. Constructing phenomenal knowledge in an unknown noumenal reality. In: Georgeon O, editor. *Biologically inspired cognitive architectures*. BICA 2015: Proceedings of the 6th annual international conference on biologically inspired cognitive architectures, vol. 71. Elsevier; Procedia Computer Science; 2015. p. 11–6.
- [14] Georgeon O, Ritter F. An intrinsically-motivated schema mechanism to model and simulate emergent cognition. *Cogn Sys Res* 2012;15–16:73–92.
- [15] Georgeon O, Wolf C, Gay S. An enactive approach to autonomous agent and robot learning. In: *Development and learning and epigenetic robotics*. ICDL-EPIROB 2013: Proceedings of the third joint international conference on development and learning and on epigenetic robotics. IEEE; 2013. p. 1–6.
- [16] Georgeon O, Marshall J. Demonstrating sense-making emergence in artificial agents: a method and an example. *Int J Mach Consci* 2013;5(2):131–44.
- [17] Meyer M, Bekkering H, Paulus M, Hunnius S. Joint action coordination in  $\frac{21}{2}$ - and 3-year-old children. *Front Hum Neurosci* 2010;4:220.
- [18] Samsonovich A, Tolstikhina A, Bortnikov P. A test for believable social emotionality in virtual actors. In: Samsonovich A, Klimov V, editors. *Biologically inspired cognitive architectures*. BICA 2016: Proceedings of the 7th annual international conference on biologically inspired cognitive architectures. Elsevier; Procedia Computer Science, 8; 2016. p. 450–8.