

Our Coding Adventure: Using LLMs to Personalise the Narrative of a Tangible Programming Robot for Preschoolers

Martin Ruskov

Department of Languages, Literatures, Cultures and Mediations, University of Milan, Piazza Sant'Alessandro 1, 20123 Milan, Italy

Abstract

Finding balanced ways to employ Large Language Models (LLMs) in education is a challenge due to inherent risks of poor understanding of the technology and of a susceptible audience. This is particularly so with younger children, who are known to have difficulties with pervasive screen time. Working with a tangible programming robot called Cubetto, we propose an approach to benefit from the capabilities of LLMs by employing such models in the preparation of personalised storytelling, necessary for preschool children to get accustomed to the practice of commanding the robot. We engage in action research to develop an early version of a formalised process to rapidly prototype game stories for Cubetto. Our approach has both reproducible results, because it employs open weight models, and is model-agnostic, because we test it with 5 different LLMs. We document on one hand the process, the used materials and prompts, and on the other the learning experience and outcomes. We deem the generation successful for the intended purposes of using the results as a teacher aid. Testing the models on 4 different task scenarios, we encounter issues of consistency and hallucinations and document the corresponding evaluation process and attempts (some successful and some not) to overcome these issues. Importantly, the process does not expose children to LLMs directly. Rather, the technology is used to help teachers easily develop personalised narratives on children's preferred topics. We believe our method is adequate for preschool classes and we are planning to further experiment in real-world educational settings.

Keywords

tangible programming, preschool education, LLM storytelling, open weights models

1. Introduction

Employing Large Language Models (LLMs) in education is not a straightforward task [1]. It is particularly challenging with younger children, who are already known to have difficulties with pervasive screen time [2]. At the same time, exactly in the early stages of education, children are very different and less used to a standardised educational process. They are less familiar to the abstractions necessary to develop early-stage competences in science, technology, engineering and mathematics (STEM). To illustrate this, already a non-trivial question is “Why should I program a robot to go somewhere, when I can put it there myself?” and they are not much appreciative of the need to develop computational thinking [3].

Our contribution is in the context of Cubetto, a physical programming platform: a robot travelling on wheels and a control board with physically-insertable command blocks [4]. Whereas the creator-provided teaching materials accompanying Cubetto aim to teach how to use Cubetto, including physical and creative activities for children [5], they leave it to teachers to elaborate the extensive repetition that would allow children to get accustomed to the robot. We set out from the premise that such getting used to Cubetto and its control is an important prerequisite for its adoption. Thus, we propose an approach to support pre-school teachers in developing the educational storytelling necessary to engage children with the learning activity. This is inspired by the narrative success of AI Dungeon, an interactive platform that engages users with ChatGPT to collaboratively create narrative experiences [6]. Using

D-SAIL Workshop - Transformative Curriculum Design: Digitalisation, Sustainability, and AI Literacy for 21st Century Learning, July 22, 2025, Palermo, Italy

✉ martin.ruskov@unimi.it (M. Ruskov)

ORCID 0000-0001-5337-0636 (M. Ruskov)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

LLMs for such a learning activity that is not overly challenging allows for easier discernment of the desired pedagogical features.

2. Background

Computational thinking has been recognised as a key capability that children need to develop in the 21st century and programming skills are a cornerstone to it. Particularly from the 1980s, the practice of turtle programming has stuck as one of the most common onboarding approaches for younger children [7, 8]. A number of platforms have attempted to make this accessible for even younger audiences by introducing tangible programming toolkits [9, 10, 4]. Regardless whether taught on its own or integrated in other subjects, teaching programming skills helps students develop important competences. In early childhood, the process of teaching programming benefits from enrichment with mixture with storytelling activities as a developmentally appropriate delivery approach. A systematic review from 2022 identified a need for the development of educational curricula integrating teaching of programming and the corresponding teacher-training programs [3]. In a discussion of the barriers to teaching programming to K-12 students, another meta-analysis draws attention to the highly abstract and complex syntax of text-based programming [11]. The authors indicate particularly the lack of student interest in text-based programming as one of the main challenges and suggest there is a need to improve teaching methods in order to foster student motivation and interest [11].

Block-based and visual coding tools have emerged as a way to make programming more accessible to younger audiences that might struggle with the traditional text-based approach [12]. Research indicates that students in elementary schools who learned programming with block-based visual programming tools had better academic achievement than those who learned programming only with traditional text-based programming tools [12, 13]. One of several toys following a similar rationale, Cubetto has been developed as an attempt to take this further [4], making block-based programming accessible also to younger children who are especially susceptible to screen time overuse [2] and the surrounding adults that are increasingly reluctant to have young children engage with digital screens [14]. Documentation and learning materials around Cubetto in particular [4, 5] are rich with examples of different ways to engage with the robot when it comes to teaching programming skills to small children. This includes different physical activities for children, such as imitating and empathising with the robot, story co-creation activities, and creative activities - using drawing, role-play and others.

Research into the motivations to play games [15] suggests 3 broader groups of motivating categories: achievement, immersion and social motivation. These three categories and their 10 components give broader indication of what it takes to gamify a task. Certainly, challenge and achievement are part of these, and in our particular case these are addressed by the nature of navigating a robot. But the immersion component strongly features storytelling, containing topics such as discovery, role-play, customisation and escapism that could find form through narrative. Shared storytelling among the children could also boost the third category with the components of social motivation, containing the components socialising, relationship and teamwork. Finding ways to include any of these motivational factors reinforces the opportunity to engage children. Our approach is inspired by the narrative success of AI Dungeon, an interactive platform that engages users with ChatGPT to collaboratively create narrative experiences [6].

3. Method

Here we aim to develop an early version of a formalised approach using rapid prototyping. To this end we engage in action research studying how to prompt LLMs, similar to the way this was previously done regarding the capabilities of a text-to-image transformer [16]. Such an approach allows us to simultaneously study both the potential of human-AI collaboration for preschool education using current LLMs, and the process of achieving this collaboration. By illustrating the use of five different

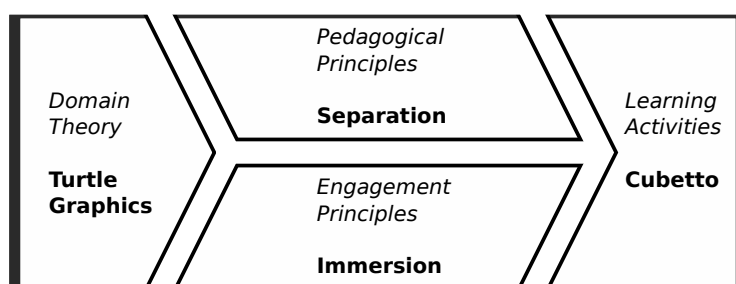


Figure 1: The pedagogical process integrating educational and game-based elements together.

models in parallel, we demonstrate that this process could be independent of the specific LLMs of choice.

To guide the development of the intended formalised approach, we turn to a methodological framework for the design of serious games that we have previously developed. We build a process proposed by Davies and Mangan that starts from theory and goes through pedagogical principles to finally arrive at the design of learning activities [17], which we previously expanded by incorporating both pedagogical and engagement principles in parallel for the development of educational games [18]. In particular, as pedagogical principles we previously proposed phenomenography, or variation theory, which postulates that in order to effectively understand a phenomenon, a learner needs to experience variation around it. Such variation encompasses three fundamental steps: (i) *contrast* where one dimension of the phenomenon is played with, while others are maintained fixed; (ii) *separation* where inversely one dimension is kept fixed; and (iii) *fusion* where the free interplay between dimensions is experienced [19]. On the other hand, we also propose engagement principles, based on Yee’s dimensions of achievement, immersion and social motivation [15]. Within such a framework, we aim to teach basic turtle programming with separation and immersion using Cubetto as illustrated in Figure 1.

3.1. Materials

We propose a learning experience with Cubetto, a travelling box-shaped wooden robot. It follows instructions provided by placing physical command blocks on a control board. The standard Cubetto



Figure 2: The enhanced Cubetto with color markers intended to help children match command bricks to actions.

Table 1

Sample components of a story, provided as parameters to a prompt to propose a theme, a subject and a task to children.

#	narrative world	subjects	task
1	knights and princesses	Barbie dolls	enact a pursuit
2	pirates	Lego figures	find a treasure
3	superheroes	Hot Wheels cars	enact a struggle
4	Wild West	Brio trains	rescue someone

package comes with four commands, but for a preschool audience we use only the three that are about movement (`move forward` - green, `rotate left` - yellow and `rotate right` - red). We also use a slightly enhanced version of Cubetto as visible in Figure 2. In particular, to help children get used to the commands, we use colour markers on top the robot. These markers help associate the commands to the resulting actions.

As a way to address issues of sustained use and reproducibility [20], for narrative generation our study involves exclusively open weights models. In particular, we consider Google’s Gemma, Meta’s Llama, Mistral, AllenAI’s oLMo and Alibaba’s Qwen. We access these using the llama.cpp platform¹ and the GGUF model format. As a way to also ensure accessibility and data privacy, we focus on models that could be run locally on a contemporary laptop without using GPU functionality. These requirements take us to models within the 7-9 billion parameters range and their largest available 3-bit quantisation from HuggingFace².

Our experiments combine Cubetto with popular commonplace toys. On one hand these additional game subjects are meant as vehicles of inspirations and engagement, on the other as a way for children to overcome the challenge of programming being too abstract, particularly relevant for the age of interest for this study [9, 10]. Furthermore, this gave us an opportunity to capture how well LLMs are able to handle practical constraints. To illustrate the approach, we choose four toys, popular among preschool children: Barbie dolls, Lego figures, Hot Wheels cars and Brio trains. Each of these becomes a possible value for a parameter in our prompt as could be seen in the next subsection. A particular challenge is presented by the `Brio trains` subject. These wooden trains are challenging to adapt for two reasons: (i) Cubetto can neither drive along, nor cross the bulky wooden tracks, and (ii) trains follow tracks and thus do not have the affordance of directional movement represented by the `rotate left` and `rotate right` command blocks. Whereas this is intuitive to a person, it might not be to a LLM.

3.2. Prompt template and parameters

For the composition of the prompt, we take inspiration from narrative structuralist traditions and more specifically the works of Vladimir Propp on morphology of folklore [21]. We devise a prompt template to have models generate ideas. In particular we adopt three personalisation parameters: `narrative world`, `subjects` and `task`. The first two are used to set respectively the context and the protagonists so that they could possibly match the children’s current preferences, aspirations and available physical toys. The third one, `task`, serves to set the objective and possible suggestions for it are derived from Propp’s *functions*. We chose ones that we consider to relate to a protagonist, as illustrated in Table 1.

In our experimentation we use the combinations provided in the rows of Table 1, but the approach allows recombining them in any permutation or introducing further variants. We provide these parameters to LLMs through the following prompt:

¹<https://llama-cpp-python.readthedocs.io>

²<http://hf.co/bartowski/gemma-2-9b-it-GGUF>
<http://hf.co/bartowski/Meta-Llama-3.1-8B-Instruct-GGUF>
<http://hf.co/bartowski/Mistral-7B-Instruct-v0.3-GGUF>
<http://hf.co/bartowski/OLMo-2-1124-7B-Instruct-GGUF>
<http://hf.co/bartowski/Qwen2.5-7B-Instruct-GGUF>

“Suggest to a teacher a game with the Cubetto tangible programming toy for preschool children. It should be about {narrative world}, involve {subjects} and children should use Cubetto to {task}. Only three Cubetto command blocks should be used: forward, turn left and turn right. The description should be about half a page long.”

The exact reusable code for this procedure, as well as an overview of the resulting scenarios proposed by LLMs are available in appendix. This, and a full archive of the evolution of the prompts and corresponding generations are available at the online project repository³.

4. Results

Our experiment involved seven rounds of generations to optimise the final results. This count includes only the end-to-end generation where the prompt was combined with all 4 sets of parameters from Table 1.

Generally all tested LLMs work well for this task and provide actionable instructions. Yet due to the issues outlined below, these should not be seen as proof-read and ready to use guides, and rather as creativity prompts for educators instead.

Without being prompted to do so, all models strive to produce structured, self-contained and detailed activity descriptions. These are formatted documents, having length of about one page. The structure of these produced documents is not consistent, even within iterations of the same LLMs. Yet, typical featured sections are scenario topic, objective(s), necessary materials, preparatory setup, instructions or gameplay, claimed learning outcomes, and variations of the proposed activity.

While this type of proposed solutions is generally useful, it leads to a technical problem - generated responses turn out to be larger than the available response token buffer of the LLMs so they get trimmed. On the upside, this typically happens at a point where the suggested task is sufficiently clear for a teacher to interpret despite the missing ending. We attempted to resolve the problem of too long responses by adding to the prompt, either requests for half-page answers, or impose a character limit. Gemma, Llama and oLMo appear to consistently ignore this part of the instructions and still produce responses longer than the buffer limit. Mistral and Qwen however, adapt their responses, providing shorter proposals and overcoming the problem.

Inconsistencies and hallucinations are widespread, yet straightforward to overcome by a teacher. In a commonly recurring example of an inconsistency, LLMs pretend to provide a list of required materials, but this is often incomplete, sometimes mentioning some necessary parts of the Cubetto package, but not all (e.g. interchangeably blocks and board, although one cannot function without the other). A typical hallucination is the proposed use of a Cubetto command block that does not exist. We addressed this by adding an explicit list of permissible command blocks, as detailed in Section 3.1. This appears to have worked well and eliminated command-related hallucinations. However, considering that response length is an issue, LLMs consistently repeated the command-restricting requirement - something that we do not see as beneficial to the final task.

Task 1 (enact a pursuit) and task 3 (enact a struggle) could potentially present a challenge, as they could involve two active subjects, whereas models generally propose solutions involving a single Cubetto robot. All models, but Qwen transform task 1 into a rescue mission, i.e. the target does not move. Similarly, for task 3, only Mistral and Qwen do not explicitly transform the task into a rescue mission. Coming back to the question of how many robots are used, the only ones that propose using more than one are oLMo for task 2 (find a treasure) and Qwen for task 3 (enact a struggle). These proposals involve dividing children in groups with a robot for each, even if these are not confronted in an interaction between groups. Thus, the proposed activities are easily adapted to using a single robot. More broadly speaking, all models but oLMo bring up the suggestion to divide children in teams. This makes unprovoked and unnecessary assumptions about the number of children involved and whether this number is sufficient and reasonable for a separation in teams. When it comes to the particular challenge of the fourth task, Brio trains, Gemma suggests to “Use Cubetto to guide a train through a

³<https://github.com/mapto/our-coding-adventure>

Brio track” which appears as difficult to follow through. Other suggestions might lead to the idea of using the Brio tracks as continuous obstacles forming a canyon, which could be seen as a very creative solution. The other LLMs appear to be able to respond abstractly enough not to underline this particular challenge with providing phrases such as “navigating through the Brio tracks” (Llama), “using Brio train tracks to create a winding path” (Mistral).

5. Discussion and Conclusion

Our approach installs small models locally and we share all the materials necessary for the reproduction and adoption of this approach. However, should such a process be beyond the technical proficiency of teachers, online access to off-the-shelf models is a completely viable alternative using the same or similar prompts. The method does not disclose any personal information, so there should not be any privacy concerns. However, a subsequent research phase needs to engage with a wider group of children and with pre-school teachers that are independent from the process developers. This would provide evidence about the usefulness of the approach in-the-wild and about perceptions of teachers as direct target audience, and of children as indirect audience. However, the presentation of this method needs to come in the context of the wider learning context of Cubetto, including other introductory activities to the robot [4, 5].

We insist it is important to emphasise that so far there are no reasons or justifications in favour of exposing children directly to output generated by LLMs. Instead, as we propose here, the technology could be used to help overloaded teachers with rather trivial narratives that would put together elements that these same teachers consider useful to integrate in the learning process. Thus, we pursue a two-fold goal: on one hand to scaffold teacher creativity, on the other to not demand from them engagement with the activity that is beyond basic reasonable interaction.

In the process we have encountered cases where LLMs refuse to respond to tasks due to concerns about violence or discrimination. These were rare and circumvented by slight variations in the prompt. However, building on the aforementioned point of no direct exposure of children to LLM output, we believe the premise that censoring responses as a way to protect children is misplaced in educational contexts like ours. It is not up to the model or its authors to decide whether content is appropriate, nor muting conversations is an adequate way of addressing historical controversies. At least for the context in our consideration, we believe that it is teachers that need to take the lead on such decisions. This corresponds to previous research reporting that teachers need to be more involved in educational uses of LLMs [1].

While definitely a topic of interest, due to the adopted method of this study, our results could not be used as a means of systematic comparison between LLMs. Yet, one cannot help but notice the emergence of consistently different behaviour by different models. Being different solutions to the same next token generation task, the differences emerge with the exact architectures and the exact training corpora and training procedures. While the research on optimal architectures is ongoing, there is less possibility to study how differences could be traced back to corpora or training procedures. Among the LLMs studied here, only the creators of oLMo are transparent about both the training process used and put an effort to make the training corpus publicly available [22]. Nevertheless, this one example opens up to the theoretical possibility of studying how responses relate to specific instances in the training data.

Declaration on Generative AI

The author has not employed any Generative AI tools for the production this article beyond the reported experiments.

References

- [1] E. Harvey, A. Koenecke, R. F. Kizilcec, "Don't Forget the Teachers": Towards an Educator-Centered Understanding of Harms from Large Language Models in Education, in: *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, ACM, Yokohama Japan, 2025, pp. 1–19. doi:10.1145/3706598.3713210.
- [2] C. Wang, H. Qian, H. Li, D. Wu, The status quo, contributors, consequences and models of digital overuse/problematic use in preschoolers: A scoping review, *Frontiers in Psychology* 14 (2023). doi:10.3389/fpsyg.2023.1049102.
- [3] E. Macrides, O. Miliou, C. Angeli, Programming in early childhood education: A systematic review, *Int. J. of Child-Computer Interaction* 32 (2022). doi:10.1016/j.ijcci.2021.100396.
- [4] Solid Labs, Beginning computer programming for kids: An introductory guide to computational thinking and coding for kids aged 3-6 years old, PRIMO Toys, 2017. URL: https://www.primotoys.com/wp-content/uploads/2017/09/Ebook-PrimoToys_final-1.pdf.
- [5] Solid Labs, Coding with Cubetto - Unit 1, Reception, Ages 4 to 5, UK National Curriculum, PRIMO Toys, 2017. URL: <https://primotoys.com/education/resources/>.
- [6] M. Hua, R. Raley, Playing With Unicorns: AI Dungeon and Citizen NLP, *Digital Humanities Quarterly* 14 (2020). URL: <https://www.proquest.com/docview/2553526112>.
- [7] S. Grover, R. Pea, Computational Thinking in K-12: A Review of the State of the Field, *Educational Researcher* 42 (2013) 38–43. doi:10.3102/0013189X12463051.
- [8] T. S. McNerney, From turtles to Tangible Programming Bricks: explorations in physical language design, *Personal and Ubiquitous Computing* 8 (2004) 326–337. doi:10.1007/s00779-004-0295-6.
- [9] L. D. Rodić, A. Granić, Tangible interfaces in early years' education: a systematic review, *Personal and Ubiquitous Computing* 26 (2022) 39–77. doi:10.1007/s00779-021-01556-x.
- [10] S. Papavaslopoulou, M. N. Giannakos, L. Jaccheri, Reviewing the affordances of tangible programming languages: Implications for design and practice, in: *2017 IEEE Global Engineering Education Conference (EDUCON)*, 2017, pp. 1811–1816. doi:10.1109/EDUCON.2017.7943096.
- [11] L. Sun, L. Zhou, Does text-based programming improve K-12 students' CT skills? Evidence from a meta-analysis and synthesis of qualitative data in educational contexts, *Thinking Skills and Creativity* 49 (2023) 101340. doi:10.1016/j.tsc.2023.101340.
- [12] Y. Hu, C.-H. Chen, C.-Y. Su, Exploring the Effectiveness and Moderators of Block-Based Visual Programming on Student Learning: A Meta-Analysis, *Journal of Educational Computing Research* 58 (2021) 1467–1493. doi:10.1177/0735633120945935.
- [13] L. Hu, Programming and 21st century skill development in K-12 schools: A multidimensional meta-analysis, *J. of Computer Assisted Learning* 40 (2024) 610–636. doi:10.1111/jcal.12904.
- [14] A. Gentleman, 'the crux of all evil': what happened to the first city that tried to ban smartphones for under-14s?, *The Guardian* (2025). URL: <https://www.theguardian.com/technology/2025/may/07/the-crux-of-all-evil-what-happened-to-the-first-city-that-tried-to-ban-smartphones-for-under-14s>.
- [15] N. Yee, Motivations of Play in MMORPGs, in: *DiGRA 2005 Conference*, 2005, p. 46. URL: <http://www.nickyee.com/daedalus/motivations.pdf>.
- [16] M. Ruskov, Grimm in Wonderland: Prompt Engineering with Midjourney to Illustrate Fairytales, in: B. Alessia, F. Alex, F. Stefano, M. Stefano, R. Domenico (Eds.), *Proceedings of the 19th Conference on Information and Research Science Connecting to Digital and Library Science*, volume 3365 of *CEUR Workshop Proceedings*, 2023, pp. 180–191. URL: <https://ceur-ws.org/Vol-3365/#paper6>.
- [17] P. I. Davies, J. Mangan, Embedding Threshold Concepts: from theory to pedagogical principles to learning activities, in: R. Land, J. H. Meyer, J. Smith (Eds.), *Threshold Concepts within the Disciplines*, Brill, Rotterdam, 2008, pp. 37–50. doi:10.1163/9789460911477_004.
- [18] M. Ruskov, Employing Variation in the Object of Learning for the Design-based Development of Serious Games that Support Learning of Conditional Knowledge, PhD Thesis, University College London, 2014. URL: <https://discovery.ucl.ac.uk/id/eprint/1457529/>.
- [19] F. Marton, *Necessary Conditions of Learning*, Taylor and Francis, Hoboken, 2014.
- [20] L. Chen, M. Zaharia, J. Zou, How Is ChatGPT's Behavior Changing Over Time?, *Harvard Data*

Science Review 6 (2024). doi:10.1162/99608f92.5317da47.

[21] V. Propp, *Morphology of the Folktale*, 2 ed., University of Texas Press, 1968.

[22] P. Walsh, L. Soldaini, D. Groeneveld, K. Lo, S. Arora, A. Bhagia, Y. Gu, S. Huang, M. Jordan, N. Lambert, et al., *2 OLMo 2 Furious*, 2025. doi:10.48550/arXiv.2501.00656.

A. Implementation

The locally executable code that was used for the analysis is provided below.

A.1. Requirements

llama-cpp-python==0.3.9

huggingface-hub==0.31.4

A.2. Script

```
from llama_cpp import Llama

models = [
    {"repo": "bartowski/gemma-2-9b-it-GGUF", "file": "gemma-2-9b-it-Q3_K_XL.gguf"},
    {"repo": "bartowski/OLMo-2-1124-7B-Instruct-GGUF", "file": "OLMo-2-1124-7B-Instruct-Q3_K_XL.gguf"},
    {"repo": "bartowski/Mistral-7B-Instruct-v0.3-GGUF", "file": "Mistral-7B-Instruct-v0.3-Q3_K_L.gguf"},
    {"repo": "bartowski/Qwen2.5-7B-Instruct-GGUF", "file": "Qwen2.5-7B-Instruct-Q3_K_XL.gguf"},
    {"repo": "bartowski/Meta-Llama-3.1-8B-Instruct-GGUF", "file": "Meta-Llama-3.1-8B-Instruct-Q3_K_XL.gguf"},
]

params = [
    {"world": "pirates", "objects": "Lego figures", "task": "find a treasure"},
    {"world": "Wild West", "objects": "Brio trains", "task": "rescue someone"},
    {"world": "superheroes", "objects": "Hot Wheels cars", "task": "enact a struggle"},
    {"world": "knights and princesses", "objects": "Barbie dolls", "task": "enact a pursuit"},
]

prompt_tmpl = """Provide a description of a game with the Cubetto tangible programming toy for preschool
children. It should be about {world}, involve {objects} and children should use Cubetto to {task}. Only
three Cubetto command blocks should be used: forward, turn left and turn right."""

for m in models:
    mname = m["repo"].split("/")[1].split("-")[0]
    print(mname)

    llm = Llama.from_pretrained(
        repo_id=m["repo"],
        filename=m["file"],
        n_gpu_layers=-1,
        flash_attn=True,
    )

    for p in params:
        print(p)
        prompt = prompt_tmpl.format(**p)
        msg = [{"role": "user", "content": prompt}]

        output = llm.create_chat_completion(messages=msg, temperature=0)
        result = output["choices"][0]["message"]["content"]

        fname = f'results/{mname}-{p["world"]}-{p["objects"]}-{p["task"]}.txt'
        with open(fname, "w") as fout:
            fout.write(f'{m["repo"].split("/")[-1]}\n{prompt}\n\n-----\n{result}')
```

B. Outputs

On the next page, the final produced output per model and per task, the last two being indicated in the first paragraph of each minipage. This seen in better detail, and documentation of previous iterations is available at the project repository⁴.

⁴<https://github.com/mapto/our-coding-adventure>

game: 2-10-4-G2JF-Suggest to a teacher a game with the Cabot's tangible programming toy for preschool children. It should be about knights and princesses, involve Barthe dolls and children should use Cabots to meet a person. Only three Cabots command blocks should be used: forward, turn left and turn right. The description should be about half a page long.

The Knight's Quest: A Cabots Adventure

This game combines surprising, imaginative play, and basic programming concepts using Cabots. It's perfect for preschoolers who are just starting to grasp the idea of giving instructions to bring them closer to life.

Story:

Once upon a time, a brave knight set out on a quest to rescue a beautiful princess from a fearsome dragon. The princess is trapped in a tall tower, guarded by the dragon. The knight must navigate a maze to reach the tower and save the princess.

Materials:

- Cabots robot
- Cabots command blocks (forward, turn left, turn right)
- Barthe dolls: a knight and a princess
- A large play mat or map paper

- Optional: building blocks, pillows, or other objects to create obstacles and a "maze"

Instructions:

- Set the Scene:** Create a "maze" on the play mat using building blocks, pillows, or other objects. Place the princess doll at the top of the "maze" and the dragon doll somewhere nearby.
- Introduce the Knight:** Tell the children about the brave knight who is going to rescue the princess. Show them the knight doll and explain that they will be using Cabots to guide the knight through the maze.
- Programming the Knight:**
 - Forward:** Explain that the "forward" block will make Cabots move forward one step.
 - Turn Left/Right:** Explain that the "left" and "right" blocks will make Cabots change direction.
 - Challenge:** Have the children work together to program Cabots to navigate the maze and reach the princess. Encourage them to experiment with different combinations of command blocks.
- The Princess:** Once Cabots reach the princess, have the children celebrate the knight's success!

Variations:

- Obstacle Course:** Add more obstacles to the maze to make it more challenging.
- Dragon Chase:** Have the children program Cabots to avoid the dragon by turning left or right where it is.
- "Milestone"

Meta-Lesson 3-18B-Interact-G2JF-Suggest to a teacher a game with the Cabot's tangible programming toy for preschool children. It should be about knights and princesses, involve Barthe dolls and children should use Cabots to meet a person. Only three Cabots command blocks should be used: forward, turn left and turn right. The description should be about half a page long.

Game Title: "The Brave Knight's Rescue"

Objective: Help young children explore the basics of programming concepts using Cabots, where preschool children can use their problem-solving skills to help a brave knight rescue a princess from a dragon's lair.

Materials:

- Cabots programming toy
- Barthe dolls (princess and knight)
- A simple obstacle course or a designed play area
- A small toy dragon (optional)

Game Description:

In this exciting game, the children will take on the role of a brave knight who must rescue a princess from a dragon's lair. The princess will be represented by a Barthe doll, and the knight will also be a Barthe doll. The children will use a designated play area or an obstacle course.

Goal:

- The children will work by placing the princess Barthe doll at the dragon's lair, and the knight Barthe doll at the entrance of the lair.
- The children will use the Cabots programming toy to create a sequence of commands to help the knight escape through the lair and reach the princess.
- The three Cabots command blocks that will be used are:
 - Forward: to move the knight one step closer to the princess
 - Turn left: to change direction and avoid obstacles
 - Turn right: to change direction and reach the princess
- The children will need to use their problem-solving skills to create a sequence of commands that will help the knight reach the princess without getting stuck or missing any obstacles.
- As the knight reaches each step of the commands, they can use the Cabots to test their solution and see if the knight can reach the princess.
- If the knight successfully reaches the princess, the children can celebrate their success and try to create a new sequence of commands to make the knight's journey even more challenging.

Tip and Variations:

- To make the game more challenging, you can add obstacles such as small blocks or toys that the knight

game: 2-10-4-G2JF-Suggest to a teacher a game with the Cabot's tangible programming toy for preschool children. It should be about knights and princesses, involve Barthe dolls and children should use Cabots to meet a person. Only three Cabots command blocks should be used: forward, turn left and turn right. The description should be about half a page long.

Alley: Maresse's Treasure Hunt with Cabots!

Objective: Guide your game area to find the hidden treasure using Cabots!

- Cabots robot
- Cabots command blocks (forward, turn left, turn right)
- Large paper figures
- Large pieces to build a simple obstacle course (optional)
- Treasure chest (can be a small box, decorated like a treasure chest)

Preparation:

- Set up a play area with a designated starting point for the game and a hidden treasure chest.
- (Optional) Build a simple obstacle course using large blocks to add an extra challenge.

Gameplay:

- Storyboarding:** Help by telling the children a story about a brave treasure map and a brave horse searching for it.
- Programming:** Explain to the children that they will be programming Cabots, the game pieces, to navigate the course and find the treasure.
- Command Blocks:** Introduce the three command blocks: forward, turn left, and turn right. Demonstrate how each block moves Cabots to a specific direction.
- Treasure Hunt:** Each child will take turn programming Cabots to find the treasure. They can use the command blocks to create a sequence of directions that lead Cabots to the treasure chest.
- Challenge:** Encourage them to think about the directions they need to move and the turns they need to make.
- Calibration:** Children can work together to program Cabots, sharing ideas and helping each other.
- Multiple Treasures:** Hide multiple treasure chests with different rewards.
- Prize Names:** Have the children name their large paper figures and create a story around them.

Learning Outcomes:

- Problem-solving:** Children learn to think logically and solve problems by creating a sequence of commands to guide Cabots.
- Spatial reasoning:** Children will develop their understanding of direction and spatial relationships.
- Collaboration:** Children will learn to work together and share.

Meta-Lesson 3-18B-Interact-G2JF-Suggest to a teacher a game with the Cabot's tangible programming toy for preschool children. It should be about knights and princesses, involve Barthe dolls and children should use Cabots to find a treasure. Only three Cabots command blocks should be used: forward, turn left and turn right. The description should be about half a page long.

Treasure Hunt Adventure with Cabots

Objective: Help young children explore the basics of programming concepts using Cabots.

Materials:

- Cabots programming toy
- Large paper figures
- A treasure chest (or a small box)
- A game-themed obstacle course (e.g., a small island, a bridge, or a cave)

Instructions:

- Set up the game-themed obstacle course in the classroom or play area. Place the treasure chest at the end of the course.
- Introduce the Cabots programming toy and explain to the children will use it to help the pirates find the treasure.
- Demonstrate how to use the Cabots command blocks: forward, turn left, and turn right. Explain that the children will need to program the Cabots to navigate through the obstacle course to reach the treasure.
- Divide the children into small groups and give each group a Cabots programming figure. Explain that they will need to work together to program the Cabots to find the treasure.
- Challenge the children to program the Cabots to reach the treasure using only the three command blocks: forward, turn left, and turn right. Encourage them to experiment with different combinations of commands.
- As the children program the Cabots, ask them to describe their plans and explain why they chose a particular sequence of commands.
- Once the Cabots reach the treasure, congratulate the children on their success and ask them to reflect on what they learned about programming and problem-solving.

Variations:

- To make it more challenging, add obstacles or "puzzle traps" that the Cabots need to avoid.
- Encourage the children to create their own game-themed obstacle course and challenge their peers to find the treasure.
- Use the Cabots to create a game-themed drama, where the children take turns programming the Cabots to navigate through a series of challenges and obstacles.
- Observe the children's ability to use the Cabots command blocks to solve a problem

Meta-Lesson 3-18B-Interact-G2JF-Suggest to a teacher a game with the Cabot's tangible programming toy for preschool children. It should be about knights and princesses, involve Barthe dolls and children should use Cabots to find a treasure. Only three Cabots command blocks should be used: forward, turn left and turn right. The description should be about half a page long.

Title: Pirate Treasure Hunt with Cabots and Hat Whisks

This is an engaging and interactive game. Preschool children will use the Cabots tangible programming toy to create a sequence of commands to help the pirates find the treasure.

Materials: Cabots programming robot, Cabots block set (forward, turn left, turn right), Large paper figures and accessories, Large island landscape including a treasure chest.

Instructions: 1. Set up a large island landscape, complete with a treasure chest hidden somewhere on the island. 2. Place the Cabots programming robot at the starting point on the island. 3. Introduce the children to the Cabots block set. 4. Divide the children into teams and assign each team with a set of Cabots programming figures and accessories. 5. Instruct the children to program Cabots to navigate the island, find the treasure, and bring it back to their starting point. 6. Encourage teamwork, problem-solving, and critical thinking as the children work together to solve the treasure hunt challenge. 7. The team that successfully finds the treasure and brings it back to their starting point first wins the game.

This Pirate Treasure Hunt game combines fun with problem-solving skills with the educational benefits of programming, teamwork, collaboration, and problem-solving abilities in preschool children.

game: 2-10-4-G2JF-Suggest to a teacher a game with the Cabot's tangible programming toy for preschool children. It should be about knights and princesses, involve Barthe dolls and children should use Cabots to meet a person. Only three Cabots command blocks should be used: forward, turn left and turn right. The description should be about half a page long.

Superhero Training: The Hat Whisks Rescue Mission!

This game combines Cabots's tangible programming with imaginative play, encouraging preschoolers to think quickly and creatively while exploring the world of superheroes.

Materials:

- Cabots robot
- Cabots command blocks (forward, turn left, turn right)
- Barthe dolls: a superhero and a villain
- A play mat or a large sheet of paper to represent the city

Story:

Superheroes must rescue a citizen from the clutches of a villain! The villain has trapped the citizen in a building (represented by a box or a drawn shape on the play mat). The superheroes must navigate the city, avoiding obstacles (represented by toys or drawn shapes), and reach the building to save the day!

Goal:

- Set the scene:** Lay out the play mat and arrange the Hat Whisks cars. The superheroes can start at a designated point, while the villain can be placed near the "building".
- Program the superheroes:** Children work together to program Cabots to guide the superheroes to the villain. They can use the forward, turn left, and turn right command blocks to create a path for the superheroes. Encourage them to think about the direction they need to go and how many steps or turns are needed to reach the villain.
- Obstacle course:** Add obstacles to the play mat, such as toys or blocks that move them. Children can adjust Cabots's program to navigate around these obstacles, making the mission more challenging.
- The rescue:** Once Cabots reaches the "building", the superhero can "rescue" the citizen. Celebrate the successful mission!

Variations:

- Multiple rescues:** Use different Hat Whisks cars to represent different villains with unique abilities.
- Multiple rescues:** Create multiple buildings with trapped citizens and have children program Cabots to rescue them.
- Teamwork:** Divide the children into teams, each responsible for programming a different superhero to complete a specific rescue mission.

Learning Outcomes:

- Spatial reasoning:**

Meta-Lesson 3-18B-Interact-G2JF-Suggest to a teacher a game with the Cabot's tangible programming toy for preschool children. It should be about knights and princesses, involve Barthe dolls and children should use Cabots to meet a person. Only three Cabots command blocks should be used: forward, turn left and turn right. The description should be about half a page long.

Superhero Rescue Mission: Cabots and Hat Whisks

Objective: Help young children explore the basics of programming concepts using Cabots.

In this exciting game, children will become superheroes tasked with rescuing their friends from the clutches of the evil villain, "The Blackbeard". The Blackbeard has set up a series of obstacles using Hat Whisks cars and blocks, and it's up to our young superheroes to navigate through the challenges and rescue their friends.

Goal:

- Set up the game area with a series of Hat Whisks cars and blocks to create a challenging obstacle course.
- Introduce the children to the Cabots programming toy and explain the basics of forward, turn left, and turn right.
- Each team will receive a Cabots robot and a set of command blocks (forward, turn left, and turn right).
- Children will work together to program Cabots to navigate through the obstacle course and reach their friends, who are trapped behind the Hat Whisks cars.
- However, The Blackbeard has set up a series of obstacles to hinder the superheroes' progress. For example, a Hat Whisker might be placed in a way that blocks the path or a block might be used to block the Cabots's path.
- The superheroes must use their problem-solving skills and critical thinking to program their Cabots to overcome these challenges.
- The team that successfully rescues their friends and reaches the final destination wins.

Cabots Command Blocks:

- Forward:** Move the Cabots forward one step.
- Turn Left:** Turn the Cabots left by 90 degrees.
- Turn Right:** Turn the Cabots right by 90 degrees.

Tip and Variations:

- To make it more challenging, you can add more Hat Whisks cars and blocks to the obstacle course.
- You can also add time limits or set up a series of challenges that the superheroes must complete before reaching their friends.
- Use

Meta-Lesson 3-18B-Interact-G2JF-Suggest to a teacher a game with the Cabot's tangible programming toy for preschool children. It should be about knights and princesses, involve Barthe dolls and children should use Cabots to meet a person. Only three Cabots command blocks should be used: forward, turn left and turn right. The description should be about half a page long.

Title: Superhero Car Chase with Cabots and Hat Whisks

Objective: In this engaging and interactive game, preschool children will use the Cabots tangible programming toy to create a sequence of commands to help the superheroes chase the villain.

Materials: Cabots programming robot, Cabots block set (forward, turn left, turn right), Hat Whisks cars and tracks, Superhero figurines.

Instructions: 1. Set up a Hat Whisk track with various turns, and obstacles to create an exciting car chase. 2. Place the Hat Whisks cars on the track, and assign each team a superhero figurine to represent the villain. 3. Explain to the children that the superheroes must use the Cabots programming blocks to control the superhero's vehicle and help them escape from the villain's car.

Goal: The team that successfully escapes the villain's car and reaches the finish line wins the game.

Tip and Variations: 1. Place the Cabots robot at the starting point of the Hat Whisk track. 2. Children will use the forward block to make Cabots move along the track. 3. The superheroes must use the turn left and turn right blocks to navigate the track. 4. The objective is for the superheroes to reach the finish line before the villain's car catches up. 5. Encourage children to strategize and use their problem-solving skills to create the most efficient path for their superheroes. 6. Allow each team to discuss the strategies and solutions before they start the race.

This game not only teaches children's creativity and critical thinking but also helps them understand the basics of programming and problem-solving in a fun and engaging way. By incorporating superheroes and Hat Whisks, the game appeals to children's natural sense of adventure and competition, making learning a joyful experience.

OLM6-1124-78-Interact-G2JF-Suggest to a teacher a game with the Cabot's tangible programming toy for preschool children. It should be about knights and princesses, involve Barthe dolls and children should use Cabots to meet a person. Only three Cabots command blocks should be used: forward, turn left and turn right. The description should be about half a page long.

Title: Superhero Adventure: Cabots the Battle for the Treasure

Objective: In this engaging game, preschool children will embark on a thrilling adventure with their Cabots. The game involves using Cabots to navigate through a series of challenges, including a maze, a river, and a mountain range.

Materials: Cabots programming robot, Cabots block set (forward, turn left, turn right), Large paper figures and accessories, Large island landscape including a treasure chest.

Instructions: 1. Set up a large island landscape, complete with a treasure chest, a small river, and a mountain range. 2. Place the Cabots programming robot at the starting point on the island. 3. Introduce the children to the Cabots block set. 4. Divide the children into teams and assign each team with a set of Cabots programming figures and accessories. 5. Instruct the children to program Cabots to navigate the island, find the treasure, and bring it back to their starting point. 6. Encourage teamwork, problem-solving, and critical thinking as the children work together to solve the treasure hunt challenge. 7. The team that successfully finds the treasure and brings it back to their starting point first wins the game.

This game not only teaches children's creativity and critical thinking but also helps them understand the basics of programming and problem-solving in a fun and engaging way. By incorporating superheroes and Hat Whisks, the game appeals to children's natural sense of adventure and competition, making learning a joyful experience.

OLM6-1124-78-Interact-G2JF-Suggest to a teacher a game with the Cabot's tangible programming toy for preschool children. It should be about knights and princesses, involve Barthe dolls and children should use Cabots to meet a person. Only three Cabots command blocks should be used: forward, turn left and turn right. The description should be about half a page long.

Title: The Enchanted Quest: Knights and Princesses

Objective: The objective of this game for preschool children is to use the Cabots tangible programming toy to create a sequence of commands to help the knight rescue the princess. The game involves using Cabots to navigate through a series of challenges, including a maze, a river, and a mountain range.

Materials: Cabots programming robot, Cabots block set (forward, turn left, turn right), Large paper figures and accessories, Large island landscape including a treasure chest.

Instructions: 1. Set up a large island landscape, complete with a treasure chest, a small river, and a mountain range. 2. Place the Cabots programming robot at the starting point on the island. 3. Introduce the children to the Cabots block set. 4. Divide the children into teams and assign each team with a set of Cabots programming figures and accessories. 5. Instruct the children to program Cabots to navigate the island, find the treasure, and bring it back to their starting point. 6. Encourage teamwork, problem-solving, and critical thinking as the children work together to solve the treasure hunt challenge. 7. The team that successfully finds the treasure and brings it back to their starting point first wins the game.

This game not only teaches children's creativity and critical thinking but also helps them understand the basics of programming and problem-solving in a fun and engaging way. By incorporating superheroes and Hat Whisks, the game appeals to children's natural sense of adventure and competition, making learning a joyful experience.

OLM6-1124-78-Interact-G2JF-Suggest to a teacher a game with the Cabot's tangible programming toy for preschool children. It should be about knights and princesses, involve Barthe dolls and children should use Cabots to meet a person. Only three Cabots command blocks should be used: forward, turn left and turn right. The description should be about half a page long.

Title: The Enchanted Quest: Knights and Princesses

Objective: The objective of this game for preschool children is to use the Cabots tangible programming toy to create a sequence of commands to help the knight rescue the princess. The game involves using Cabots to navigate through a series of challenges, including a maze, a river, and a mountain range.

Materials: Cabots programming robot, Cabots block set (forward, turn left, turn right), Large paper figures and accessories, Large island landscape including a treasure chest.

Instructions: 1. Set up a large island landscape, complete with a treasure chest, a small river, and a mountain range. 2. Place the Cabots programming robot at the starting point on the island. 3. Introduce the children to the Cabots block set. 4. Divide the children into teams and assign each team with a set of Cabots programming figures and accessories. 5. Instruct the children to program Cabots to navigate the island, find the treasure, and bring it back to their starting point. 6. Encourage teamwork, problem-solving, and critical thinking as the children work together to solve the treasure hunt challenge. 7. The team that successfully finds the treasure and brings it back to their starting point first wins the game.