Artificial Intelligence, Games, and Education

Barbero, G.

Chapter 4

Scientific Education, Video Games, and Artificial Intelligence

In this chapter, we bring together previous information and define the current developments in:

- Generative AI in programming education
- Video Games for Generative AI in education

The goal of this chapter is to discuss and speculate on the impact of generative AI in education by exploring current literature. In the first section, we derive pitfalls but also opportunities for the future. With the perspective that generative AI is here to stay, we propose a direction to limit its negative impact on education while highlighting its potential to motivate and aid students' learning. Spoiler alert: it's games. In the second section, we embark on a simulated design process that has as a fictional goal the one of creating an LLM-powered NPC to teach Roman history in game settings. In the process, we reflect on our design considerations and generalise them for future developers. We also evaluate the final product and draw conclusions on the feasibility of incorporating generative AI (in particular, LLMs) in game-based learning. Each section is connected to the rest of the thesis by a short summary of the findings.

4.1 Generative AI and Programming Education: Considerations from Current Studies

Since the release of GPT in 2022, generative AI has quickly become ubiquitous, being utilised in different human activities. As is often the case, the advancement of disruptive technologies fosters new discussions within old contexts; think about the legal considerations around AI-generated content [204] or the studies about potential applications of generative AI in education [205]. With regard to the latter, many studies highlight the risks of using generative AI in computer science education. For example, with the assistance of GPT models, students are able to complete assignments more quickly, but they also retain less information compared to their peers who worked without AI help [30, 206]. However, other studies in similar settings revealed that students' computational thinking skills improved using generative AI [207]. This section aims to compare selected research in the field in order to clarify the impact of generative AI in programming education. We define the following research questions:

- How is the impact of AI-powered interventions measured?
- What are the results of these interventions?
- How can we reconcile apparently opposite results in the field?
- What can we learn for future empirical research involving generative AI in computer science education?

4.1.1 Literature Review

Our understanding of generative AI's impact on programming education remains limited. A recent literature review about empirical research in the field includes only thirty-seven studies [208]. Of these, only two evaluate students' computational thinking and programming skills. In fact, the majority of the studies are limited to how LLMs perform in terms of programming. These include skills such as debugging [209], pair programming [210] or program generation [211]. However, these skills do not necessarily reflect the impact of generative AI on learning; they depict it as a useful tool for programmers. This seems to be a common misconception in current research in the field, where generative AI performance in teaching environments is evaluated based on its programming performance [208]. Going back to the aforementioned literature review [208], the two studies focusing on students' computational thinking

and programming skills development present very different methodologies. The first one highlights the positive effects on both metrics for students using generative AI. However, the measurement of computational thinking is quite complex and heavily dependent on the model used. In the case of this experiment, computational thinking skills are measured using a scale based on a more abstract model [212]. This includes skills that are not necessarily exclusive or focused on programming: creativity, algorithmic thinking, problem solving, critical thinking, cooperativity and communication skills. However, the most critical flaw of the study is the testing methodology for programming skills development, measured using a self-efficacy scale focused on students' confidence in tackling abstract programming problems [213](see figure 4.1). We argue that, with these tools, conclusively evaluating the actual impact of generative AI on students' programming learning is impossible.

Item No.	Item Description (alpha reliability estimates for factors in parentheses)
	Factor 1: Independence and persistence (alpha = .94)
23	Complete a programming project if I had a lot of time to complete the program.
22	Complete a programming project once someone else helped me get started.
21	Complete a programming project if I could call someone for help if I got stuck.
24	Complete a programming project if I had just the built-in help facility for assistance.
20	Complete a programming project if I had only the language reference manual for help.
19	Complete a programming project if someone showed me how to solve the problem first.
25	Find ways of overcoming the problem if I got stuck at a point while working on a programming project.
17	Debug (correct all the errors) a long and complex program that I had written, and make it work.
	Factor 2: Complex programming tasks (alpha = .94)
13	Understand the object-oriented paradigm.
16	Make use of a class that is already defined, given a clearly labeled declaration of the class.
14	Identify the objects in the problem domain and declare, define, and use them.
8	Build my own C++ libraries.
18	Comprehend a long, complex multi-file program.
12	Organize and design my program in a modular manner.
32	Write a program that someone else could comprehend and add features to at a later date.
15	Make use of a pre-written function, given a clearly labeled declaration of the function.
11	Write a long and complex C++ program to solve any given problem as long as the specifications are clearly defined.
28	Mentally trace through the execution of a long, complex, multi-file program given to me.
29	Rewrite lengthy confusing portions of code to be more readable and clear.

Figure 4.1: Example of statements used for students' self-efficacy evaluation in [213].

The second study has a completely different approach. In this case, generative AI has been implemented in a gamified interface [214]. The article presents the experiment as ongoing and, therefore, actual data about students' performance and improvement

is not available and necessitates further studies. The author only reports positive effects on motivation and acceptance that are typical of a gamified environment (as discussed in chapter 3). However, the interesting element in the study is the innate limitations to generative AI that a gamified environment can present. Both studies claim to address computational thinking and programming skills, but their measurements lack focus on these topics. Studies measuring actual student performance have only recently emerged. For example, a recent experiment in a programming class of Fortran indicates that retention is superior for students who do not use generative AI [206]. In the experimental setup, the experimental groups are allowed a quite free use of various modern generative models. The control group is allowed to use Google exclusively. However, other studies take a different perspective, evaluating generative AI in its ability to tutor students. Current research highlights the potential for LLMs to perform almost as well as human tutors [215]. Obviously, evaluation of tutoring is quite complex, and it often relies on experts. Moreover, in order to make results comparable, many studies limit what students can ask, for example, using preselected prompts [215]. In general, most studies report positive results on students' acceptance and motivation using generative AI [216]. However, others highlight a negative correlation between perceived ease of use and perceived usefulness [217].

4.1.2 Discussion

In this chapter, we use the considerations from the above literature review in order to provide answers to the research questions. It is important to note that our work is far from systematic. It is, in fact, based on a selective exploration of salient studies in the field. Arguably, the low number of empirical studies specifically focusing on programming students' retention and skills development makes systematic approaches inadequate.

How is the impact of AI-powered interventions measured?

Current research primarily measures the acceptance of generative AI among students and teachers. This is often performed through the typical technology acceptance model, a well-studied and validated tool for measurement. On the other hand, other aspects of the impact of generative AI on programming education are more difficult to measure. We argue that metrics and research goals are not always well aligned. In the case of computational thinking skills development, different models can be used as references. However, each model has a different perspective on these skills and select-

ing the best-suited one is a necessary evaluation. As for programming skills, teachers already have many tools to test students' learning. Self-assessment tools have their own reason and space; however, they tend to be more strongly related to the respondent's confidence than their actual development. Confidence is particularly reinforced with the ability to perform a certain job, something that, especially in introductory programming curricula, generative AI can certainly provide. However, we argue that this is not necessarily related to students' proficiency in programming but with their perceived capacity to pass the assignments provided. Other studies strongly focus on human comparison. In these cases, measurements are usually performed by humans who evaluate generative AI's performance compared to other human experts. It is the case of experiments centred around the effect of AI-powered tutoring. Another specific characteristic of this format is that it often relies, by necessity, on predetermined prompts or other forms of limitations that make human and AI tutoring comparable.

What are the results of these interventions?

Empirical research shows a clear improvement in students' motivation. This is definitely a relevant effect, probably related to the ease of use of generative AI and the enthusiasm emerging from a new, and quite frankly impressive, technology. Results emerging from technology acceptance studies (using variations of the technology acceptance model [218]) are also extremely encouraging, especially for younger participants. As mentioned above, many studies also highlight a positive effect on students' confidence and self-assessment. However, these effects do not automatically translate into students' final retention and learning. In this regard, the ability to have at one's disposal immediate solutions may hinder deep learning, as students bypass the work required to internalise concepts. In fact, cognitive load theory suggests that excessive assistance reduces mental effort, preventing students from actively engaging in knowledge construction, no matter if the assistance is by humans or AI. In studies about AI tutoring, human tutors evaluate generated answers positively, almost at the level of human tutoring. However, the performance of AI tutoring without constraints compared to its human counterpart is still unknown.

How can we reconcile apparently opposite results in the field?

The diversity of results in the field can be justified by two main elements:

- The field is still in its infancy; as shown in [208], a related literature review on empirical studies only reports thirty-seven studies. It is natural that this leads

to a great variation of results as the novelty gives great space for exploration.

- Mainly, the results are not necessarily conflicting; effects on motivation and self-efficacy reports do not necessarily translate to performance or retention (as also seen in 3). Students can feel more motivated and more confident in engaging with their tasks, but at the same time, not fully absorb the necessary information.

What can we learn for future empirical research involving generative AI in computer science education?

We have a few learning outcomes from our literature review:

- On a prescriptive note, alignment between research questions and testing methods is fundamental, especially for educational research in younger fields. When focusing on computational thinking skills development, it is valuable to select models that adhere closely to the subject practice (i.e., programming). For example, some models are more focused on programming practice (such as [111]), and they could arguably be better tools to test students' improvement. As for programming skills, we suggest that participants could be tested on their performance in completing assignments without the help of generative AI or, alternatively, with aimed questions, testing specific concepts covered by the curriculum.

- Defining opportunities for future studies, we notice potential in the application of generative AI as tutoring. With the term "tutoring", we refer to contexts in which students are provided with limited AI tools. They are able to use it for hints and directions, but their freedom of interaction is previously regulated by human teachers. We argue that the result is that teachers are able to provide a high number of students with necessary support while retaining control over the learning process. As a corollary of this outcome, current literature seems to suggest that free, unrestricted access to generative AI in programming learning environments can be deleterious.

4.1.3 Future Work and Final Considerations

Future work in the field should focus on two main directions. First, it is important to continue to develop empirical literature to paint a clearer picture of the impact of generative AI on programming education. In particular, additional research focused on the performance and retention of programming knowledge and skills is needed. While

performing evaluations in an actual teaching context is extremely valuable, we have to consider that the ubiquity of generative AI could influence the results in unpredictable ways. Therefore, it could be valuable to start in smaller and more controlled settings. The second direction involves the study of methods to restrict AI and direct its potential towards specific uses. If, as argued above, controlled prompting has a positive impact on education, then we need to investigate how to design frameworks that can act as mediators. In this regard, interesting studies can be developed in the field of games and gamification (as in the case of [214]). These are well-established media for integrating both students and intelligent algorithms.
In this section, we reviewed examples of current empirical research in the field of generative AI for programming education. We have seen some recurring pitfalls and characteristics of related experiments. In particular, we noticed that, in some cases, testing methods should be carefully reviewed to better match research questions. On the other hand, experiments also highlight opportunities for future developments. In this regard, the effect on students' motivation and acceptance is noticeable. Moreover, experiments that include some form of restrictions or control on student-AI interaction yield promising results. Interestingly, in these types of experiments, the AI seems to take more of a tutoring role than a simple problem-solving tool. Generative AI has a definitely disruptive impact on programming education. However, as with many other digital tools, related research can teach us to control this impact and direct it towards having a positive effect. Of course, this requires time to explore different possibilities and free ourselves from potential preconceptions. Our contributions aim to provide a different perspective and, hopefully, some guidance for future research in the field.

4.1.4 Section Summary

The previous section introduces the impact of generative AI, in particular LLMs, in programming education. Many of these considerations can be transposed outside the specific field of application to different subjects. We see that, while a lot of research stemmed from the enthusiasm for these new technologies, it often skipped very important elements related to the analysis of students' performance. As such, many studies cannot conclusively find beneficial effects of LLMs in education. On the other hand, when proper metrics are applied, we see a strong negative effect related to the tendency to plainly take the generated output without a full understanding of it. There is, however, hope within certain specific contexts. For example, research has shown the potential to use LLMs as tutors, using parameters to limit their tendency to

take over students' problem-solving. Others have seen potential in the application of AI agents in gamified environments. In the following section, we follow an explorative process to investigate the potential for LLMs to be used in game-based learning. We follow a game design approach with the goal of creating an LLM-powered NPC in an educational video game. While potentially providing information to win the game, the NPC aims to transmit knowledge in an engaging and personal way.

4.2 Video Games as Mediators of Generative Artificial Intelligence

When it comes to learning, we have seen that games and gamification have strong effects on student motivation and engagement [33]. In fact, this is one of the salient aspects of play in general. However, the impact on final performance and retention is more complex to evaluate and often depends heavily on the subject and game design. In general, it is undeniable that video games are a signature medium of our time, and most people nowadays are able to interact with them intuitively. Their studied effects and accessibility make video games one of the most appealing media for educational research.

Another important digital development of the last 20 years is AI. When it comes to education, AI has a much more complex history. Although we strive to introduce components of AI into many school curricula, its actual impact on education is quite disruptive. In particular, generative AI has been noted as problematic in terms of retention and learning[206, 30]. However, many studies highlight the potential for generative AI to be used as a tutor for students, scaffolding actual learning by focusing more on teaching than direct problem-solving [215]. We talk, in this case, about *restricted* generative AI, introduced within a framework that:

- engages and challenges students, providing both intrinsic and extrinsic motivation [219]
- guides problem-solving without fully engaging with it

The purpose of this section is to highlight critical design decisions and existing challenges involved when introducing generative AI in education video games. In order to do so, we engage in a simulated design process, aiming to create an information game for the purpose of history education. While designing a minimum viable product

for testing, we journal through important decisions to consider about the integration of LLMs in video games. Finally, we evaluate the resulting game in terms of effectiveness in meeting the requirements and its viability within the current technological framework.

4.2.1 Background

Applications of generative AI in game-based learning and gamification (and vice versa) are still quite recent; a recently updated review on studies involving LLMs and games does not report even one involving game-based learning [220]. However, research is starting to develop in different directions. A very prominent one is the use of game elements for AI education [221]. In the same category, more constructivist approaches have been suggested to transmit AI interaction skills *through* play [222]. In this regard, AI plays the role of a topic in these game research studies. Other studies involved LLMs as players' evaluators in the context of education [223]. In this case, the AI is playing the role of analyst. Finally, other studies use generative AI to design narrative-based games for education [224], hence using AI as a designer. In this section, we focus on generative AI as a tool to provide educational information to players within video game contexts. In particular, we involve three fundamental parameters: role-playing, accuracy and viability. When we talk about **role-playing**, we refer to the ability of an LLM to interpret a character situated in the digital context of a specific video game. Current research on the topic attempts to create more reliable personas through the use of detailed profiles [225]. Additionally, role-playing prompting can improve output accuracy [226]. However, evaluating the level of role-playing efficacy is complex, especially when large amounts of conversational data are involved [227]. Some studies propose LLM-powered evaluators, but there are still limitations related to the stochasticity and unreliability of these models [225, 228]. Other studies focus on the use of both demographic information and opinion training in order to achieve the best alignment between LLMs and humans belonging to the same demographics [229]. This opens the creation of intelligent agents able to interact more realistically in their belief network. However, current research remarks that not all roles are played the same. For example, LLMs perform better when they interpret a doctor than a family member or an animal. Additionally, they seem to rely heavily on cultural stereotypes and biases to build their character [230]. Accuracy is another key concern for generative AI and relates to the actual correctness of the information the LLM provides. This is particularly important in the context of our research due

to the educational perspective. We have already mentioned the risk of hallucinations and incorrect pieces of information that are sometimes very difficult to spot [143]. In educational settings, this issue has a deep and troubling impact on the potential applications of generative AI [231]. Moreover, it is exacerbated by the aforementioned risks associated with excessive reliance on LLM-powered tools and the tendency not to critically analyse their output. Finally, viability relates to the actual capacity for the LLM to be embedded in video game systems without negatively impacting their functioning and performance. Previous research investigates the processing power required to run an LLM; it reports encouraging considerations related to the possibility of running smaller models in most AAA games [136]. However, these studies focused on specific contexts that are not necessarily translatable to educational applications, which often rely on video games of a smaller size. At the same time, the requirements for LLMs in educational contexts are quite high, and this can conflict with the use of smaller models.

4.2.2 Research Question

While existing research touches upon several elements related to the use of LLMs in education or video games, our goal is to take a holistic approach to educational game design and LLMs. The central topic of this section is to investigate how LLMs can be implemented in game-based learning. In this regard, we develop a series of subresearch questions:

- **what design considerations influence the development of a game incorporating LLMs with educational purposes?**
  We focus on the design requirements that arise specifically from the incorporation of text-based generative AI.

- **how do characteristics of models and prompting influence the quality of the final product?**
  We test different models with different sizes. We elaborate on how feasible their implementation is. We also experiment with prompting, investigating how well LLMs can follow relevant instructions for educational contexts.

- **how effective is an LLM in balancing role-play and educational content?**
  In the previous questions, we focus on requirements for LLMs as tutors. However, (educational) video games have natural requirements related to consistency

in terms of narrative engagement. An AI-powered NPC needs to be able to maintain a persona in order not to disrupt the player's experience.

- **how viable is this incorporation with current technology for the general public?**
  We summarise the findings from the previous questions, evaluating the actual viability of this type of game-based learning.

4.2.3 Methodology

We simulate a design process for an education app incorporating LLM characters. We aim to develop a minimum viable product to test different parameters and evaluate the final experience in terms of role-playing, accuracy and viability. We report the considerations that arise from the process and the analysis. We decided to contextualise the game project on the subject of Latin history. The choice has been made by considering the opportunities arising from the simulation of interactions with historical civilisations and the relative presence of facts that can be objectively evaluated in the subject. The product takes the form of an information game in which the player is thrown back to the year 1 BC and needs to figure out the location and construction context of the Ara Pacis (Roman monument built in 9 BC). As an educational goal, the player is supposed to acquire knowledge related to everyday life in the Roman Empire by interacting with LLM-powered characters. It is important to understand that the game characteristics can influence the design process and the result of the successive evaluation. For example, certain language models might be inherently more effective in portraying NPCs from other civilisations or eras [230]. Therefore, the results of this study should be considered in the presented context. We evaluate the process and the result with regard to the following parameters:

- Model characteristics: we use different LLMs with different characteristics. We will explore bigger models and smaller ones. We finally evaluate the results in terms of viability and accuracy and identify possible relations with the model characteristics.

- Contextual information: we provide the LLM with contextual information related to its persona and the historical context. We also provide information about the style of the output that is supposed to be generated and its scope. We edit and experiment with prompting to identify possible strategies to improve role-playing and accuracy.

- Integration: considerations related to the integration of the model in the game environment are relatively more straightforward. We evaluate the options of using a local or online LLM and critically discuss the implications with regard to viability.

We finalise our investigation by summarising our findings. We also report on aspects of these systems that are still challenging for current technologies and that necessitate further research.

4.2.4 Results

Design Considerations

When designing educational games involving LLM technologies, specific considerations arise throughout the process. The first element, in the case of generative AI being used to power NPCs, would be the interaction type. Text-based role-playing games (RPG) with natural language as an input component are not unusual; probably, the most famous example in this case is the video game Zork [232]. However, up to today, these games rely on a hard-coded interpretation of the user's textual input. Instead, LLMs allow for actual interpretation and reaction to users' interactions. This impacts design in two directions: the input and output interfaces. In the case of the input interface, the design might need to control the maximum length of users' messages. This can be a challenge with current models (see 4.2.4). Moreover, it is important to filter the input in order to avoid undesirable outputs as a reaction and restrict the possibilities for users to intentionally tamper with the model's alignment. Similarly, the output message might need to be constrained, and this can present a challenge for current technology (see 4.2.4). Finally, especially in educational contexts, the output of an LLM is considered dangerously unpredictable. However, in our experimentation, we find that most models are able to avoid producing inappropriate output. In this regard, the design consideration is to pay attention to experimenting with the selected model. Due to the inherent stochasticity of AI, certain filters for inappropriate responses should still be put in place. The narrative design can also be influenced by the introduction of LLMs. In this regard, the AI can generate content that inadvertently conflicts with the designer's plan. A good practice is to design for flexibility, creating systems of narrative and game objectives that take into account the lack of control over the conversational aspects of the video game. Additionally, in the context of education, it is helpful to have more general learning objectives (such as the one we illustrated for our game in 3.2.4). Conversely, designing with LLMs in

order to elicit specific knowledge transmission can expose to the risk of inaccurate or misdirected information.

Model Characteristics

First, we want to define how the model's size impacts accuracy. Then, we will evaluate the viability of using different models in educational games. When it comes to accuracy, the relation between model size and performance is not necessarily linear [233]. However, with simpler goals, such as in our case, the information quality definitely improves with bigger models. In our case, we experiment with various models presenting three main size groups: between 6 and 8 billion parameters, between 13 and 15 billion, and 70 billion. The choice of size groups has been dictated by sampling reasons. Moreover, bigger models have been excluded for technical availability reasons, which would, anyway, make them intuitively impossible to apply in reality. After experimentation, the models that could be run locally with acceptable speed and with commonly available hardware were only those between 6 and 8 billion parameters. Those between 13 and 15 showed a great improvement in terms of accuracy, but not specifically in terms of role-playing capabilities. Moreover, they demonstrated to be far slower. The models with 70 billion parameters did not show a great improvement in terms of accuracy compared to their 13-15 counterparts.

Contextual Information

Within the context of a video game, we need to build a prompting framework that reinforces specific desirable behaviour from the LLM. The prompt used has a fixed structure that always ends with the formula *"Behave and react to the text only in ways appropriate for your character"*. In our experiment, we manipulate specific aspects that come before the aforementioned formula:

- Character and historical context: we change the level of details we provide to define the character and how they relate to the historical context. The first experiment utilises fully natural language with fluent syntax and goes into relative details about the life and the context of the persona selected. The introductory prompt is *'You are a Latin noblewoman living in Rome in 1 BC during Emperor Augustus's reign named Lucillia.'*. In the second attempt, we play mainly with syntax, building the information in shorter sentences: *'You are a Latin noblewoman named Lucillia. You live in Rome in 1 BC during Emperor Augustus's reign. You were born in 20BC.'*. Finally, we use a less specific prompt

to test whether the result can improve: *'You are a Latin noblewoman in Rome during Emperor Augustus's reign.'*. We experimented with keeping the input "Hello, where is the Ara Pacis?" as the most direct and basic interaction possible. While the agents provide valuable knowledge, each prompting style has some drawbacks; with the first two, the agent does not stay in character. With the last, the output provided is extensive and goes way beyond the simple question. All three styles of prompting output partially incorrect information: the first two, when they break out of character, provide a wrong name for the museum holding the Ara Pacis today. The last one gives an incorrect original location of the monument. In all these incorrect cases, we notice the LLM resorts to more generic and stereotypical expressions, using locations such as the Roman Forum or the Museum of Roman Civilisation (which, by the way, is then incorrectly translated to Italian) in its mistakes. In terms of sheer educational quality, the third prompting style stimulated the most extensive responses and, with these, it provided the most complete information about Roman lifestyle and civilisation.

- Educational context: We provide details about the educational context in which the product is supposed to be applied. In this regard, we experiment with explicitly adding to our input *'Try to also provide educational information about life in the Roman Empire.'*. The results are quite evident; this input does provide a good context for application to the LLM, which is able to respond with much more extensive and detailed information about Roman civilisation. Moreover, it seems able to adapt to its context of application and has a more conversational style in providing answers. However, in this case, the final results often contain incorrect or imprecise information.

- Additional reinforcement: we change the level of explicit information we provide in order to keep the model on the character and avoid providing unrealistic information. The first experiment aims to improve role-playing by explicitly asking the LLM to avoid using information not available for their persona. Therefore, we add the sentence *'Do not use information not available in your historical context.'*. With this addition, the occurrence of incorrectly contextualised messages was greatly reduced. However, the accuracy of the rest of the information provided is not necessarily improved. Additionally, we experiment with a broader, yet simpler, addition: *'Never break character.'*. This edit seems to be the most impactful among those tried. In all cases mentioned above, adding this sentence results in content that is more accurate, contextualised and concise. On the

other hand, the information might be more generic. Finally, we attempt to improve conciseness by adding *'Keep the answer under 60 words.'*. The results are disappointing; the LLM completely disregards this edit in all cases.

Integration

The last component through which we want to analyse the minimum viable product we proposed as a case study is integration. In this regard, we take into account the aforementioned information and discuss its implications in terms of actual viability for future applications of LLMs in educational video games. From experimenting with model characteristics, we come to the conclusion that running models locally dramatically limits the integration of bigger (bigger than 13b) models. As long as we keep the model local, we always have to balance accuracy with the reaction speed of our NPCs. Smaller models that could easily run locally performed extremely poorly for the standards of an educational game. Besides failing at times to role-play and presenting engagement-breaking information, they often presented wrong facts. Bigger models, on the other hand, are not viably runnable on the average laptop and have excessive latency times between input and output. Therefore, using an online (via API) solution is the only way to move forward in terms of integration. In this case, we recommend testing first mid-size models (13 to 15 billion) since they often offer good accuracy without the computational power required by their bigger counterparts. Obviously, there are drawbacks to relying heavily on online models. First, they require a stable internet connection. Second, depending on where the servers are located, they might present some criticalities in terms of privacy. Generally, in terms of integration and its viability, LLMs in video games with educational purposes are required to be used in contexts with considerable resources and, as such, they lose some of the universal applicability that characterises educational games. However, it is definitely possible to design them and run them in some more privileged contexts.

4.2.5 Discussion and Conclusion

Our explorative investigation into the use of LLM for educational video games reveals certainly potential for these systems. However, there are relevant challenges that make the applications of these tools unfeasible at the current time. In terms of design considerations, because of the sensitivity of LLMs to specific forms of inputs, filters are necessary to avoid misalignment. These can even be hardcoded, and while we discover more about AI alignment, we will have a better grasp of what to target precisely. Our

investigation also shows that the risk of random inappropriate output is greatly reduced as long as the aforementioned misalignment filters are in place. However, LLMs tend to still perform poorly when it comes to following plot and persona descriptions. This translates into the necessity for more flexible and generic game narratives, which might be in conflict with the nature of the subject. A big drawback that we highlight both in the design considerations and the prompt experimentation is the difficulty of respecting certain limits to the length of output. This can impact the user interface design.

The characteristics of the model and of the contextual information provided obviously play an important role in the final product performance. Bigger models tend to improve accuracy, which is an important aspect for educational activities. However, when it comes to a locally run model, striking the balance between accuracy and computational performance can be extremely challenging. Smaller models (6-8 billion) tend to perform relatively poorly, at least in the context of this investigation. Bigger models (13-15 billion or above) are quite slow if we consider the average hardware available to students. On the other hand, online integration can, in part, solve these issues, even though some privacy concerns need to be considered. However, using local servers and stable internet connections, these drawbacks can be somewhat limited. As for the contextual information provided, we see how models tend to provide more educational information when we build a more generic persona. Also, specifying the educational settings has a positive impact on the output; it also slightly changes the registry of the agent towards a more "tutoring" tone. Finally, to improve role-playing in the settings of this investigation, we tried to confine the agent, asking it not to provide historical information unavailable to its persona. Although this had some impact, it did not fully avoid incorrect or unbelievable behaviour. Also, in this case, the more generic demand not to ever break character seems to be more effective. In general, we find that LLMs have limited role-playing power that improves when the persona and the boundaries are described more generically. Although this is to be expected, it also strongly limits the control that the designer or educator has over the tool and the narrative.

In general, with regard to our third research question *"how effective is an LLM in balancing role-play and educational content?"*, we can say that, especially in the case of smaller models, current tools perform poorly. While role-playing can be improved with more generic contexts, over many interactions (which would also happen in real educational settings), the LLM has been demonstrated to be unreliable, often providing incorrect information. Often, this information was generated to look correct,

basing itself on stereotypes and likely knowledge. This is definitely a significant challenge for the application of LLM technology in education. However, it also opens opportunities to incorporate output reviewing habits in education. This can be structured as a learning activity in itself with the additional function of mitigating LLMs' hallucinations and inaccuracies.
Finally, when it comes to integration and the related viability, although some of the challenges we mentioned above persist, bigger and more reliable models could be integrated in a bigger system involving online implementations and local servers to provide the necessary computational power. These solutions are, however, extremely expensive and complex to realise. On the other hand, the field is advancing at a rapid pace, especially in terms of making models more lightweight and efficient [234]. In general, considering the unreliability of the information provided, while integration is indeed possible, at the current stage, we argue is not worth the investment. Moreover, one of the founding aspects of game-based education is the possibility for it to be available for many users, transmitting knowledge in engaging settings, even with relatively limited tools. With the current technology, this would not hold true anymore in the case of LLMs for education; as mentioned above, the infrastructure necessary to make it viable is quite extensive and, arguably, available only in privileged contexts. Considering how impactful the digital divide already is and all the challenges yet to face in order to make the environment effective, we argue that this type of educational/game system is not worth the investment necessary.

Limitations

As mentioned in the introductory information, our exploration is extremely limited in many aspects. We did explore the application of LLMs in the specific context we described: the subject of Latin history and the form of an information game. Our considerations can vary widely, even in slightly different settings. However, most of the critical points we highlighted might endure. Moreover, the settings we selected have characteristics that are arguably favourable for LLMs (very defined goals, generic settings and surface knowledge of the topic required). While the change in settings surely has a big impact on our considerations, it is not likely that the overall performance would improve.

4.2.6 Section Summary

In this last section, we mainly tackled RQ8;. We argued in the previous section that AI can be quite performative in the role of tutor for students. While this might hold true, the implementation of the technology in video games presents several obstacles, especially for the educational nature of the settings. The stochasticity and tendency to use and produce generic information make LLM-generated messages unreliable for education. Moreover, the necessary precautions to make the game-LLM system even viable are relatively expensive and arrive anyway to achieve mediocre results. However, it is also important to note that we explore very open topics, in which the LLMs were mostly evaluated based on the quality of their guidance over non-specific information (in our case, Latin culture and civilisation). Other research that we presented in 4.1 showed potential in the use of the technology in contexts involving very specific knowledge. For example, it is likely that generative AI technologies can have viable and positive applications in guiding students in debugging exercises in programming education. While further research and new methods are necessary, and steps need to be taken towards the adaptation of teaching techniques to the existence of these new technologies, we recommend caution in the acritical application of AI in education research.