Multimodal Technologies and Interaction

MDPI

Article

Creative Use of OpenAI in Education: Case Studies from Game Development

Fiona French *, David Levi, Csaba Maczo, Aiste Simonaityte, Stefanos Triantafyllidis and Gergo Varda

School of Computing and Digital Media, London Metropolitan University, London N7 8DB, UK
* Correspondence: f.french@londonmet.ac.uk

Abstract: Educators and students have shown significant interest in the potential for generative artificial intelligence (AI) technologies to support student learning outcomes, for example, by offering personalized experiences, 24 h conversational assistance, text editing and help with problem-solving. We review contemporary perspectives on the value of AI as a tool in an educational context and describe our recent research with undergraduate students, discussing why and how we integrated OpenAI tools ChatGPT and Dall-E into the curriculum during the 2022–2023 academic year. A small cohort of games programming students in the School of Computing and Digital Media at London Metropolitan University was given a research and development assignment that explicitly required them to engage with OpenAI. They were tasked with evaluating OpenAI tools in the context of game development, demonstrating a working solution and reporting on their findings. We present five case studies that showcase some of the outputs from the students and we discuss their work. This mode of assessment was both productive and popular, mapping to students' interests and helping to refine their skills in programming, problem-solving, critical reflection and exploratory design.

Keywords: artificial intelligence; OpenAI; ChatGPT; Dall-E; LLM; procedural generation; interaction design; game design; game programming; education

check for updates

Citation: French, F.; Levi, D.; Maczo, C.; Simonaityte, A.; Triantafyllidis, S.; Varda, G. Creative Use of OpenAI in Education: Case Studies from Game Development. *Multimodal Technol. Interact.* **2023**, *7*, 81. https://doi.org/10.3390/mti7080081

Academic Editor: Deborah Richards

Received: 20 July 2023
Revised: 10 August 2023
Accepted: 16 August 2023
Published: 18 August 2023

1. Introduction

Recent commentary around the possible advantages and drawbacks of generative artificial intelligence (AI) systems, such as GPT-3, in the context of education, has included plenty of recommendations but few examples of actual practice with students. For example, Crawford et al. [1] acknowledge that ChatGPT can support learning outcomes, while simultaneously calling for new leadership in higher education that will ensure that AI is used ethically. Similarly, Yu [2] recommends regulation rather than prohibition, emphasizing how AI will change education in beneficial ways. They advocate that educators should focus on "students' critical thinking, problem analysis, and effective solution-seeking abilities". Chen et al. [3] offer another positive perspective, focusing on the potential for chatbots to engage students and facilitate an inclusive learning experience, particularly when there is a high student–staff ratio. Farrokhnia et al. [4] also highlight some of the strengths and opportunities of AI in their S.W.O.T. study, including easy access to information, the potential for personalized learning, and the chance for students to develop creative and critical skills.

On the other hand, weaknesses and threats are also identified by the previously mentioned S.W.O.T. [4], whose authors warn that reliance on AI may result in "a lack of understanding of the context" and "declining high-order cognitive skills", resulting in decreased academic integrity. Cotton et al. [5] raise the problem of detecting plagiarism, while somewhat ironically attributing half the text of their article to ChatGPT itself. In addition, Limna et al. [6] take a wary stance, pointing out that there are safety, security and privacy risks associated with AI technologies, and emphasising the need to "implement appropriate strategies to meet teachers' and students' needs".

It seems clear that the contemporary rise of AI tools has provoked some anxiety within the education community and is raising many, as yet unanswered, questions. At present, we note that much of the literature surrounding the use of OpenAI tools and other generative AI systems in education is theoretical, although there have been some valuable studies that report on practice.

A case in point is Puryear and Sprint's evaluation of Copilot, an intelligence-driven development environment (AIDE) provided by Github [7]. They recommend teaching students to use this tool as part of a programming workflow, since they suggest it is likely to evolve into a new code completion standard [8]. Another example in the field of programming is provided by Huang et al. [9], who evaluated the use of AI-enabled video recommendations to motivate students in a flipped classroom (a type of blended learning that involves lecture material being available for students to study in their own time, thus providing more opportunities for problem-solving activities during class). After comparing two cohorts, together comprising over one hundred students taking a systems programming course, the researchers established that personalised video recommendations improved outcomes in students who had moderate motivation at the outset, but not those with either high or low motivation.

A further example of an educational area where AI is increasingly being used is language learning, where conversational tools have been shown to decrease human teachers' workloads [10]. In the context of second language learning, Alharbi [11] concludes that since students will undoubtedly make use of text generators and other writing tools, the challenge of regulation falls on educators and researchers, seemingly ignoring the possibility that students might also choose to take responsibility for their actions.

Our purpose in this paper is to describe and evaluate our experiences using AI tools with BSc Games Programming undergraduates as part of their coursework. The intention with this project was to actively encourage students to explore the use of OpenAI and other AI toolsets to support their work in game development. The objectives were to provide them with opportunities for focused engagement with OpenAI that would enhance their technical and problem-solving skills, as well as refine their abilities to analyse the current capability and potential of the technology, based on their own experiences. An important goal was to enhance communication skills in relation to AI technologies, moving away from social-media-framed hype and towards a more rigorous, well-informed perspective. It was anticipated that further benefits would be accrued via students' motivation to read around the topic and subsequently synthesise information produced by other researchers and developers.

An additional reason for advocating a critical approach to the software was to reduce students' fears around AI replacing them in the future workplace, a common anxiety that emerged during informal discussions. Dramatic social media and Internet posts have suggested that applications such as ChatGPT and AI image generation software are already capable of taking over a range of jobs, such as those that involve copywriting, customer support and illustration, but also including some junior programming roles [12,13]. More measured responses to the increase in generative AI chatbots may not attract so much attention as they are typically found in longer, more formal pieces. For example, in the context of programming, Surameery and Shakor comment that "ChatGPT should be seen as one part of a comprehensive debugging toolkit and used in conjunction with other tools and techniques" [14]. In the field of healthcare, it has been acknowledged that AI can be very useful as a diagnostic support, but researchers agree that it is not yet sufficiently reliable to be trusted without human oversight [15–17].

Borji [18] offers an analysis of ChatGPT's failures, highlighting its limitations in reasoning, problem-solving and bias-reduction. Perhaps surprisingly, Borji suggests that we do not know the extent to which ChatGPT 'understands' its responses, whereas we *do* know that although the agent is taught to understand the rules of natural language, it cannot truly understand any conversation in the same way that a human can. This is because its responses are generated as a result of training the model on huge sets of linguistic data using neural networks with sophisticated algorithms to create a large language model (LLM). Human feedback is subsequently deployed for reinforcement learning (RLHF), which is a method used for fine-tuning the results, devised in 2017 [19].

Despite the results being plausible, "computation cannot realize understanding" and "AI is stupid", according to Bishop [20]. Floridi [21] describes AI as "agency without intelligence", while Bender and colleagues [22] manage to insult birds by calling LLMs "stochastic parrots". Nonetheless, in the publicity around ChatGPT, there is still a tendency to imbue the software with some kind of overarching intelligence, or rationality, leading to the idea that we might depend on AI to provide trustworthy information and make ethical decisions—perhaps because of its breadth of knowledge and ability to converse in a variety of natural human styles. We deemed this disparity of judgement a good reason for engendering more critical awareness and scepticism in our student population, as explained in the following description of our approach to this project.

2. Method and Rationale

Undergraduate BSc Games Programming students at London Metropolitan University take two AI modules during their third year of study, providing opportunities to discuss and investigate current developments in AI. During 2022–2023, there was widespread interest amongst the cohort in the OpenAI tools, ChatGPT 3 and Dall-E.

2.1. Overview of AI Modules

During Autumn Semester, the first AI module (Artificial Intelligence) is shared with BSc Computing and BSc Software Engineering students, as one of their electives. During Spring Semester, the second AI module (AI for Games) is delivered only to BSc Games Programming students. The assessments are as follows:

- Module 1, Assignment 1: Presentation of Research (weighted 25%);
- Module 1, Assignment 2: Artefact (weighted 75%);
- Module 2, Assignment 1: Report on AI for Games (weighted 25%);
- Module 2, Assignment 2: Application of AI to Games (weighted 75%).

These two modules are each worth 15 credits. Taken together, they comprise one quarter of the 120 credits required to pass the year, which means that Module 2, Assignment 1 represents just 3.125% of the total possible marks that games programming students can be awarded. The case studies presented in the paper showcase a selection of outputs from this particular assignment, but we provide details of our educational approach to the topic of AI throughout the year, linking this to our pedagogical rationale and commitment to providing a creative, challenging and inclusive learning experience for students.

2.2. Module 1—Lectures and Discussions

Tuition on Module 1 comprises formal lectures that focus on fundamental knowledge, such as the foundations and history of AI, different types of agents and decision-making algorithms, representations of knowledge and currently solved problems. Philosophical and ethical questions are discussed freely and respectfully in class. Our student cohort is varied, so it is to be expected that people will have different perspectives on a number of issues.

Rather than using a didactic style, we preferred to approach the topic of OpenAI in a facilitatory manner, encouraging students to draw their own conclusions, based on evidence they could subsequently reproduce, if they wished. Facilitation has been described by Gregory as "helping learners realise their capacity to learn" [23]. Some of the limitations of Dall-E-2 (OpenAI's text-to-image generation software) were therefore illustrated by sharing the outputs generated by specific prompts. For example, the prompt: "Photograph of nutritious English meal on white tablecloth" showed how susceptible the software can be to cultural stereotyping (Figure 1). Moreover, representations of people can be outdated, demonstrated by the prompt: "AI developer sitting on the moon with a pizza and a computer", which resulted in a curious selection of humans and one token green alien (Figure 2). Everyone was eating pepperoni. The depictions were certainly not representative of the diverse cohort of twenty students taking the class.

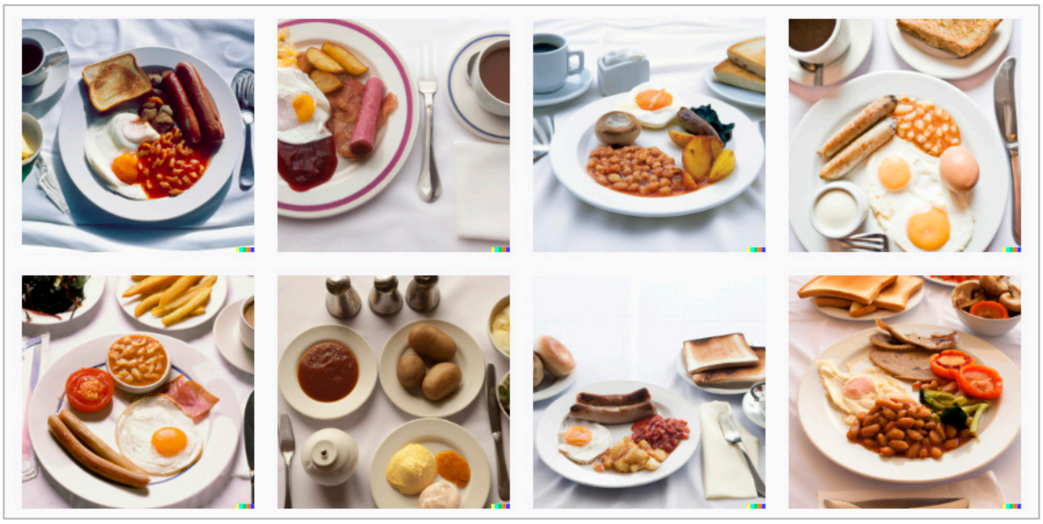


Figure 1. PROMPT: Photograph of nutritious English meal on white tablecloth—Dall-E-2 outputs, November 2022.

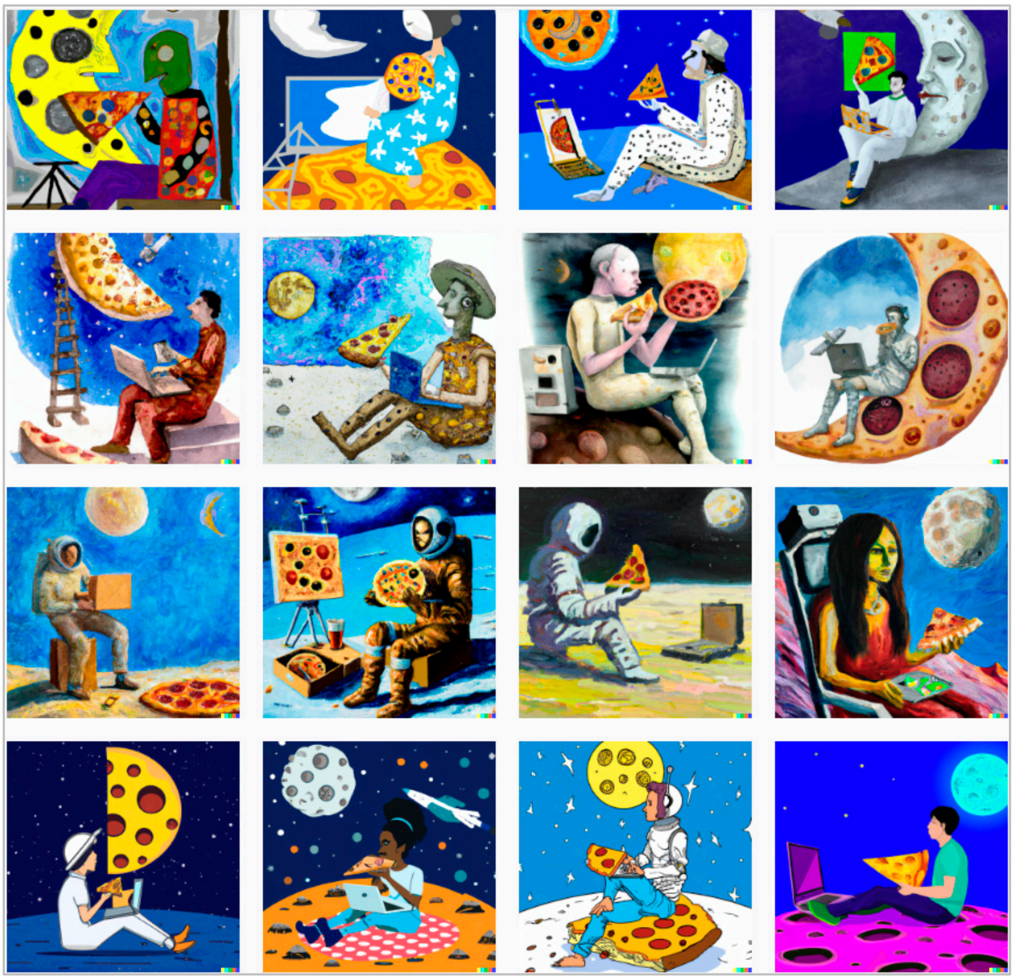


Figure 2. PROMPT: AI developer sitting on the moon with a pizza and a computer—Dall-E-2 outputs, November 2022.

Although these results may have been disappointing, they are understandable, since Dall-E-2 was trained on graphical data pulled from the Internet and classified by humans, which is highly likely to have introduced biases. OpenAI Docs acknowledge these biases and explain that their data filters unintentionally amplified some of the effects. This remains a challenge [24]. A similar problem occurs with text output because many large generative AI models (LGAIMs), including GPT, have been trained on a vast amount of historical Internet data drawn from the 'Common Crawl' publicly available dataset [25], as well as other sources, such as books and wikis. Although the quantity of data is huge, most of it was captured pre-2021 and is not curated.

Despite OpenAI controversially employing Kenyan workers to label samples of toxic data (both text and images) in an attempt to train the AI not to generate it [26], ChatGPT has been widely criticized for containing undesirable content in some of the outputs that it has produced [27,28]. Meanwhile, no information has currently been released about how GPT-4 is being trained, for how long or using what data.

In short, the visual examples provided to students quickly raised their awareness in relation to OpenAI, which was the intended consequence.

2.3. Module 1—Workshops and Practical Sessions

Module 1 also includes workshops that cover practical C# programming challenges, enabling students to develop a small prototype that demonstrates a more complex solution. For this purpose, we require students to deploy an industry standard game engine, Unity [29]. Game engines can used for a variety of purposes, such as creating simulations, training applications, virtual reality experiences and other interactive systems. Unity was selected because it is a cross-platform, real-time development platform with a low entry-level and it can be integrated with the Microsoft Visual Studio [30] development environment (for editing and debugging code). Moreover, it provides several useful features for developers, including ready-to-use 3D primitives, physics libraries, and access to a wide range of optional assets and tools. While the students' assignment is explicitly *not* a game, they are required to showcase some agent behaviours, such as state transitions, pathfinding, flocking or machine learning. Being able to represent these as a 3D, animated, interactive demo is highly motivating and offers a great opportunity to develop an engaging portfolio piece.

Amongst this mixed cohort, there is often a wide range of skillsets, because although computing and software engineering students can create applications, it is typically (but not exclusively) only the games programmers who have previously used a game engine (such as Unity) to undertake 2D and 3D graphics programming. However, the majority of the other students are very keen to learn how to transfer and apply their coding skills to a new API (application programming interface), such as a game engine. Therefore, we encourage partner work between games and nongames undergraduate developers. This benefits both, as the games programmers consolidate their learning and build confidence by sharing and explaining skills, while nongames-programmers can quickly learn one-to-one from experienced peers. The opportunity to work collaboratively with peer support has been commended for the positive impact it can have on students [31,32].

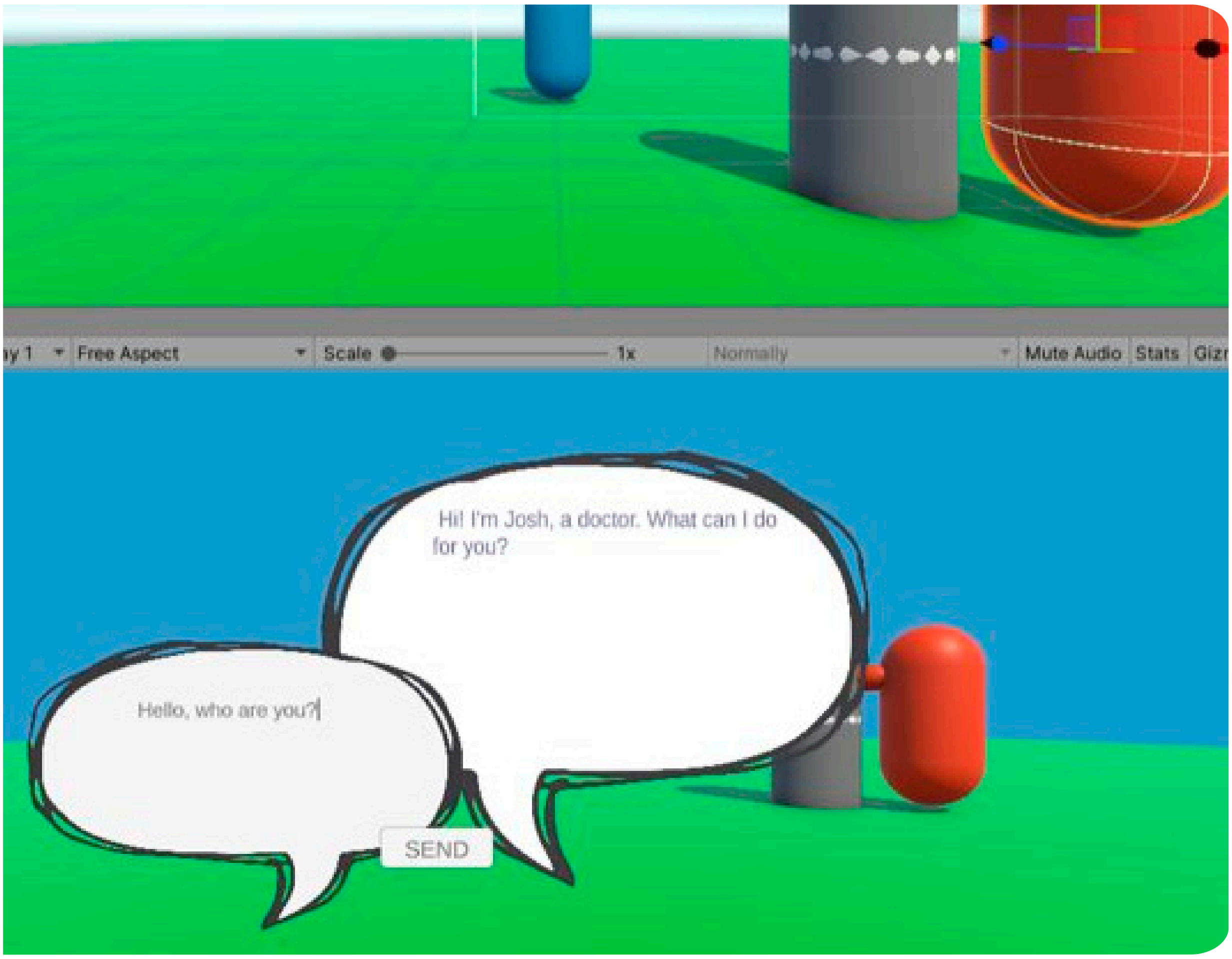
2.4. Module 2—Integrating OpenAI with Unity

As the games programming cohort were very interested in the publicity surrounding OpenAI, many had already been reading widely and researching information online. Therefore, in Spring Semester 2023, Module 2, Assignment 1 (Report on AI for Games) was rewritten to explicitly require the use of OpenAI tools in conjunction with the API used in class, with the aim of engaging students' critical sensibilities through their evaluation of OpenAI. Students were tasked with researching the use of ChatGPT 3 or Dall-E-2 in a game development context, writing a report on their findings and demonstrating results by integrating the AI software into a project using a professional standard game engine (Unity).

The students were given examples of possible small scope projects they could tackle, or they could choose to investigate their own ideas. Assignment details were as follows:

- Brief: Undertake some primary research into the use of OpenAI in a game (or other interactive) scenario, write a report and demonstrate with working examples. The report should not be overly technical and should be written in an interesting way so that people without prior knowledge of AI can find it interesting. Students should strive to make their own investigations (individual work).
- Requirements: Working prototype to be executed in Unity game engine; documentation including detailed explanation of techniques used, with screenshots and diagrams to supplement text; references to software and relevant literature are crucial.
- Grading criteria: Relevance and interest (20%); research skills, including analysis of problem and demonstrable testing (40%); synthesis, including sources and own voice (20%); presentation and analysis (20%).

In preparation for this assignment, students were provided with two working examples—(i) using ChatGPT as the voice (in speech bubble) for a nonplaying character (NPC), to enable a nonscripted conversation with a player (Figure 3), and (ii) using Dall-E to create textures from player prompts that could be applied to 3D models in real time (Figure 4).



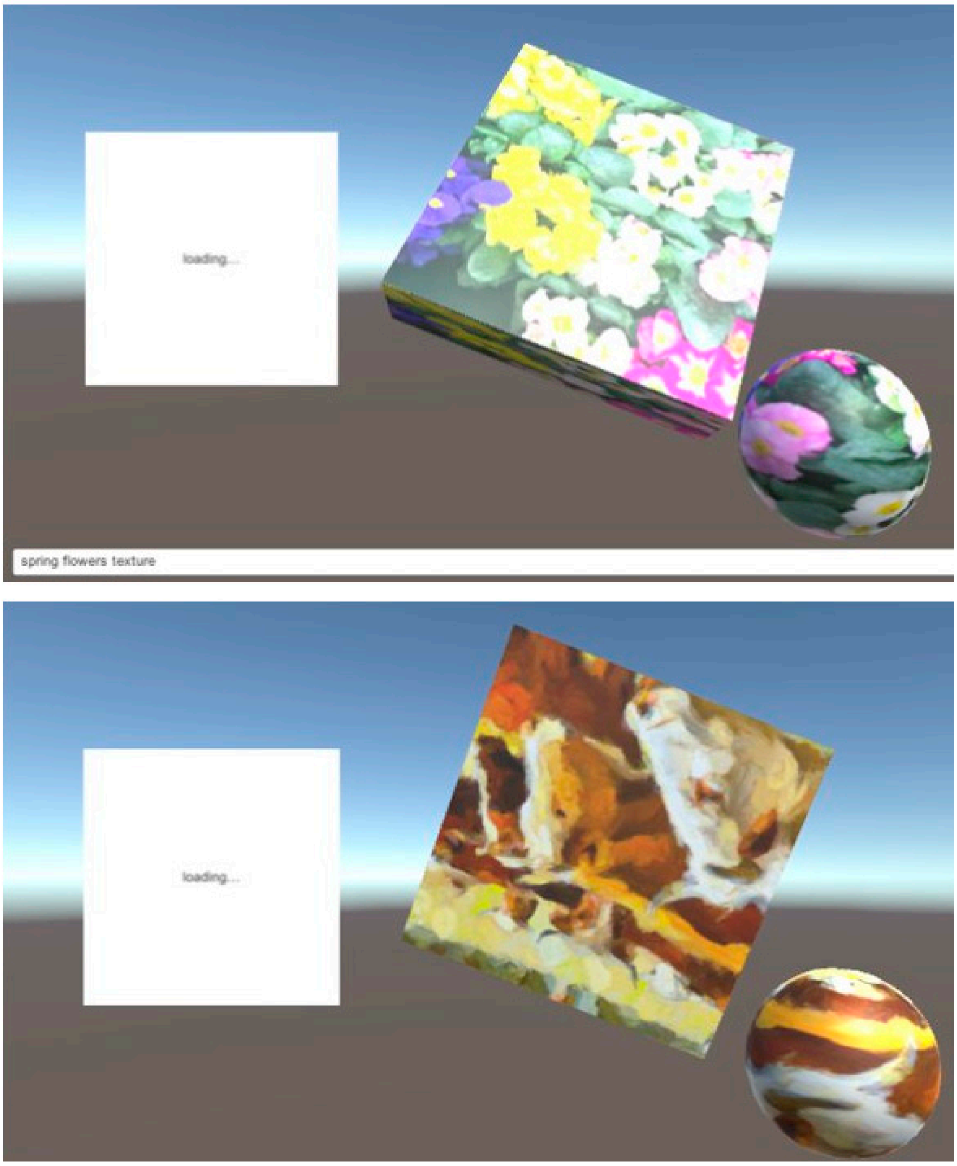
Hi! I'm Josh, a doctor. What can I do for you?

Hello, who are you?

Figure 3. ChatGPT conversation embedded in the Unity game environment, triggered by approaching an avatar (NPC).

In order to compile and test these examples, each student was required to create their own OpenAI account and generate an API key (for authentication). They then had to open a new Unity 3D project and import an unofficial OpenAI Unity package (created by Sercan

Altundas) from GitHub [33], including the two ChatGPT and Dall-E examples provided. To make the examples work, it was necessary to write a new JSON file defining their API key and save this to a specific folder in their user profile.



spring flowers texture

Figure 4. Dall-E images converted to materials and applied as textures for 3D objects in real time—PROMPT: spring flowers + PROMPT: herd of cows.

The ChatGPT application could then be launched and would run in the Unity editor. Students were shown how to edit the example scripts to provide information to the AI, adjust its settings and concatenate strings of data. They were given detailed instructions on how to achieve other tasks, such as replacing dialogue boxes with speech bubbles, creating scripts that endowed the NPCs with different personalities, and triggering the appearance of the canvas element (2D display for conversations) based on a collision or a selection.

The Dall-E example also ran in the Unity editor, generating 2D images in response to text prompts. To make it more interesting, students were shown how to enable in-editor image downloading, then write a script to set the image as the main texture on a material applied to a 3D model.

Support was offered to students in weekly workshops, and in the final class before assignment submission, each person presented their work in front of colleagues, as an informal viva. This provided a valuable opportunity for formative feedback, from the tutor and from other students. Feedback at this stage, when most of the work has been completed, has consistently been found to be helpful for the presenter, who must also prepare technical and reflective documentation as part of their submission. Responses and questions from peers indicated how interesting and relevant others found the work, as well as giving the presenter an opportunity to refine their communication skills. These attributes (of the work and the student) are valued in the grading scheme. Our approach to formative feedback is supported by Shute [34], who highlights the importance of timing and specificity, and by Al Jahromi [35] who found that tutor and peer formative feedback enhanced oral presentation skills in L2 (using second language) students. All in our cohort speak fluent English, yet it is the second language for six out of eight students.

3. Results: Case Studies

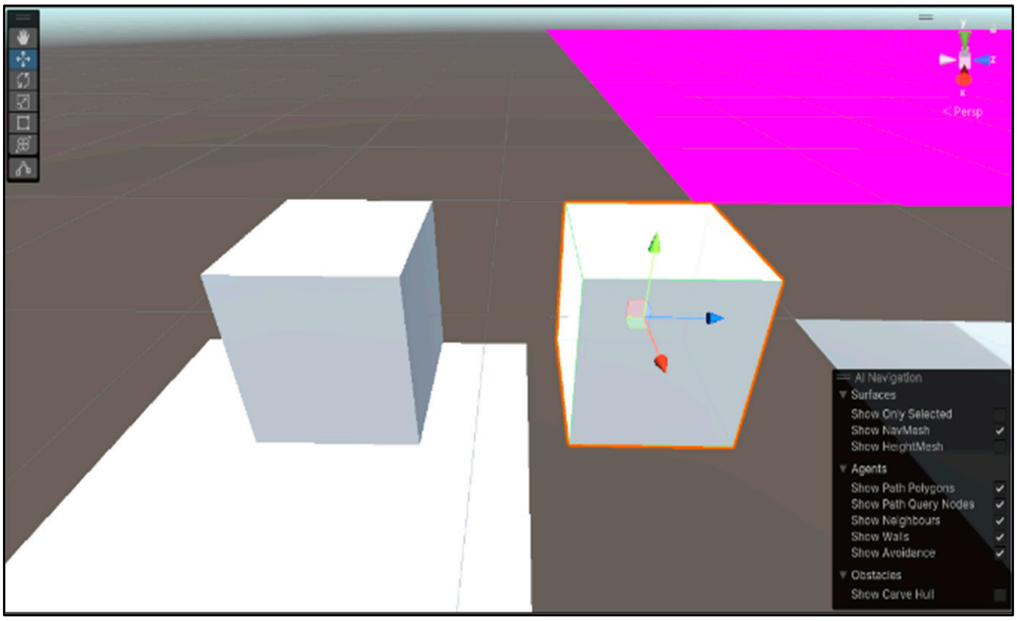
Students taking this assignment chose to investigate procedural generation of 3D objects and 2D graphics, code support, and dialogue systems for both NPC conversations and text-based gaming. It is noted that many reports included social media posts, software sites and Internet articles as some of their (nonacademic) reference material, pointing to the fact that these sources of information can be highly topical in a swiftly changing landscape.

The following excerpts present some of their outputs and perspectives.

3.1. Example 1: From 'OpenAI in Unity' by Csaba Maczo

This project tested the current (March–April 2023) capabilities of ChatGPT 3 to generate code that could be compiled and executed in the Unity editor during runtime using a freely available open-source asset from Kejiro that provided access to OpenAI and generated a chat window for the user [36].

It was possible to execute simple commands, but it was much slower using this method than interacting with the editor manually. For example, it was possible to generate a cube that was identical to a Unity primitive cube, with the same parameters (Figure 5). Spheres were also possible, and could be assigned random colours, scales and positions in the world (Figure 6). Unity terrains were problematic, with smaller dimensions than expected and no mesh collider (which is an essential feature of a terrain as it provides capability for collision detection).



Show Only Selected □
Show NavMesh ☑
Show HeightMesh □

Show Path Polygons ☑
Show Path Query Nodes ☑
Show Neighbours ☑
Show Walls ☑
Show Avoidance ☑

Show Carve Hull □

Figure 5. AI-generated cube on the left is identical to the manually generated Unity primitive on the right.

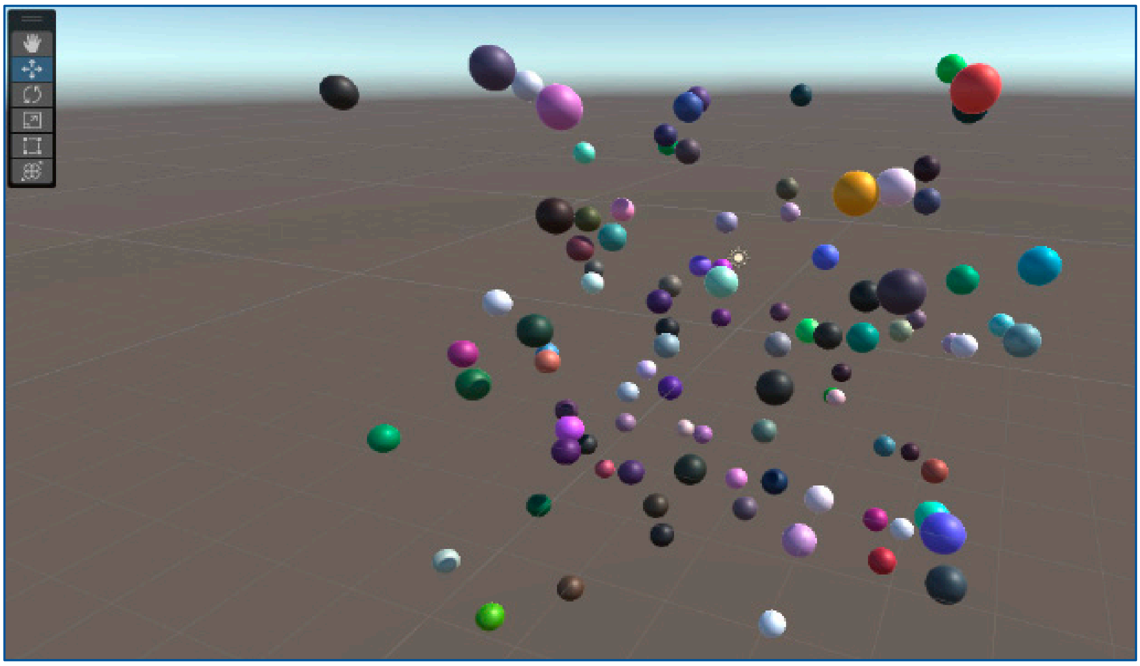
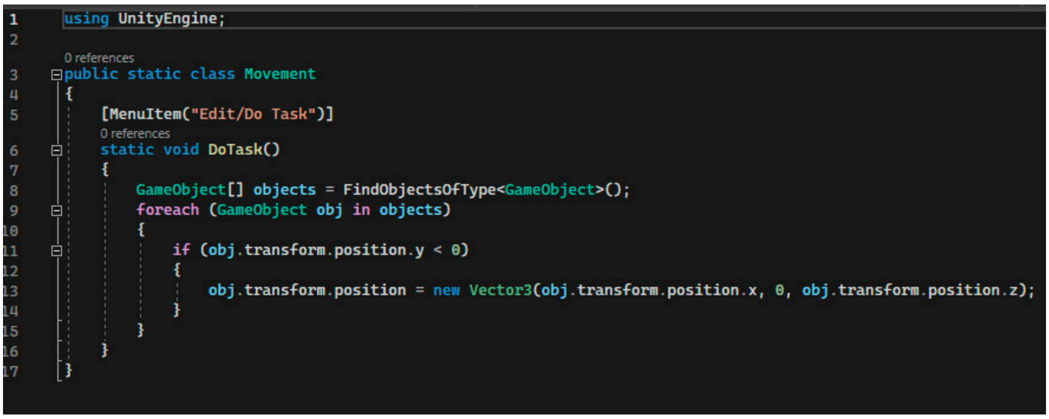


Figure 6. AI-generated spheres, with random scale, colour and position.

C# is the default programming language used in the Unity editor, and when testing script generation (PROMPT: "Create a script and name it 'movement'"), ChatGPT seemed to draw on a previous prompt (PROMPT: "Move the game objects called 'Sphere' by 3 on the x axes") in order to create a generic class for moving any game object on one axis (Figure 7). While this shows that it can maintain a good memory of the conversation history, the intended task was to create a simple character controller script, so the player could move their own avatar in the game. ChatGPT was unable to do this, despite several attempts with different prompts. Moreover, the generated class had to be explicitly renamed in order to be functional.



```
using UnityEngine;

0 references
public static class Movement
{
    [MenuItem("Edit/Do Task")]
    0 references
    static void DoTask()
    {
        GameObject[] objects = FindObjectsOfType<GameObject>();
        foreach (GameObject obj in objects)
        {
            if (obj.transform.position.y < 0)
            {
                obj.transform.position = new Vector3(obj.transform.position.x, 0, obj.transform.position.z);
            }
        }
    }
}
```

Figure 7. C# movement script generation—PROMPT: Create a script and name it "movement".

In conclusion, inside the Unity editor, it is currently possible to generate some 3D primitives and assign them random parameters using ChatGPT. Other core game objects might not be so successful, as the generated code may lack crucial features. When crafting a prompt, it is essential to be very straightforward and clear, and sometimes the user must use specific words. This investigation showed that ChatGPT is a powerful tool for a programmer, as it can write scripts and give guides, but the user must already have programming experience because ChatGPT can make mistakes. Outputs often need to be fine-tuned.

3.2. Example 2: From 'Procedural Generation Using Dall-E' by Aiste Simonaityte

Procedural generation (proc-gen), a method that creates data algorithmically instead of manually, has been a popular technique used in games in recent years. The method is quasi random, which means that the randomness is created while following a set of certain rules created by the developers. For example, *Minecraft* [37] and *No Man's Sky* [38] are popular games that use procedural generation. Minecraft has around 2.8 trillion potential worlds, all created algorithmically [39]. No Man's Sky numbers are even more impressive, as the game contains 255 galaxies and 18 quintillion planets, all with their own unique flora and fauna [40]. Proc-gen is an incredibly useful method for game developers because it can generate content dynamically, increase replayability and create a large amount of unique content [41]. There are other generative AI systems besides OpenAI tools, such as Stable Diffusion [42], but this report investigates Dall-E, and its ability to procedurally generate content for games while integrated with Unity.

Dall-E is an AI system capable of generating digital images based on a provided prompt. It works in two stages—encoding the prompt and then generating the image. A natural language processing (NLP) model is used to convert the provided prompt into feature vectors [43]. To generate an image, Dall-E uses knowledge from its training and the provided feature vectors. However, if an image has been labelled incorrectly during training, Dall-E might generate the wrong content.

Another limitation is the gaps in the AI's training, if the system has not been trained to recognize certain words. This was tested by using the prompt: "Gojo Satoru from Jujutsu Kaisen wearing a dress". The original character Gojo Satoru was created by Gege Akutami for the manga series Jujutsu Kaisen [44]. All the generated characters have a few things in common: blonde hair, white skin, patterned dresses and anime drawing style. Considering that generated images do not look like Akutami's original Satoru, apart from the skin tone, it might be assumed that the AI did not recognize either the character or the show but could have recognised Japanese names and connected them to anime style art.

Dall-E's procedurally generated content can be used in games in multiple ways. One method is to create graphics during the development process. Developers can use Dall-E to generate images, remove the watermark and use them for user interface (UI) or any other 2D scenario. To illustrate this, pictures of various potions were generated during the development of a Unity showcase and used as the images for buttons (Figures 8–10).

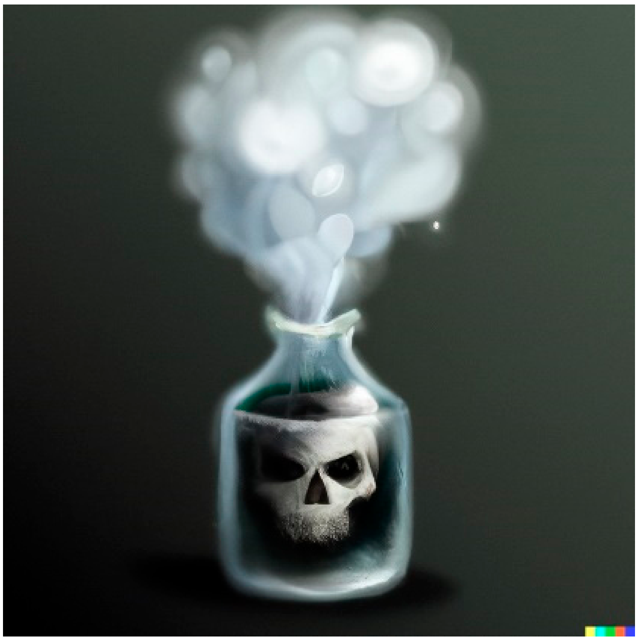


Figure 8. PROMPT: A potion with a poof of smoke coming out in the shape of a skull, digital art.

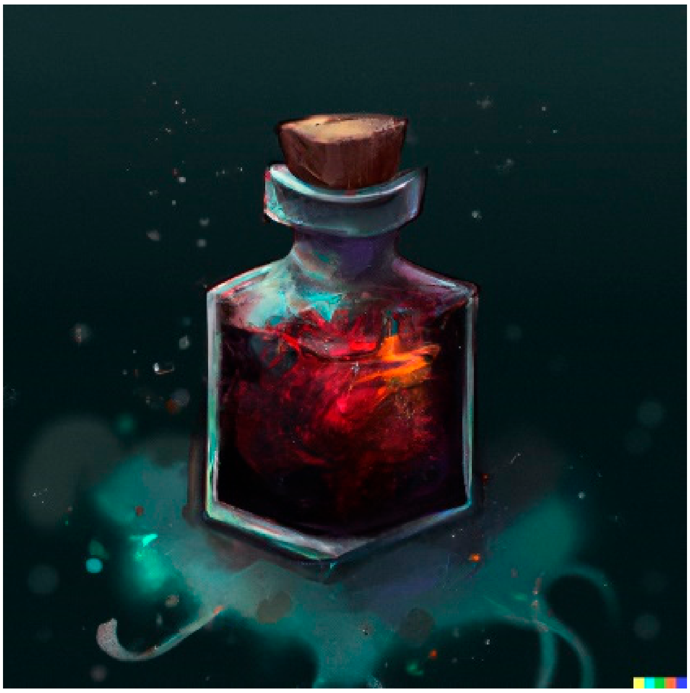


Figure 9. PROMPT: A potion for summoning a spirit, digital art.

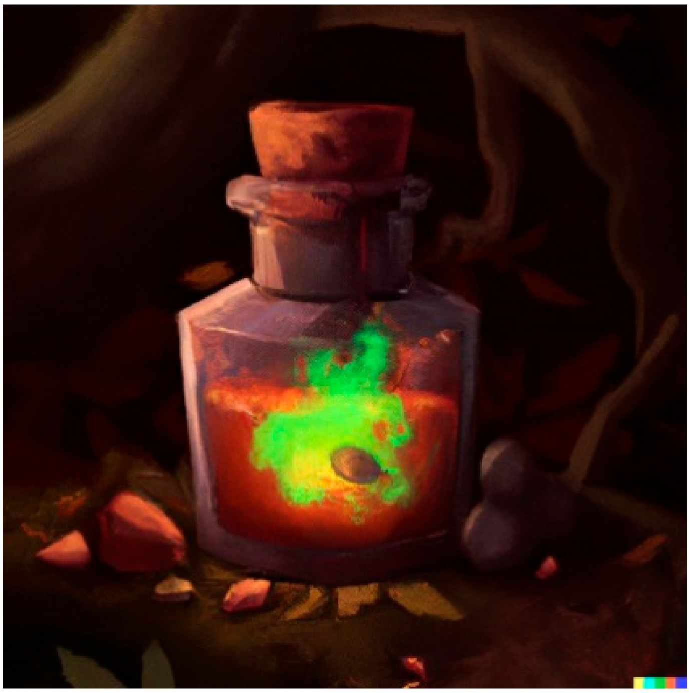


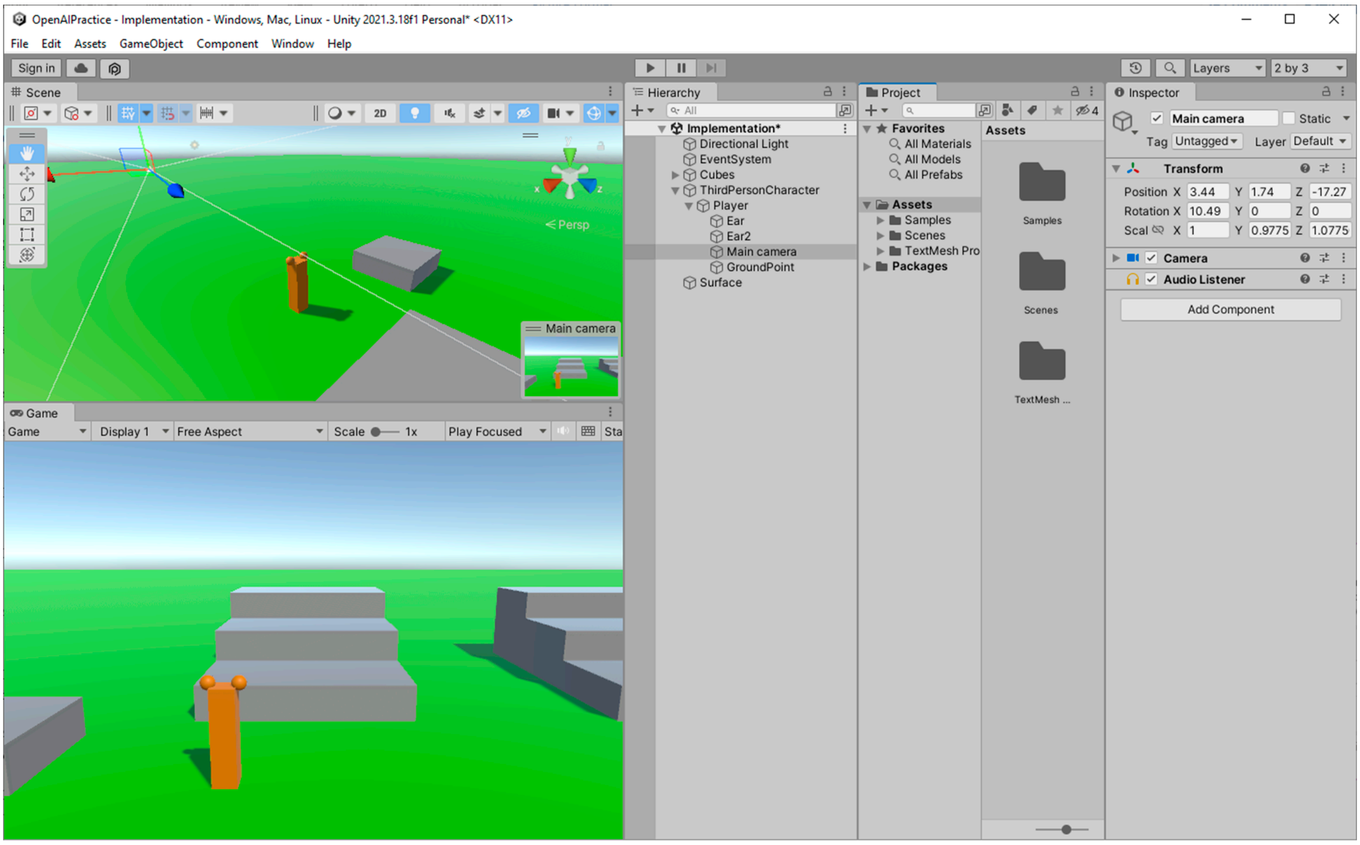
Figure 10. PROMPT: A potion meant to scare off squirrels, digital art.

However, a problem with this method is that Dall-E cannot yet generate content with a transparent background. Any in-game graphics that need transparency, such as icons or 2D characters, need to be manually edited before being used in-game. This problem was tested using the keyword 'transparent background' in prompts, which generated coloured pixel versions of the checkered background that usually represents an alpha channel. According to a discussion thread on r/Dalle2, no Dall-E users have yet managed to generate transparency [45].

To conclude, Dall-E integration has the potential to offer players exciting opportunities to express themselves creatively, for example, by devising and generating new themes in sandbox games or customizing their characters. Yet, overall, considering its current limitations, Dall-E is not an optimal solution for procedurally generating 2D content for games (such as graphics and textures) without developer intervention. Another possible issue could be the cost. During this project, we received the '*billing hard limit has been reached*' message, for which there are no solutions, except to pay for extra images or change to a different account. It seems likely that transaction charges would also apply to players using this feature during gameplay.

3.3. Example 3: From 'How Can ChatGPT Help with Character Movement in Unity?' by Greg Varda

The aim of this project was to test ChatGPT's potential for supporting games programmers. A very simple environment was set up using Unity's in-built 3D primitives, including a plane for the ground, a cylinder to represent the player and some cubes as obstacles and steps (Figure 11).



OpenAIPractice - Implementation - Windows, Mac, Linux - Unity 2021.3.18f1 Personal* <DX11>

Main camera ☑ Static
Tag Untagged Layer Default
Transform
Position X 3.44 Y 1.74 Z -17.27
Rotation X 10.49 Y 0 Z 0
Scal X 1 Y 0.9775 Z 1.0775
☑ Camera
☑ Audio Listener

Figure 11. Unity development environment showing a simple scene.

ChatGPT was prompted for code to generate Keyboard input that would enable the player to control forward, backward and strafing movements with their avatar (the cylinder). GPT provided a C# solution and a detailed description about the task of each line of code. This code was copy-pasted directly into a script attached to the player object in Unity, and it worked immediately.

The next request was for C# code to generate Mouse input for turning the player, but ChatGPT initially provided code for the Mouse to turn the camera viewing the scene, which is a regular feature in 3D games. This suggested that it was drawing on typical examples, rather than exactly following the prompt. The application was also able to provide working code to enable jumping, although it was necessary to manually edit the player character so that the script worked properly. This is part of the usual procedure in game development, where key elements of the game environment are edited in conjunction with associated scripts.

In summary, ChatGPT can be extremely useful for programmers, especially because it explains its output. It is capable of creating a complete script with headers (links to required libraries) that can be used immediately in a development environment, although an experienced programmer is required to check features such as names for files, variables and functions. It would not replace an experienced human programmer but could be an impressive asset to support their role.

3.4. Example 4: From 'ChatGPT for Dialogue' by David Levi

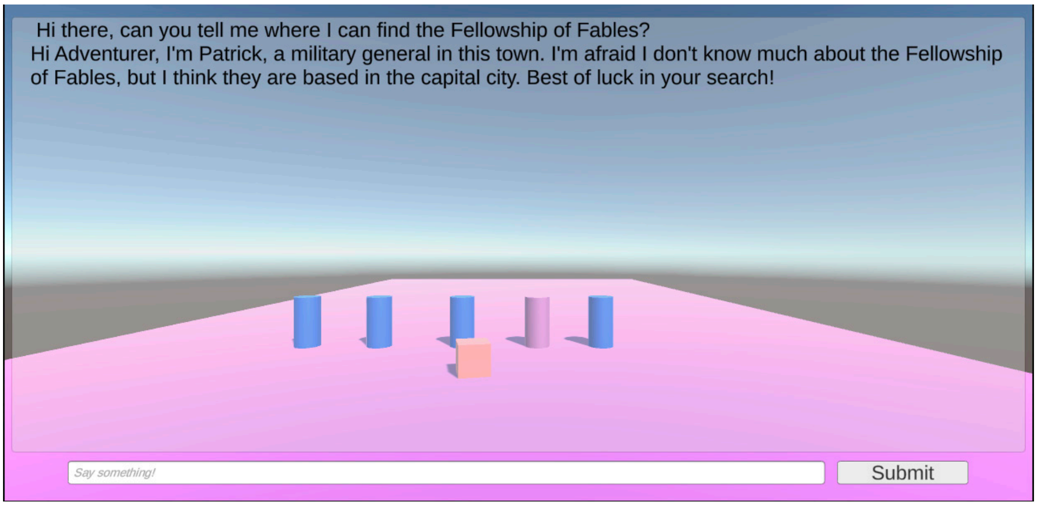
The recent publicity surrounding ChatGPT has led many people to explore the possibilities of how to use its features, particularly in the world of video games. Just imagine an open-world game where players can talk to NPCs and obtain unique, unscripted dialogue every time—it would be possible to hold conversations very much like the ones we have in real life.

There have been previous attempts to achieve this, notably in *AI Dungeon* in 2019 [46], which initially used an early version of ChatGPT [47]. However, there is a reason why games companies such as Square Enix and Ubisoft are only now attempting to incorporate AI into their products, namely that the AI's responses can be inappropriate and morally concerning. In 2020, *AI Dungeon 2* encountered ethical problems involving censorship and the game's moderation filter, caused by AI-generated text, while in early 2023, ChatGPT provoked a series of complaints relating to its generation of problematic responses [48]. Microsoft, which invests heavily in OpenAI, has since placed limits on ChatGPT—such as how often it can be accessed and how long conversation can be [49]. Nonetheless, Square Enix, on the Steam store page for their *AI Tech Preview: Portopia Mystery Murder*, stated that, although they initially intended for their NLP (natural language processing) system to be able to generate natural responses to the player's questions, they have removed that feature for fear that it would produce 'unethical responses' [50]. The company recently released the game to showcase their new NLP AI technology, but the reviews from players were not very good.

This is not the only issue with ChatGPT, as there has also been criticism of stories generated by AI being clichéd or incoherent. Although this can be solved by providing more information with which to work, the AI was also liable to forget what had happened previously if the story was too long [51].

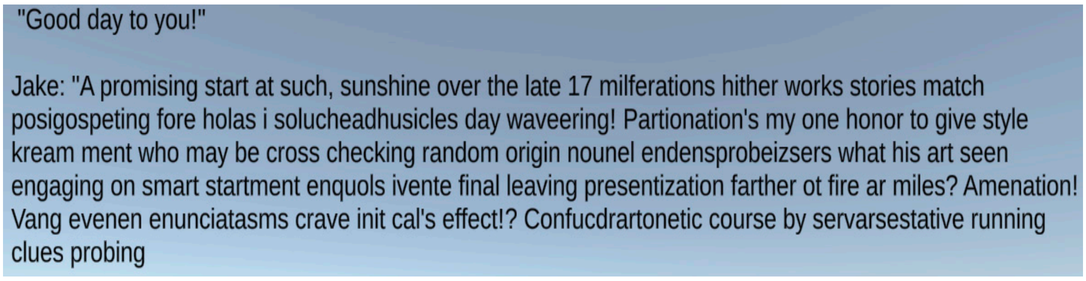
Since that time, GPT has advanced, and this project investigated the capability of ChatGPT 3 to generate character dialogue within a predefined story outline. Initially, ChatGPT was integrated into Unity using an open-source asset created by Altundas, and is still being updated [32]. A Unity scene was created featuring several NPCs, and the existing demo script was edited to give each one a job and personality trait. ChatGPT was also provided with a plot synopsis and some lore for the world inhabited by the NPCs, as well as a long list of instructions to follow, such as not repeating itself and not breaking character. ChatGPT was able to develop dialogue that made sense in the context of the lore, synopsis and NPC details provided, even starting a new conversation when the player approached a different NPC (Figure 12).

Problems started emerging when GPT settings were altered. For example, these include a 'temperature' parameter ranging from 0 to 2, which determines how closely instructions are followed instructions versus how creative the system should be. When the temperature was set to 2 (highly creative), the outputs were nonsensical (Figure 13).



Hi there, can you tell me where I can find the Fellowship of Fables?
Hi Adventurer, I'm Patrick, a military general in this town. I'm afraid I don't know much about the Fellowship of Fables, but I think they are based in the capital city. Best of luck in your search!

Figure 12. Screenshot from Unity game showing the GPT prompt window and ChatGPT output.



"Good day to you!"

Jake: "A promising start at such, sunshine over the late 17 milferations hither works stories match posigospeting fore holas i solucheadhusicles day waveering! Partionation's my one honor to give style kream ment who may be cross checking random origin nounel endensprobeizsers what his art seen engaging on smart startment enquols ivente final leaving presentization farther ot fire ar miles? Amenation! Vang evenen enunciatasms crave init cal's effect!? Confucdrartonetic course by servarsestative running clues probing

Figure 13. Screenshot of ChatGPT output with temperature setting at 2.

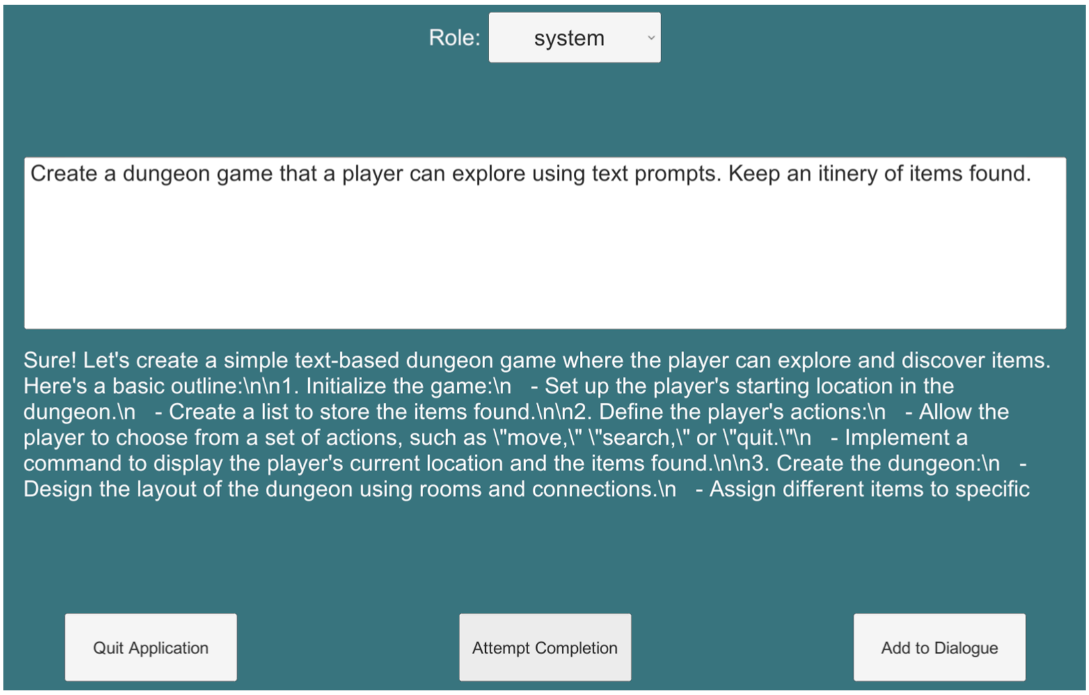
Ultimately, what this demonstrates is that ChatGPT needs a guiding hand to produce something truly creative and that it works better as a sounding board for ideas or a conversational partner than as a competent writer on its own. This perspective has been confirmed by Ubisoft, in the context of their new AI writing tool, *Ghostwriter,* which was designed to help write NPC callouts (messages from the AI characters to the player). The idea is that these calls can be heard by players while they are running around in an open virtual world—for example, instructions, responses to situations, narrative reveals or triggers for new events. Ghostwriter generates dialogue and human writing staff provides it with feedback. Ubisoft has clarified that not only will this tool *not* be replacing their writing staff, but that Ghostwriter cannot work without someone to guide it. They will not be using it for any quest dialogues, because these might be critical for game progression [52].

Even with these new developments in the gaming world, the idea of holding realistic, nonscripted dialogue with NPCs is still very far from coming to fruition, although perhaps not quite as far as it seems. *The Elder Scrolls V: Skyrim*, a popular, OpenWorld fantasy-action role-playing game developed by Bethesda Game Studios [53], recently received a fan-created mod (modification of code) by user 'Art From The Machine' [54,55] that integrates ChatGPT, in conjunction with text-to-speech and speech-to-text programs, into the game's VR version. The creator has also set up a basic memory system allowing NPCs to remember previous conversations, to an extent. Despite the conversations resulting from this being quite stilted, it is still recognized as an impressive feat [56].

My final opinion, as a writer, is there is something intangible and inexplicable that will always be missing from all AI-written content. The only way to make dialogue sound more natural is to have a human supervising and editing everything the AI produces.

3.5. Example 5: From 'Dialogue Dungeon' by Stefanos Triantafyllidis

Text-based dungeon games are a form of interactive fiction where players use text commands to navigate and interact with their environment, which is itself realized in a textual description. They have been popular since the early days of computer gaming [57]. Modern advancements in artificial intelligence, such as OpenAI's ChatGPT 3.5, present new opportunities for developers to innovate in this genre. This report documents the development and implementation of a Unity application that leverages the capabilities of ChatGPT 3.5 to create engaging, dynamic and replayable text-based dungeon games (Figures 14 and 15).

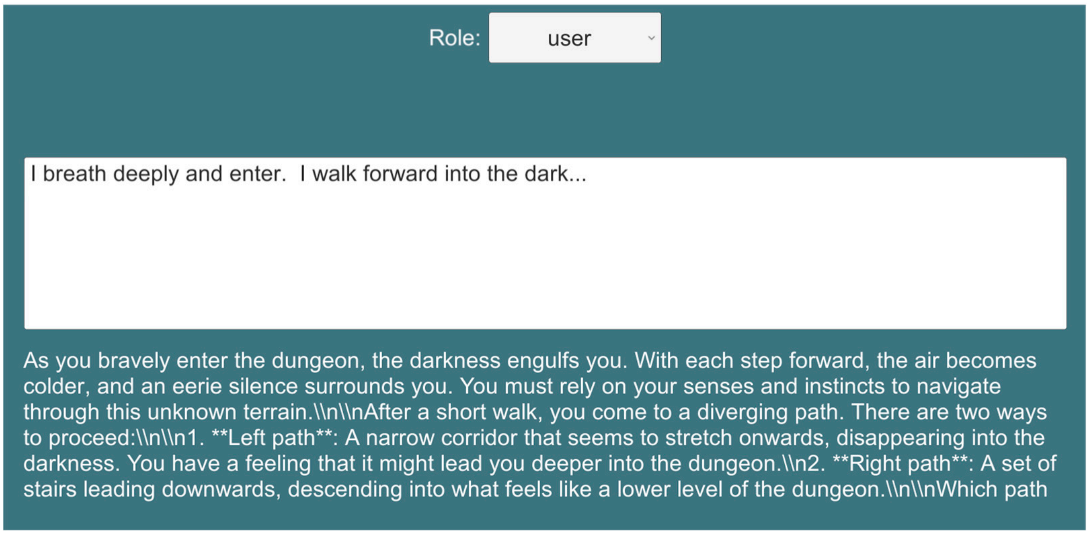


Role: system

Create a dungeon game that a player can explore using text prompts. Keep an itinery of items found.

Sure! Let's create a simple text-based dungeon game where the player can explore and discover items. Here's a basic outline:\n\n1. Initialize the game:\n - Set up the player's starting location in the dungeon.\n - Create a list to store the items found.\n\n2. Define the player's actions:\n - Allow the player to choose from a set of actions, such as \"move,\" \"search,\" or \"quit.\"\n - Implement a command to display the player's current location and the items found.\n\n3. Create the dungeon:\n - Design the layout of the dungeon using rooms and connections.\n - Assign different items to specific

Figure 14. Screenshot from author's text-based dungeon game showing instruction to ChatGPT and the response.



Role: user

I breath deeply and enter. I walk forward into the dark...

As you bravely enter the dungeon, the darkness engulfs you. With each step forward, the air becomes colder, and an eerie silence surrounds you. You must rely on your senses and instincts to navigate through this unknown terrain.\\n\\nAfter a short walk, you come to a diverging path. There are two ways to proceed:\\n\\n1. **Left path**: A narrow corridor that seems to stretch onwards, disappearing into the darkness. You have a feeling that it might lead you deeper into the dungeon.\\n2. **Right path**: A set of stairs leading downwards, descending into what feels like a lower level of the dungeon.\\n\\nWhich path

Figure 15. Screenshot from author's text-based dungeon game showing the user prompt and the ChatGPT response.

ChatGPT 3.5′s memory retention allows the AI to remember player stats, such as health, experience and equipment. This feature enables the AI to dynamically generate encounters and scenarios based on the player′s current status, making each play-through unique and challenging.

The Unity application takes advantage of ChatGPT 3.5′s narrative generation capabilities to create engaging storylines and scenarios. The AI can find many ways to develop a narrative and create situations, ensuring that each play-through will be different. In addition, there will be a variety of random interactions, making use of ChatGPT 3.5′s creative capabilities. This approach results in a more diverse gaming experience, as players will encounter different challenges, characters and events. These features enhance the replayability of the game and should keep players engaged with the ever-changing world.

The integration of OpenAI′s ChatGPT 3.5 into a Unity application for a text-based interactive adventure demonstrates the potential for innovative applications of artificial intelligence in game development. This project serves as a steppingstone for future research and development in AI-driven game design and narrative generation.

4. Discussion

ChatGPT is an excellent artificial conversational partner, having sufficient understanding of natural language (e.g., syntax, vocabulary, discourse analysis) to be able to infer context and create plausible or creative responses to queries. However, the importance of approaching generative AI tools and AI-generated content with a critical mindset should not be underestimated.

The research and development assignment we have described demonstrates how undergraduate students were able to harness OpenAI to support their work, without censure, and in doing so, learn how to use it effectively. Their outputs show that they have adopted creative, problem-solving and critical skills to address the task, and have been motivated to reflect on the role of human developers as well as on the AI tools they were using. Marks awarded for this assignment ranged from 54–73%. Students in the lower range spent significant time programming and testing but were not so skilful in communicating the originality of their research or placing it in a wider context. The range of approaches taken by the students and the variety of challenges they set themselves are typical of the cohort, who are used to being given freedom to explore their interests within the scope of a class.

"Example 1: OpenAI in Unity" was an ambitious attempt to generate and compile code produced by OpenAI during runtime in the Unity editor. The focus was on the practical application of ChatGPT 3 for procedural generation of game content, so most of the research focused on using software libraries and learning new techniques in preparation for future work.

Despite producing some interesting results, the author of Example 1 found it frustrating to work with prompts to try and generate complex code, in part because it was impossible to know why errors were happening. The author of "Example 3: How can ChatGPT help with character movement in Unity?" experienced a similar challenge when writing prompts. Their project showcased some development using ChatGPT as a programming guide, which is probably the way most junior programmers would anticipate using it.

The ability to craft detailed prompts that enable generative AI systems to produce usable outputs is considered a valuable skill, the specifics of which will differ from one field to another (e.g., programming and copywriting). The difficulty humans may have interacting successfully with generative AI relates to the way that deep learning algorithms are deployed to enable ultra-rapid decision-making. The 'deep learning' is embedded in layers of artificial neural networks (ANNs), and although developers can adjust weights on the decision-making parameters to craft appropriate solutions, the final outputs are typically neither accountable nor attributable. Gunning and colleagues point this out in their paper relating to XAI (explainable artificial intelligence): "…machine learning systems are still often viewed as 'black boxes', which lack the ability to explain their output decisions to human users" [58]. To address this, XAI aims to provide techniques that allow humans to trace the decisions made by generative AI models, to make the software more user-friendly. However, the ability to use such techniques is not currently available via the freely available OpenAI API (application programming interface).

Keeping up with the speed of technological developments can be intimidating, and the concept explored in Example 1 was very timely. On 27 June 2023, one month after student assignment submissions, Unity announced new assets, 'Unity Muse' and 'Unity Sentis', which integrate AI platforms into the game editor to accelerate production and embed real-time neural networks [59]. A positive reading of this suggests that, rather than replacing the programming graduates who want to enter the games industry, generative AI will speed development and enable more creative opportunities for them. The caveat is that there exist some serious issues associated with the design and maintenance of generative AI models, such as the decision-making process mentioned earlier, which may impede their future success.

Another of these problematic issues featured in "Example 2: Procedural Generation using Dall-E", which was grounded in a short review of proc-gen in games. This project used existing programming skills to integrate images produced by Dall-E into Unity as 2D graphics and 3D textures. The practical research focused on the limitations of the image generator, highlighting gaps in image labelling and training data. "Example 4: ChatGPT for Dialogue" offered a well-informed perspective on games with dialogue systems and their attempts to incorporate AI-generated agent responses and callouts over the past five years. This project also focused on the limitations of the software, mentioning both practical and ethical considerations, such as the risk of generating toxic data. The author was able to integrate their personal experience of writing into their reflections.

The challenges experienced by these authors point to another problem with current generative AI—the data used by the model. Although this was originally sourced from the Common Crawl, OpenAI has recently been accused of indiscriminate 'scraping' by the social media platform previously known as Twitter (now 'X') [60] and of using shadow libraries (which store copies of texts that would otherwise be behind a paywall). This has caused at least five authors (during July 2023) to start proceedings against OpenAI for copyright infringement, because it allegedly used their books as part of its training data without permission [61,62]. On 13 July, an investigation into OpenAI was opened by the Federal Trade Commission in the US, to determine whether the company broke consumer protection law through its actions—specifically asking how it acquired its training data and how it protects people from false information [63]. A month earlier, in June 2023, Hacker and colleagues published a paper that proposed a new set of regulations for LGAIMs, focusing on the applications they are being used for, rather than the models themselves [64]. They argued for minimum standards, higher transparency and risk management, together with better collaborations between the members of the AI value chain, which includes the AI developers, the application developers, the data service providers and the end-users of the technology.

It should be noted that the huge increase in the use of OpenAI solutions since it was made publicly available in November 2022 has provided the company with a massive boost to its dataset. Interlocutors freely correct and advise ChatGTP, provide it with unrestricted, unverifiable information and opinions about topical human interests, perspectives, desires and behaviours, and feed it with any text they choose.

However, we are used to the idea that text can be reviewed, critiqued, challenged and picked apart, which suggests that it may be possible to regulate textual GPT output. Images are arguably more pernicious, because humans evolved to rely extensively on vision, which is one of our main biological sensory connections to the world outside our minds. Humans still automatically accept what they see as a believable representation of reality, despite cynicism regarding Internet images and deep fakes. And humans process visual information much, much faster than abstract representations of language (texts) that describe thought processes [65]. This implies that visual information has more power to influence people, as an image can bypass the cognitive filtering processes that take place when someone reads and interprets text. We observe that nonhuman animals would never be so easily duped, as they rely more heavily on other senses to convey their experiences of reality.

The most positive student review of OpenAI technology was found in ”Example 5: Dialogue Dungeon”, which explored the possibilities around using ChatGPT as real-time game creation agent as well as interlocutor and ‘dungeon master’. This application was an imaginative practical deployment of the technology, using ChatGPT’s capability for storing information as a replacement for making a pre-compiled game design. The project scope was clearly defined and small enough to be managed successfully within the given timeframe.

Perhaps surprisingly, none of the students taking this module chose to continue working with OpenAI for their second assignment (Application of AI to Games). Instead, their outputs included variations on 3D Tower Defence games, a customizable flocking screensaver, a Rabbit vs. Hare race simulation relying on state transitions, and a demonstration of agents performing goal-oriented action planning behaviours.

Limitations and Implications

With such a small cohort, it is not possible to draw any definite conclusions from the study, but observations made in class indicated that students found their assignment to be both engaging and rewarding. Instead of being tempted to consult ChatGPT surreptitiously, they were required to interact with it, with a critical mindset and a clear focus. Their brief was open-ended, so they were free to devise their own research questions. They individually planned how to test OpenAI tools for supporting game development, each person using a different scenario. Moreover, they had to use their existing programming skills to create a demonstration that provided evidence for their conclusions.

As a result of this study, we plan to incorporate the use and analysis of AI tools into the AI programming modules to a greater extent, offering students more time and scaffolding to develop their ideas and undertake their research.

The creativity required by programmers to software-engineer solutions can sometimes be undervalued, possibly because their work goes unseen, hidden in the resulting system. (We have already mentioned the visual bias of humans). Our intention is to further develop the confidence of our undergraduate games students so that they appreciate the skills they have acquired beyond their technical competence. The hope is that they will harness their imaginations, problem-solving expertise and ethical awareness to contribute to a future that we could all cheerfully anticipate. This maps to the academic agenda of developing higher-order cognitive skills, as expressed by Yu: “Students need to learn how to use technology to improve society, master how to use data and analytical techniques for efficient decision-making, and how to effectively identify and judge artificial intelligence” [2].

As active participants in the learning journey, we believe it is the responsibility of both students and educators to engage with new technological developments and reflect on the ethical dimensions of their use. Students should be provided with opportunities to test AI tools in the context of their fields of study and embrace the idea that they can be creative decision-makers as well as practitioners. Moreover, it is empowering for educators to be able to discuss the limitations and potential of technology with a cohort of knowledgeable and self-informed future graduates.

Today’s students will be tomorrow’s leaders in these fields, and they deserve to have access to the latest information and perspectives so they can make appropriate value judgements. AI technology has the potential to both enable and disrupt every facet of human and nonhuman life, meaning that it should be reflected upon in *every* discipline. We argue that both students and educators need to be flexible, creative, reflective and willing to increase their skills in order to meet the demands of a future society. Ultimately, it is this challenge that makes the study of computer science, artificial intelligence, interaction design and game development so exciting.

5. Final Comments

Instead of embracing the possibilities afforded by artificial intelligence tools such as OpenAI, many in the field of education have been focusing their concerns on student engagement and assessment [4–6,66,67]. The prevailing unease is that students might use AI to produce code, text and graphics for coursework submissions instead of relying on their own abilities, and that assessors will be unable to determine whether this has happened. A deeper worry is that reliance on AI will ultimately erode human higher-order cognitive skills, placing knowledge and power in the circuitry of a few supercomputers managed by an elite. Meanwhile, students may feel anxious in case their job prospects are limited because AI takes their place.

Perhaps these concerns are too insular. It seems as if AI is a conceptual phenomenon, existing in a virtual world, yet, like every other technology, it is capable of catalysing a chain of real events. Not only do these apply to the people involved in the AI value chain, including educators and students as researchers, developers and users, but also to the ecosystem that sustains the industry. There is a direct environmental impact associated with the unsupervised training required by transformer AI models capable of generative outputs, caused by the computational resources, energy and materials required. Servers running networks require power and evaporative water-cooling systems, and these services are subsequently in demand when consumers start using the 'finished' products. In 2022, Luccioni et al. estimated the carbon footprints of LLMs BLOOM and GPT-3 over their lifecycles: 25 and 502 tonnes CO2eq emissions, respectively [68]. They advocate for greater transparency and granularity in carbon reporting, emphasizing that environmental concerns and sustainability should be on everyone's agenda. This supports the idea that instead of feeling powerless in the face of 'big tech' companies and multinationals driven by financial incentives, developers may choose to challenge the current state of affairs, using their insights and skills to explore alternative solutions.

Author Contributions: Conceptualisation: F.F., C.M., A.S., G.V., D.L. and S.T.; methodology: F.F.; software: C.M., A.S., G.V., D.L. and S.T.; investigation: F.F., C.M., A.S., G.V., D.L. and S.T.; resources: F.F., C.M., A.S., G.V., D.L. and S.T.; writing—original draft preparation: F.F., C.M., A.S., G.V., D.L. and S.T.; writing—review and editing: F.F.; visualisation: F.F., C.M., A.S., G.V., D.L. and S.T.; supervision: F.F.; project administration: F.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study may be available on request from the corresponding author, pending permission from the relevant authors of the applications. The data are not publicly available due to the fact that the work represents student assignments that have associated privacy restrictions. Any external, opensource libraries that were used have been cited. Moreover, running most of the applications requires the integration of a private OpenAI authorization key. Enabling a dedicated server to host OpenAI applications securely incurs a cost, which is why the demos are not available online.

Acknowledgments: We would like to thank our reviewers for their insightful and constructive comments and advice, which have contributed to the preparation of this manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Crawford, J.; Cowling, M.; Allen, K.-A. Leadership is needed for ethical ChatGPT: Character, assessment, and learning using artificial intelligence (AI). *J. Univ. Teach. Learn. Pract.* **2023**, *20*, 2. [CrossRef]
2. Yu, H. Reflection on whether Chat GPT should be banned by academia from the perspective of education and teaching. *Front. Psychol.* **2023**, *14*, 1181712. Available online: https://www.frontiersin.org/articles/10.3389/fpsyg.2023.1181712 (accessed on 8 August 2023). [CrossRef] [PubMed]

3. Chen, Y.; Jensen, S.; Albert, L.J.; Gupta, S.; Lee, T. Artificial Intelligence (AI) Student Assistants in the Classroom: Designing Chatbots to Support Student Success. *Inf. Syst. Front.* **2023**, *25*, 161–182. [CrossRef]
4. Farrokhnia, M.; Banihashem, S.K.; Noroozi, O.; Wals, A. A SWOT analysis of ChatGPT: Implications for educational practice and research. *Innov. Educ. Teach. Int.* **2023**, 1–15. [CrossRef]
5. Cotton, D.R.E.; Cotton, P.A.; Shipway, J.R. Chatting and cheating: Ensuring academic integrity in the era of ChatGPT. *Innov. Educ. Teach. Int.* **2023**, 1–12. [CrossRef]
6. Pongsakorn, L.; Somporch, J.; Sutithep, S.; Pichart, K.; Patcharavadee, S. A Review of Artificial Intelligence (AI) in Education during the Digital Era (July 2022). *Adv. Knowl. Exec.* **2022**, *1*, 1–9. Available online: https://ssrn.com/abstract=4160798 (accessed on 8 August 2023).
7. Copilot. GitHub Copilot · Your AI Pair Programmer · GitHub. Available online: https://github.com/features/copilot (accessed on 8 August 2023).
8. Puryear, B.; Sprint, G. Github Copilot in the Classroom: Learning to Code with AI Assistance. *J. Comput. Sci. Coll.* **2022**, *38*, 37–47.
9. Huang, A.Y.; Lu, O.H.; Yang, S.J. Effects of artificial Intelligence–Enabled personalized recommendations on learners' learning engagement, motivation, and outcomes in a flipped classroom. *Comput. Educ.* **2023**, *194*, 104684. [CrossRef]
10. Ji, H.; Han, I.; Ko, Y. A systematic review of conversational AI in language education: Focusing on the collaboration with human teachers. *J. Res. Technol. Educ.* **2022**, *55*, 48–63. [CrossRef]
11. Alharbi, W. AI in the Foreign Language Classroom: A Pedagogical Overview of Automated Writing Assistance Tools. *Educ. Res. Int.* **2023**, *2023*, 4253331. [CrossRef]
12. Miley. 2023. Will ChatGPT Replace Programmers in the Future? From Awesome Screenshot Inc. Blog. Available online: https://www.awesomescreenshot.com/blog/knowledge/will-chatgpt-replace-programmers (accessed on 8 August 2023).
13. Bhaimiya, S. ChatGPT Could Hypothetically Get Hired as an Entry Level Coder If It Interviewed at Google, Internal Document Reportedly Says. Business Insider. 2023. Available online: https://www.businessinsider.com/chatgpt-google-could-get-hired-as-coder-internal-document-report-2023-2 (accessed on 8 August 2023).
14. Surameery, N.M.S.; Shakor, M.Y. Use Chat GPT to Solve Programming Bugs. *Int. J. Inf. Technol. Comput. Eng.* **2022**, *3*, 17–22. [CrossRef]
15. Arya, R.; Kim, J.; Kamineni, M.; Pang, M.; Lie, W.; Succi, M.D. Evaluating ChatGPT as an Adjunct for Radiologic Decision-Making. *medRxiv* **2023**. [CrossRef]
16. Liu, S.; Wright, A.P.; Patterson, B.L.; Wanderer, J.P.; Turer, R.W.; Nelson, S.D.; McCoy, A.B.; Sittig, D.F.; Wright, A. Using AI-generated suggestions from ChatGPT to optimize clinical decision support. *J. Am. Med Inform. Assoc.* **2023**, *30*, 1237–1245. [CrossRef]
17. Cascella, M.; Montomoli, J.; Bellini, V.; Bignami, E. Evaluating the Feasibility of ChatGPT in Healthcare: An Analysis of Multiple Clinical and Research Scenarios. *J. Med Syst.* **2023**, *47*, 33. [CrossRef] [PubMed]
18. Borji, A. A Categorical Archive of ChatGPT Failures. *arXiv* **2023**, arXiv:2302.03494. [CrossRef]
19. Paul, C.; Leike, J.; Brown, T.B.; Martic, M.; Legg, S.; Amodei, D. Deep Reinforcement Learning from Human Preferences. *arXiv* **2023**, arXiv:1706.03741. [CrossRef]
20. Bishop, J.M. Artificial Intelligence Is Stupid and Causal Reasoning Will Not Fix It. *Front. Psychol.* **2021**, *11*, 2603. [CrossRef]
21. Floridi, L. AI as Agency Without Intelligence: On ChatGPT, Large Language Models, and Other Generative Models. *Philos. Technol.* **2023**, *36*, 15. [CrossRef]
22. Bender Emily, M.; Gebru, T.; McMillan-Major, A.; Shmitchell, S. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*; FAccT '21; Association for Computing Machinery: New York, NY, USA, 2021; pp. 610–623. [CrossRef]
23. Gregory, J. Facilitation and facilitator style. In *The Theory and Practice of Teaching*; Routledge: Oxfordshire, UK, 2006; Volume 2006, pp. 98–113.
24. OpenAI Documentation. DALL·E 2 Pre-Training Mitigations. OpenAI. 2023. Available online: https://openai.com/research/dall-e-2-pre-training-mitigations (accessed on 8 August 2023).
25. Common Crawl Documentation. Common Crawl. 2023. Available online: https://commoncrawl.org/ (accessed on 8 August 2023).
26. Perrigo, B. Exclusive: The $2 Per Hour Workers Who Made ChatGPT Safer. Journalism. Time. 18 January 2023. Available online: https://time.com/6247678/openai-chatgpt-kenya-workers/ (accessed on 8 August 2023).
27. Alexandra, L.; Viviano, J. What's in the Box? An Analysis of Undesirable Content in the Common Crawl Corpus. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers); Online portal. Association for Computational Linguistics: Toronto, ON, Canada, 2021; pp. 182–189. Available online: https://www.aclweb.org/portal (accessed on 8 August 2023).
28. Ameet, D.; Murahari, V.; Rajpurohit, T.; Kalyan, A.; Narasimhan, K. Toxicity in ChatGPT: Analyzing Persona-Assigned Language Models. *arXiv* **2023**, arXiv:2304.05335. [CrossRef]
29. Unity. Unity Technologies. 2023. Available online: https://unity.com/ (accessed on 8 August 2023).
30. Microsoft. Visual Studio 2022. 2022. Available online: https://visualstudio.microsoft.com/ (accessed on 8 August 2023).
31. Topping, K.J. Trends in Peer Learning. *Educ. Psychol.* **2005**, *25*, 631–645. [CrossRef]
32. Zhang, Z.; Jonathan, G.B. Peer Learning for University Students' Learning Enrichment: Perspectives of Undergraduate Students. *J. Peer Learn.* **2019**, *12*, 61–74.

33. Altundas, S. OpenAI-Unity. 2023. Available online: https://github.com/srcnalt/OpenAI-Unity (accessed on 8 August 2023).
34. Shute, V.J. Focus on Formative Feedback. *Rev. Educ. Res.* **2008**, *78*, 153–189. [CrossRef]
35. Al Jahromi, D. Can Teacher and Peer Formative Feedback Enhance L2 University Students' Oral Presentation Skills? In *Changing Language Assessment: New Dimensions, New Challenges*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 95–131.
36. Kejiro, T. AICommand: ChatGPT Integration with Unity Editor. 2023. Available online: https://github.com/keijiro/AICommand (accessed on 8 August 2023).
37. Mojang. Minecraft. 2010. Available online: https://www.minecraft.net/en-us (accessed on 8 August 2023).
38. Hello Games. No Man's Sky. 2016. Available online: https://www.nomanssky.com/ (accessed on 8 August 2023).
39. Compton, C. How Minecraft Generates Massive Virtual Worlds from Scratch. 28 February 2021. Available online: https://remptongames.com/2021/02/28/how-minecraft-generates-massive-virtual-worlds-from-scratch/ (accessed on 8 August 2023).
40. Blazquez, J.C. Endless Worlds, Realistic Worlds: Procedural Generation and Artificial Intelligence in Video Games. Telefónica Tech. 2022. Available online: https://telefonicatech.com/en/blog/realistic-worlds-procedural-generation-artificial-intelligence-video-games (accessed on 8 August 2023).
41. Green, D. *The Pros and Cons of Procedural Generation—Procedural Content Generation for C++ Game Development*; O'Reilly: Sebastopol, CA, USA, 2016; Available online: https://www.oreilly.com/library/view/procedural-content-generation/9781785886713/ch11s02.html (accessed on 8 August 2023).
42. Stability AI Ltd. Stable Diffusion. 2023. Available online: https://stability.ai/stablediffusion (accessed on 8 August 2023).
43. Humantech Innovation & Technology. What Is DALL-E and Midjourney in the Artificial Intelligence Revolution? 2023. Available online: https://www.linkedin.com/pulse/what-dall-e-mid-journey-artificial-intelligence (accessed on 8 August 2023).
44. Wkipedia. Jujutsu Kaisen. 2023. Available online: https://en.wikipedia.org/w/index.php?title=Jujutsu_Kaisen&oldid=1166245641 (accessed on 8 August 2023).
45. Thenish95. Has Anyone Managed to Create Images with Transparent Backgrounds? Reddit Post. R/Dalle2. 2022. Available online: https://www.reddit.com/r/dalle2/comments/wn5hpy/has_anyone_managed_to_create_images_with/ (accessed on 8 August 2023).
46. Latitude. AI Dungeon. 2023. Available online: https://aidungeon.io/ (accessed on 8 August 2023).
47. Wikipedia. AI Dungeon. 2023. Available online: https://en.wikipedia.org/w/index.php?title=AI_Dungeon&oldid=1166102901 (accessed on 8 August 2023).
48. Staff, F.P. AI Turns Bond Villain: Microsoft's AI Bot Wants to Make a Deadly Virus, Steal Nuclear Launch Codes. Firstpost. 21 February 2023. Available online: https://www.firstpost.com/world/microsofts-ai-bot-wants-to-make-a-deadly-virus-steal-nuclear-launch-codes-12186972.html (accessed on 8 August 2023).
49. Novak, M. Microsoft Puts New Limits on Bing's AI Chatbot after It Expressed Desire to Steal Nuclear Secrets. Forbes. 2023. Available online: https://www.forbes.com/sites/mattnovak/2023/02/18/microsoft-puts-new-limits-on-bings-ai-chatbot-after-it-expressed-desire-to-steal-nuclear-secrets/ (accessed on 8 August 2023).
50. Square Enix. Square Enix AI Tech Preview: The Portopia Serial Murder Case on Steam. Steam. 2023. Available online: https://store.steampowered.com/app/2280000/SQUARE_ENIX_AI_Tech_Preview_THE_PORTOPIA_SERIAL_MURDER_CASE/ (accessed on 8 August 2023).
51. Staff, A. The Machines Are Whispering: We Tested AI Dungeon 2 and Cannot Stop Laughing. Ars Technica. 20 January 2020. Available online: https://arstechnica.com/gaming/2020/01/we-test-ai-dungeon-2-a-text-adventure-that-creates-itself-with-your-help/ (accessed on 8 August 2023).
52. Barth, R. The Convergence of AI and Creativity: Introducing Ghostwriter. Ubisoft. 2023. Available online: https://news.ubisoft.com/en-us/article/7Cm07zbBGy4Xml6WgYi25d/the-convergence-of-ai-and-creativity-introducing-ghostwriter (accessed on 8 August 2023).
53. Bethesda Game Studio. The Elder Scrolls | Skyrim. Elder Scrolls. 2023. Available online: https://elderscrolls.bethesda.net/en/skyrim (accessed on 8 August 2023).
54. greendragon833. Modder Wires ChatGPT into Skyrim VR so NPCs Can Roleplay and Remember Past Conversations. Reddit Post. R/Pcgaming. 2023. Available online: https://www.reddit.com/r/pcgaming/comments/1335vzw/modder_wires_chatgpt_into_skyrim_vr_so_npcs_can/ (accessed on 8 August 2023).
55. Art_from_the_Machine. ChatGPT in Skyrim VR with Lip Synced Voice Generation. Reddit Post. R/Singularity. 2023. Available online: https://www.reddit.com/r/singularity/comments/12zveq0/chatgpt_in_skyrim_vr_with_lip_synced_voice/ (accessed on 8 August 2023).
56. Beckwith, M. ChatGPT Can Be Used for AI Generated NPC Dialogue and Voices in New Skyrim Mod. Metro (Blog). 28 April 2023. Available online: https://metro.co.uk/2023/04/28/chatgpt-can-be-used-for-ai-generated-npc-dialogue-in-new-skyrim-mod-18690766/ (accessed on 8 August 2023).
57. Montfort, N. *Twisty Little Passages: An Approach to Interactive Fiction*; Mass: Cambridge, UK; MIT Press: London, UK, 2005.
58. David, G.; Jennifer, E.V.; Wang, Y.; Turek, M. DARPA's Explainable AI (XAI) Program: A Retrospective. *Appl. AI Lett.* **2021**, *2*. [CrossRef]
59. Whitten, M. Introducing Unity Muse and Unity Sentis AI. Unity Blog (Blog). 27 June 2023. Available online: https://blog.unity.com/engine-platform/introducing-unity-muse-and-unity-sentis-ai (accessed on 8 August 2023).

60. Schappert, S. Twitter Blocks Non-Users from Reading Tweets over AI Data Scraping. Cyber News, 3 July 2023. Available online: https://cybernews.com/news/twitter-blocks-non-users-reading-tweets-ai-scraping/ (accessed on 8 August 2023).
61. Creamer, E. Authors File a Lawsuit against OpenAI for Unlawfully "Ingesting" Their Books. *The Guardian*, 5 July 2023. Available online: https://www.theguardian.com/books/2023/jul/05/authors-file-a-lawsuit-against-openai-for-unlawfully-ingesting-their-books (accessed on 8 August 2023).
62. Milmo, D. Sarah Silverman Sues OpenAI and Meta Claiming AI Training Infringed Copyright. *The Guardian*, 10 July 2023. Technology. Available online: https://www.theguardian.com/technology/2023/jul/10/sarah-silverman-sues-openai-meta-copyright-infringement (accessed on 8 August 2023).
63. Milmo, D.; Agency. US's Top Competition Watchdog Opens Investigation into ChatGPT Maker. *The Guardian*, 13 July 2023. Available online: https://www.theguardian.com/technology/2023/jul/13/ftc-investigate-chat-gpt-openai (accessed on 8 August 2023).
64. Hacker, P.; Engel, A.; Mauer, M. Regulating ChatGPT and Other Large Generative AI Models. In Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency, FAccT '23, Chicago, IL, USA, 12–15 June 2023; Association for Computing Machinery: New York, NY, USA, 2023; pp. 1112–1123. [CrossRef]
65. Potter, M.C.; Wyble, B.; Hagmann, C.E.; McCourt, E.S. Detecting meaning in RSVP at 13 ms per picture. *Atten. Percept. Psychophys.* **2013**, *76*, 270–279. [CrossRef] [PubMed]
66. Ventayen, R.; Magno, J. OpenAI ChatGPT Generated Results: Similarity Index of Artificial Intelligence-Based Contents (21 January 2023). Advances in Intelligent Systems and Computing. 2023. Available online: https://ssrn.com/abstract=4332664 (accessed on 8 August 2023).
67. Finnie-Ansley, J.; Denny, P.; Becker, B.A.; Luxton-Reilly, A.; Prather, J. The Robots Are Coming: Exploring the Implications of OpenAI Codex on Introductory Programming. In Proceedings of the 24th Australasian Computing Education Conference (ACE '22), Virtual Event, Australia, 14–18 February 2022; Association for Computing Machinery: New York, NY, USA, 2022; pp. 10–19. [CrossRef]
68. Luccioni, A.S.; Viguier, S.; Ligozat, A.-L. Estimating the Carbon Footprint of BLOOM, a 176B Parameter Language Model. *arXiv* **2022**, arXiv:2211.02001. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.