

Team Members: Cole Arduser, Luke Farmer, Sam Nicklaus, Sam Loecke

ECE:5995 Generative AI Tools: ChatGPT and Beyond

Final Project Report

Source Code: [Here](#)

# GAIT Tanks Final Project

## Project Background

For our project, GAIT Tanks, we wanted to see how the current state of AI was able to make a basic video game. For our game, we settled on a clone of [Wii Tanks](#) in a simpler style (top down, retro design for simpler sprites). The goal of the game is to eliminate all enemy tanks on the map by shooting them. We thought about user experience as well with a main menu with level selection and a controls page. From an AI standpoint, every aspect of the game is either generated by AI or assisted by AI, including the generation of image assets, an intro video, level design, music selection, and more.

## Overview of System

Our game is built using the Python library [Pygame](#), a popular python library used for video games. With the help of ChatGPT we initially created the main game logic and code structure. The main game loop's tickrate is set to 30 which will run the main game loop 30 times per second, processing data, updating objects and rendering those new objects on screen. All game states have their own class inherited from the State class in state.py. Each has its own update and render functions which handle updating the game objects and then rendering the screen with those updated objects. The main Game object which is running the game loop holds a list of the states with the last one in the list being the current state. Update and render are run on the last item in the list which updates the objects in that state and then renders the screen to show the state on the screen. In each state's update function there is logic to determine when to switch and which state the game should be switched to.

## **Core Game Mechanics Overview**

Main Game File (game.py): Initializes the main game loop, which handles updating and rendering game states. When the player moves from state to state in the app the current state is updated to show to correct one on the screen.

States (src/states/): Each state inherits from the main state.py class. Each has its own update and render functions which handle updating the game objects and then rendering the screen with those updated objects.

Audio Management (audio.py): Manages all audio aspects of the game, including loading and playing sound effects for different game actions like tank movements and explosions.

Pathfinding (pathfinding.py): Implements A\* pathfinding algorithms for AI-controlled tanks, ensuring efficient movement and navigation within the game environment.

## **Objects**

Player and Enemies: Both inherit from the main Tank Class which provides base properties that apply to both player and enemies. Both inheriting classes add the more specific functionality that only applies to them.

Bullet and FireEffect: Both housed in the bullet.py file and deal with the creation, updating, rendering, and deletion of the bullet sprites during the gameplay.

## **Assets**

Audio, Images, and Videos: The project includes a comprehensive set of assets such as audio files, background images, sprites, and videos, crucial for the game's visual and auditory elements.

## **Technical Aspects**

### **Overall GAITs Used:**

- Text - [GPT-4](#), [GPT-3.5](#)
- Image Generation - [DALL-E 3](#), [Stable Diffusion](#)
- Audio Generation - [Stable Audio](#), [Facebook Musicgen](#), [Riffusion](#)
- Video Generation - [Pika](#)
- Text-to-speech Generation - [ElevenLabs](#)

### **Workflow:**

Conceptualization with ChatGPT: Initially, we used ChatGPT to brainstorm game concepts, mechanics, and narratives. ChatGPT's ability to provide instant feedback and

suggestions was helpful in this creative phase. Additionally, ChatGPT was able to supply a base code for our game and allowed us to troubleshoot issues if we had any.

Design and Asset Creation: Once the game concept was solidified, we moved to design and asset creation using DALL-E 3 and Stable Diffusion for visuals, and Stable Audio and Facebook Musicgen for audio. This phase involved a lot of back-and-forth, fine-tuning the AI-generated assets to fit what we wanted.

## Discussion

### **Contributions:**

- Cole Arduser - Main game logic, enemy pathfinding, sprite creation, and level creation
- Luke Farmer - Music and sound generation and integration into gameplay, controls image generation
- Sam Loecke - Introduction video creation and addition into game, controls page implementation
- Samuel Nicklaus - Image generation, background music generation, wrote report, assisted on main code base when needed

### **Issues Encountered**

Throughout the project, we faced several challenges. Balancing the AI-generated images and sprites with the gameplay mechanics was a significant hurdle. We learned the importance of fine-tuning AI outputs to align with what we wanted the game to look like. To address these issues, we iterated on AI prompts and integrated manual adjustments when needed.

When writing the main game logic we ran into a couple challenges with ChatGPT not having our entire repository which would lead to overlap in the code. We eventually found that GPT-4 has a Github Plugin that allows you to link a public repo that it can look at and process.

### **Ethical Considerations**

Our project raises questions about the creative process in the age of AI. By relying heavily on AI tools, we explored the boundaries between human creativity and AI assistance. This project reflects on the evolving landscape of digital entertainment, highlighting potential shifts in how games are developed and experienced. Additionally, our use of AI-generated content brought to the forefront important considerations regarding copyright and intellectual property. As AI tools like DALL-E 3 and Stable Diffusion generate content based on datasets of existing art and media, the question of originality and ownership becomes complex. In recent news, OpenAI has come out and stated that they would pay for all legal fees that are related to copyright, which will be interesting in the future when a business inevitably gets sued for using AI generated content.

## **AI Retrospective**

What makes this project so interesting is that we used a high variety of GAITs to achieve our final product. Image generation is probably the most significant portion of our project and was where we encountered the most issues. With the current state of image generation (Stable Diffusion, DALL-E 3, etc) there is no way to edit or tweak existing AI generated images. Specifically with DALL-E and GPT-4, when we prompted GPT-4 to edit an image it was unable to get context from the previous part of the conversation. Additionally, generating really specific styles of images was tough for the AI. For example, getting DALL-E to display a top down view of any prompt was a feat in itself. We tried keywords like 'dropdown', 'icon', 'favicon', 'sprite', etc and more times than not it would give a result we were not satisfied with. With that being said, image generation is progressing fast enough where I don't think this will be a problem even 6 months from now. It will be interesting to see how the quality of assets change through image gen in video games in general.

Audio generation was also interesting for creating background music or the sound effects for the tank. In its current state, audio generation is a very powerful tool but can be inconsistent at times, meaning you have to be very specific when you prompt the AI to make good music. For shorter audio clips such as the tank shooting sound, The AI did a great job at making what we wanted. In my opinion, audio generation falls under the same category as image gen where it doesn't feel as polished as text generators like ChatGPT.

Video generation definitely felt the most unpolished of the three, which makes sense since from a technical perspective it is the most complicated. It was cool to see this area of AI

grow throughout the semester watching Pika as a platform mature.

The pace at which GAITs are progressing points towards a transformative impact on game development. In the next few years, we hope to anticipate more refined and intuitive interfaces for AI tools, enabling seamless integration into creative processes. The potential for AI to not only assist but also co-create with developers will likely advance the boundaries of game design and development.

## Conclusion

GAIT Tanks serves as a historical snapshot of what AI can currently accomplish in game design. As we look to the future, it's clear that the integration of GAITs will play a significant role in shaping the landscape of digital creativity and entertainment. We're excited to see how these tools evolve and influence the industry in the years to come.