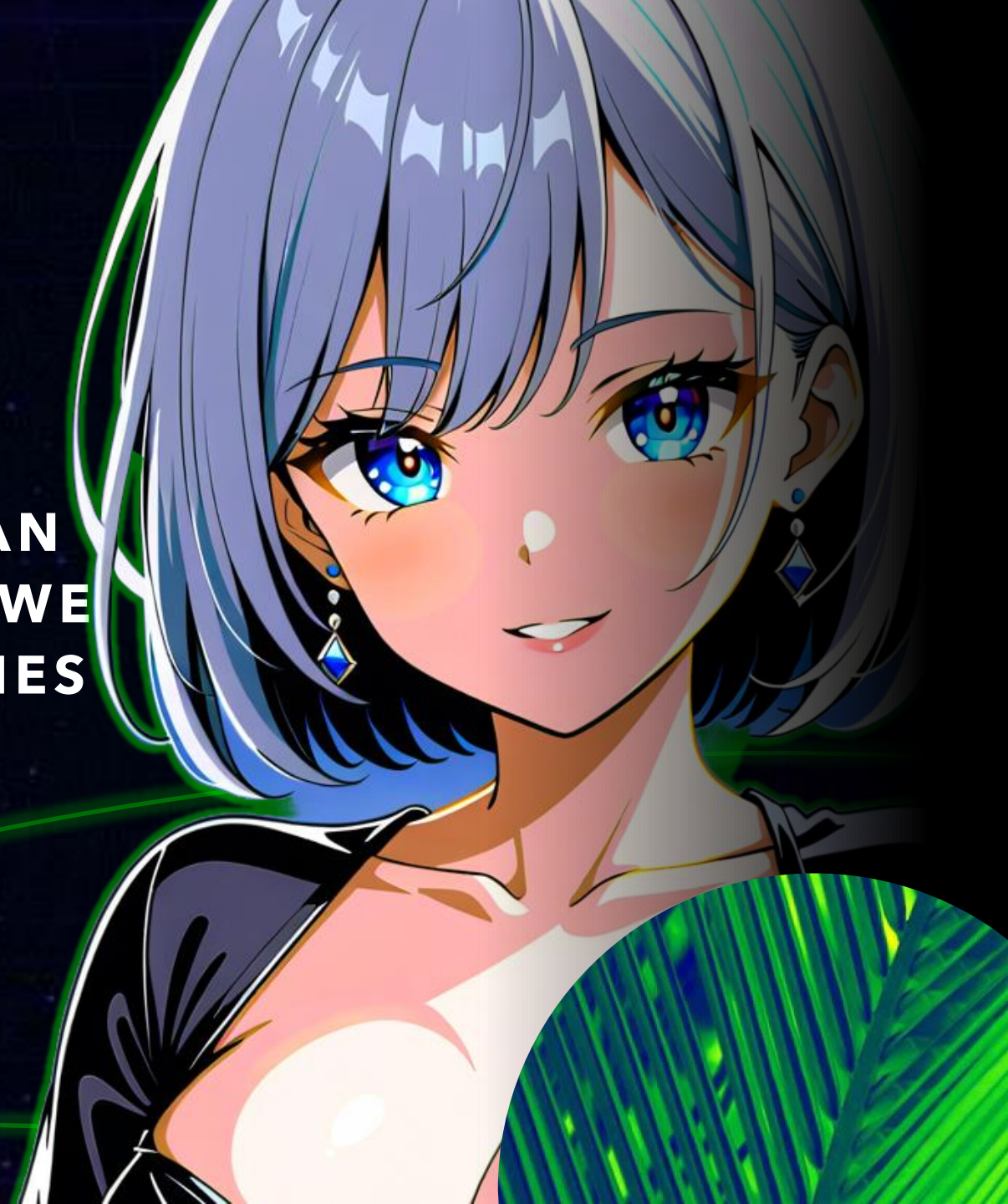
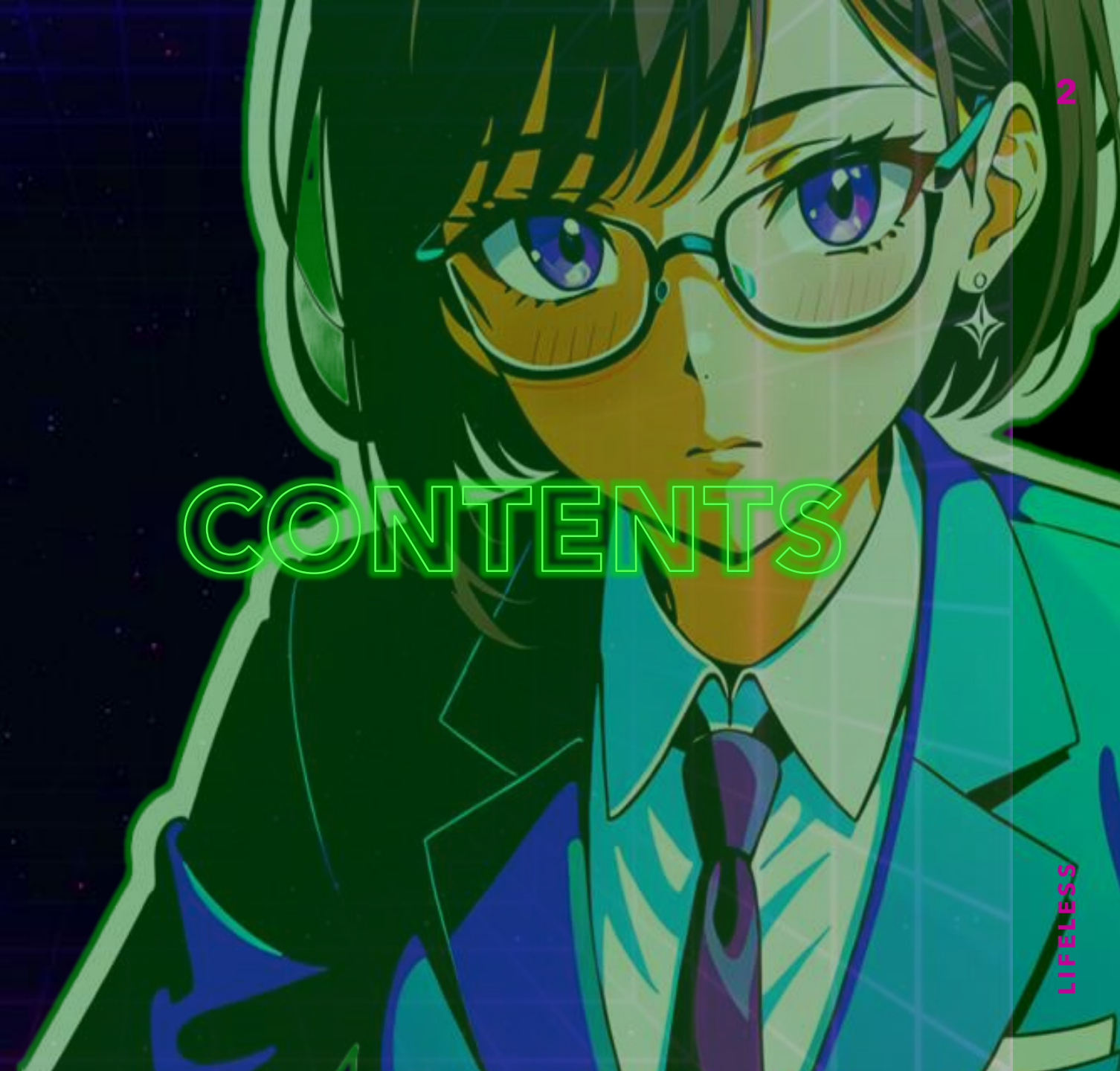
LIFELESS

VIDEO GAME'S NEW ERA

OR, HOW AI CAN CHANGE HOW WE PLAY OUR GAMES



CONTENTS



ABOUT ME

Luis Chacon, Game designer by passion.

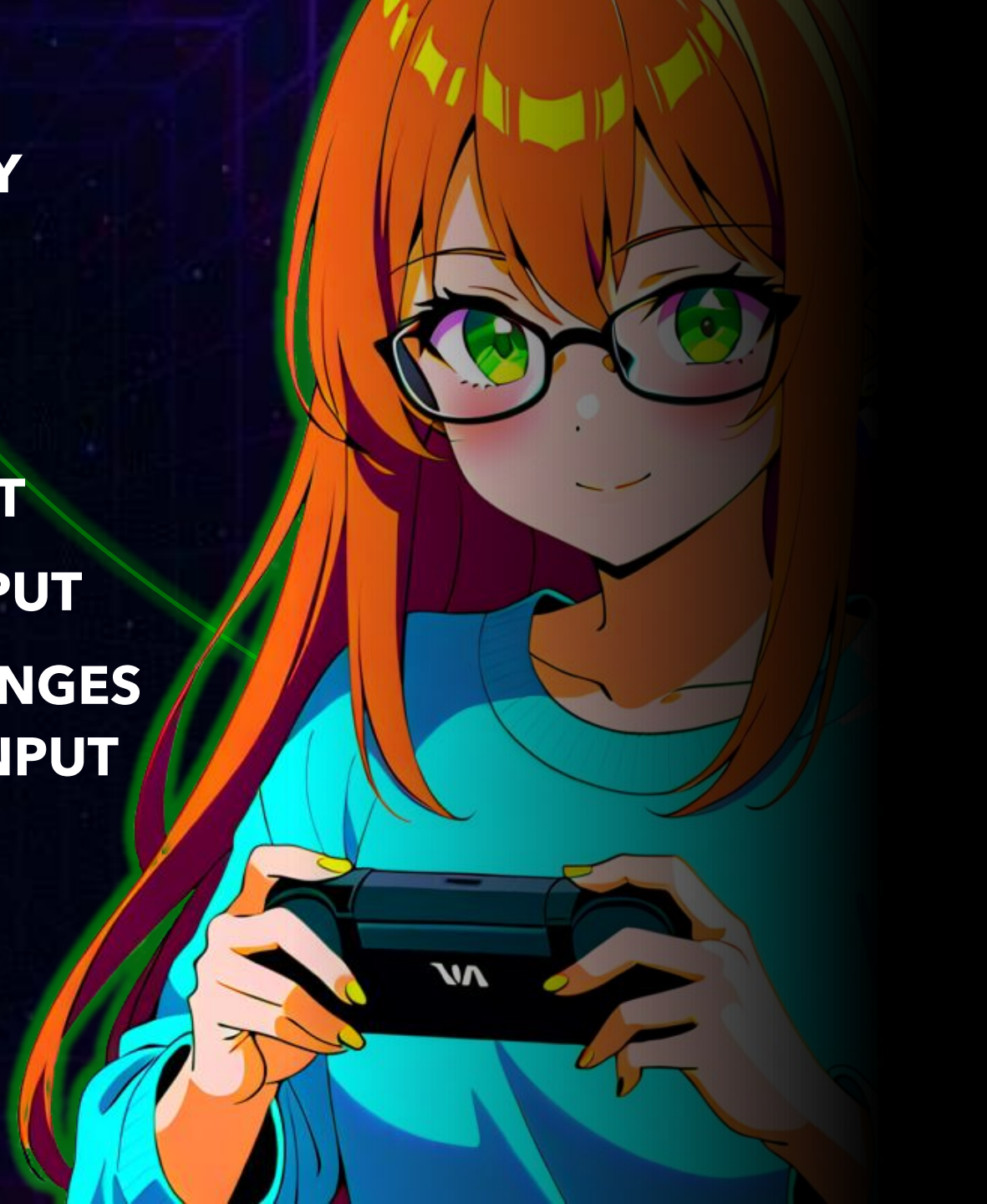
Systems Engineer Student

Microsoft Cloud Support Engineer

BACKGROUND

HOW WE CURRENTLY INTERACT WITH GAMES

- PREDEFINED INPUT
- PREDEFINED OUTPUT
- LITTLE TO NO CHANGES BASED ON USER INPUT

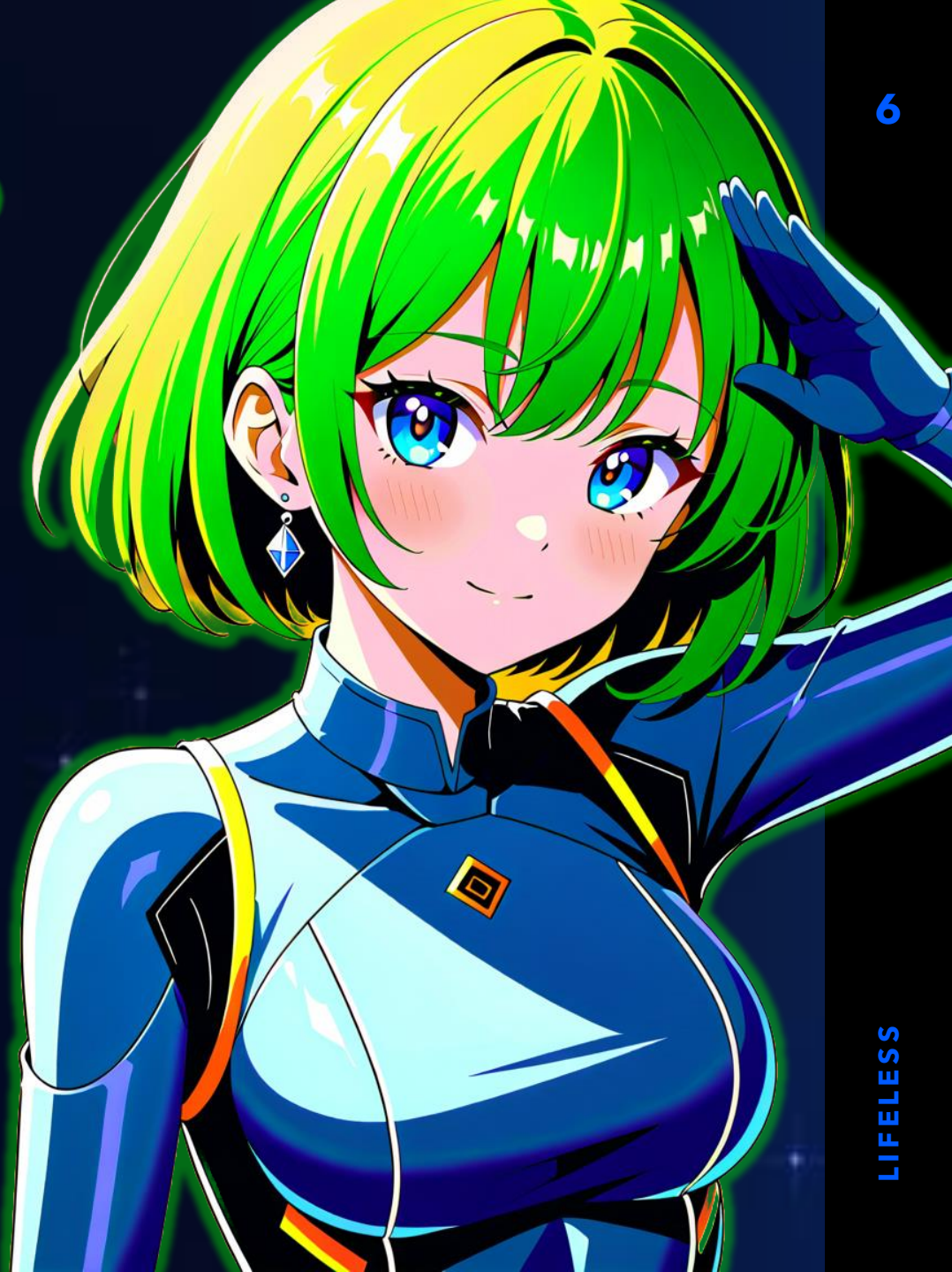


CURRENT LIMITATIONS

- Scripted Responses
- Limited Behavioral Complexity
- Inconsistent Reactions
- Lack of Personality

HOW CAN AI HELP?

- Dynamic Dialogue Generation
- Adaptive Behaviors
- Contextual Awareness
- Memory Systems
- Emotional Responses

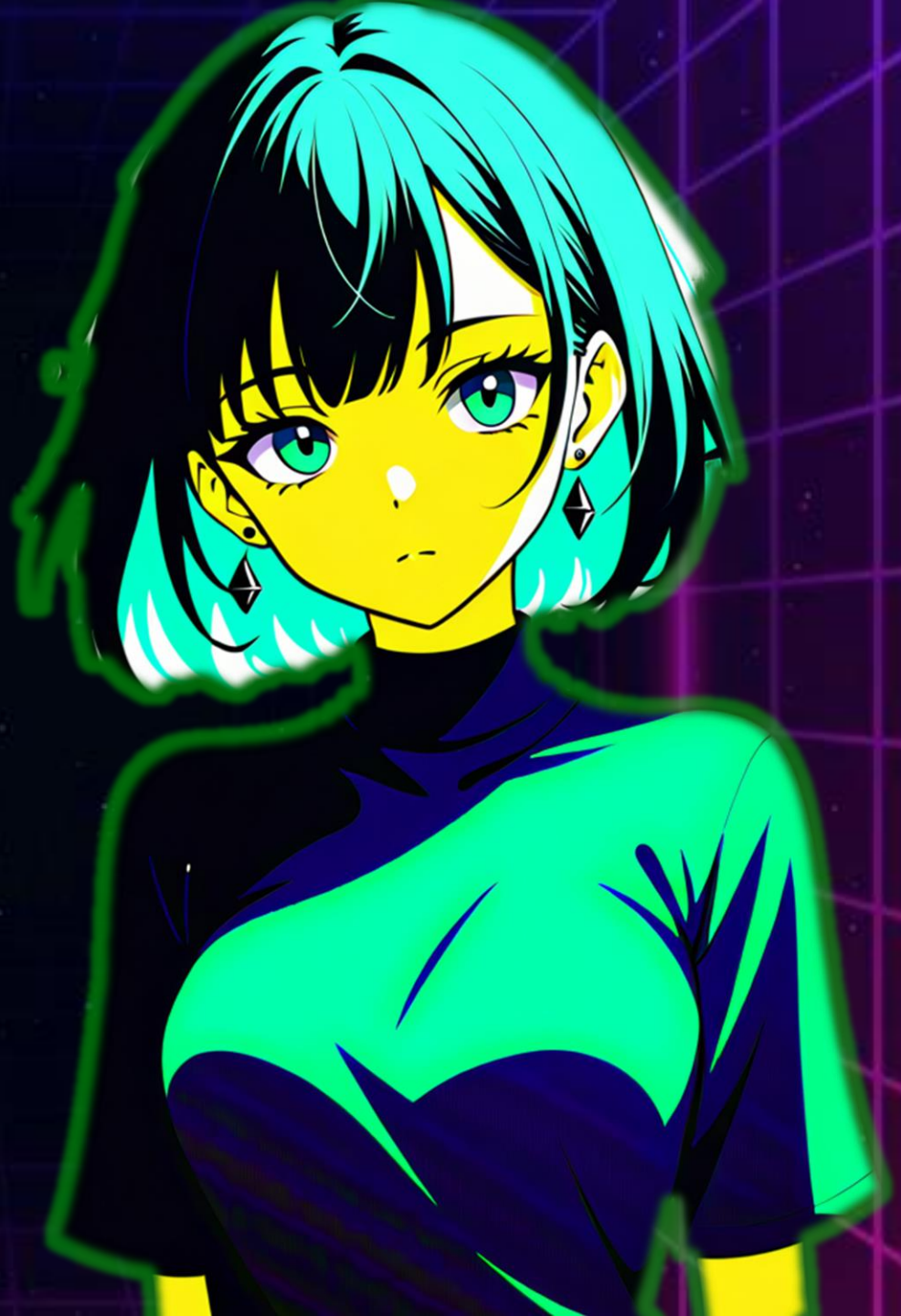


OVERVIEW OF LIFELESS.

EARLY PROTOTYPE AND PROOF OF CONCEPT

MAIN GOALS

- UNSCRIPTED CHARACTERS
- CONVERSATIONAL MEMORY
- CHARACTER BASED PERSONALITY
- EMOTIONAL RESPONSE TO PLAYER INPUTS



DEVELOPMENT AND MAIN TECHNOLOGIES USED

OLLAMA

As Backend for LLM Models

GODOT 4

Game Engine

JSON FILES

Storage of Character Data

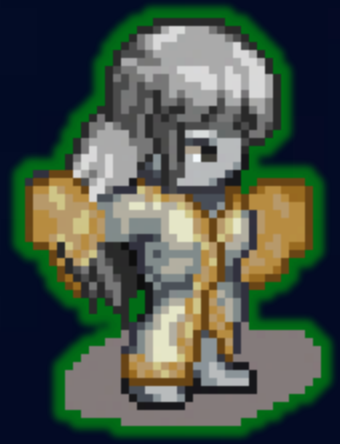
Customization

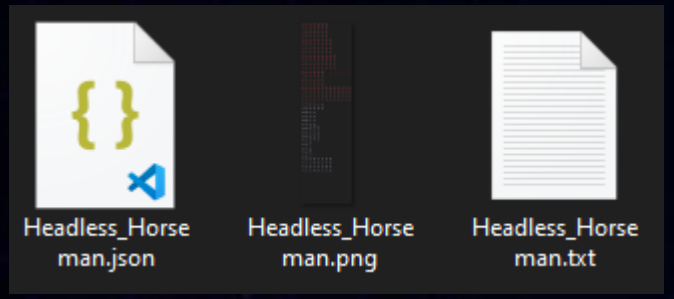
HTTP REQUESTS

Connecting With Backend LLM

WHAT IS A LIFELESS?

Dynamic customizable character that can respond to input from the user in a cohesive way and can keep short to medium term memory of events.






```
{
    "Name":"Headless",
    "Surname":"Horseman",
    "Description":"I am the headless horseman, I use a pumpkin as
    a head as I do not have one. I take care of those who wrong
    the innocent, killing them as punishment"
}
```

Lifeless have 2 main components:

JSON File – This keeps the Name, Surname and Description for the character personality

PNG File – This is the sprite sheet used as a skin for the Lifeless. Is based on the LCP Sprites from Open Game Art.

(There is a txt in the default skins for the attribution of the sprite sheet)

User created Lifeless are stored at: "%APPDATA%/AmeNoHi/Lifeless/Lifeless Skins"

COMPONENTS OF A LIFELESS

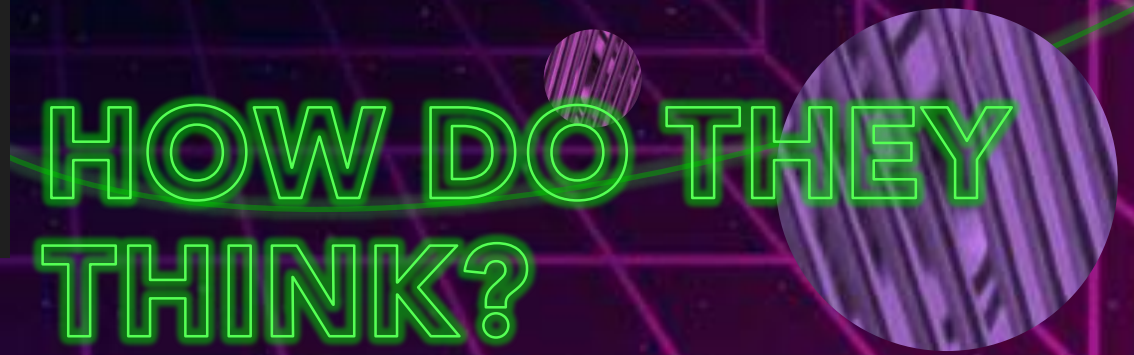
Scripts > Simulation > System > SentienceSystem.gd > ...

```gdscript
extends Node

# HTTP Request
var http_request_generate_completion = HTTPRequest.new()


# Called when the node enters the scene tree for the first time.
func _ready():
    # Add HTTP Request as child and connect signals
    add_child(http_request_generate_completion)
    http_request_generate_completion.request_completed.connect(_handle_generate_completion)

    # Connect to Herald for communication across nodes using the bus
    Herald.request_sentience_generative_completion.connect(_generate_completion)

# Connected to the signal "http_request_generate_completion" so it can be called from the UI
func _generate_completion(model: String, prompt: String, context: Array):
    # Prepare the endpoint and make the reques
    var endpoint = global.address + ":" + global.port + global.path_generate
    # Prepare parameters to send
    var parameters = {
            "model": model,
            "prompt": prompt,
            "context": context,
            "stream": false
        }

    # Prepare headers as a PackedArray
    var headers = [
        "Content-Type: application/json"
    ]

    # Make the request
    http_request_generate_completion.request(endpoint, headers, HTTPClient.METHOD_POST, JSON.stringify(parameters))

# Handles the Generated response and returns it down the pipeline to be displayed
func _handle_generate_completion(_result, response_code, _headers, body):
    if response_code == 200:
        var body_dictionary = JSON.parse_string(body.get_string_from_utf8())
        Herald.return_sentience_generative_completion.emit(body_dictionary)
    else:
        print("I failed, master")
```

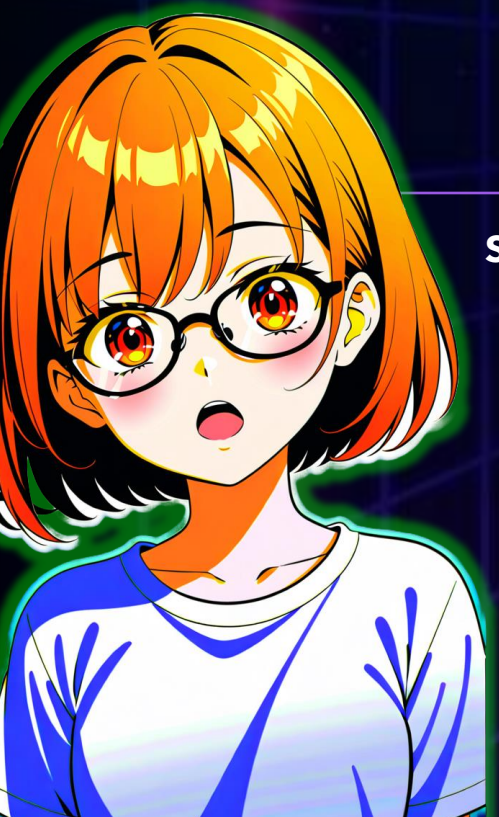
Ollama Backend API

Using the Sentience System developed on Godot, it sends a request to Ollama using the Generate Completion.

The completion is returned, and the Generated response is sent back down the pipeline to be displayed along with the context to keep memory of the conversation.



FLOW OF INTERACTION



Step 1 — USER INTERACTS WITH LIFELESS

Step 2 — LIFELESS DATA IS LOADED

Step 3 — HTTP REQUEST IS MADE TO OLLAMA API

Step 4 — RESPONSE IS GENERATED AND SENT BACK

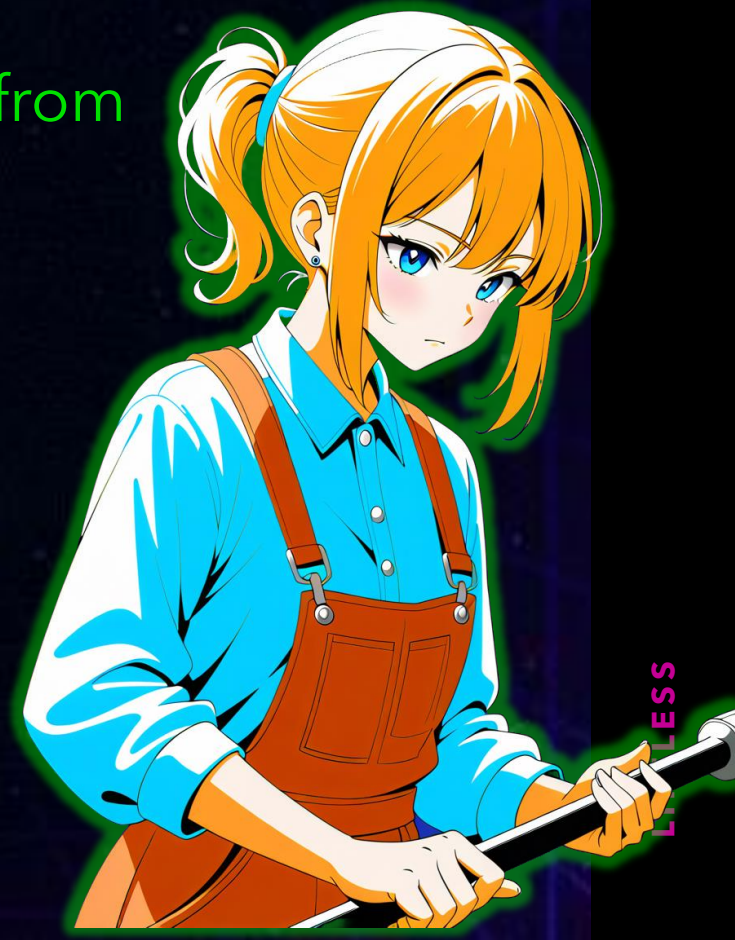
Step 5 — GENERATION IS DISPLAYED ON CHAT

LIVE DEMO

EARLY PROTOTYPE AND PROOF OF CONCEPT

TECHNICAL LIMITATIONS

- Input Methods
- Portability
- Resource Requirements
- Data Processing Capabilities
- Limited Response from Lifeless



BENEFITS AND IMPACT

SINGLE PLAYER

+ Added Depth to Characters
+ World Awareness
+ Tailored and Unique Experience

ONLINE

+ Better NPCs
+ Reaction to Player Evolution
+ World Building

STORY FOCUSED GAMES

+ Adaptiveness to player actions on the world
+ More Realistic Interactions
+ Added Immersion

FUTURE
AND ITS POSSIBILITIES

TAILORED MODELS

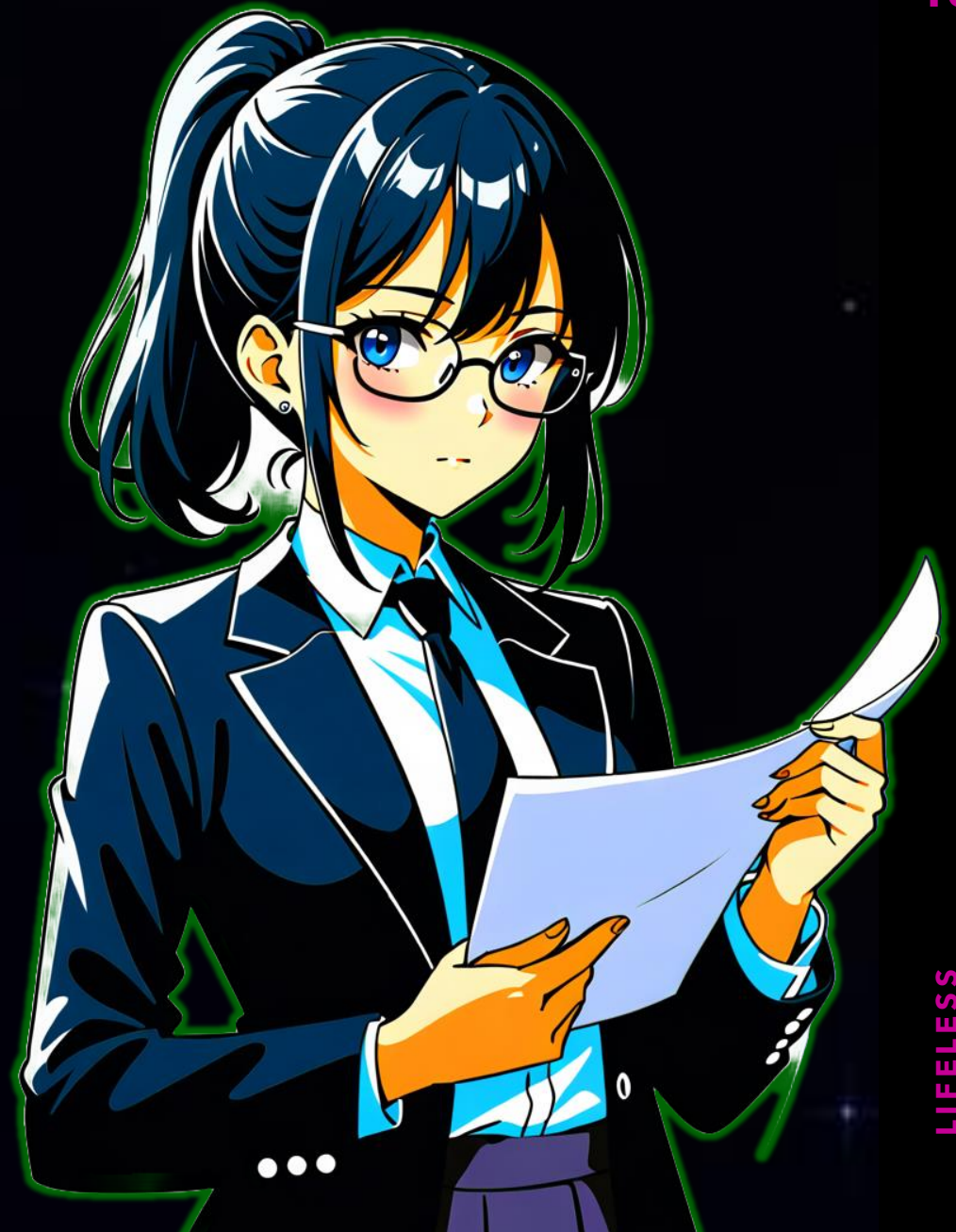
BETTER INTEGRATION WITH GAME ENGINES

NEW WAYS TO PROCESS DATA

GENERATION OF ASSETS ON THE FLY

BUSINESS IMPACT

- Enhanced Player Retention and Engagement
- Revolutionizing Game Narrative and Storytelling
- Opening New Genres and Gameplay Styles
- Improving Game Accessibility
- Boosting Development Efficiency and Creativity
- Enhanced Social and Educational Uses
- Global Market Expansion



TANK YOU VERY MUCH!