

# Greetings and Wellbeing



hi chatgpt how are you



Hi! I'm doing great, thanks for asking. How about you?



if I chat with you today would you remember the next time I chat with you ?



I can remember details from our conversation if you'd like, but only for the duration of our ongoing sessions. If you want me to remember something specific for future chats, I can make a note of it. Would you like me to remember anything for next time?



I want to do Master degree by research about combining Computer Graphic with Artificial Intelligence. I am interested in Heuristic Research and Deep learning Neural Network. I plan to do this master by using software like Blender and Unreal Engine. I want to do this master as a portfolio for my Sole proprietorship company. I want to know your advice on this matter



 **Memory updated**

Your research idea sounds fascinating and well-aligned with emerging trends in both AI and computer graphics! Here are some thoughts and advice to guide you:

## 1. Research Focus:

- Combining **AI and computer graphics** could have many directions, such as procedural content generation, real-time rendering, or creating intelligent agents in 3D environments.
- With **heuristic research**, you might explore optimization techniques for tasks like rendering, animation, or object recognition within virtual environments.
- **Deep learning neural networks** can help automate complex tasks like texture generation, lighting, character animation, or even creating AI-driven virtual environments.

## 2. Tools (Blender, Unreal Engine):

- **Blender** is excellent for modeling and rendering, and with Python integration, you can leverage deep learning models to interact with the environment.

- **Unreal Engine** is powerful for real-time simulation, and its Blueprints system can integrate AI models for tasks like NPC behavior, procedural generation, and environmental interactions.

### 3. AI Techniques:

- Look into **Generative Adversarial Networks (GANs)** for generating textures, 3D models, or other assets.
- **Reinforcement learning** could be great for teaching AI agents to navigate and interact with environments created in Unreal Engine.
- Using **neural style transfer** could enhance visuals in unique ways, blending artistic styles with real-time environments.

### 4. Portfolio Building:

- Since you're planning to use this for your sole proprietorship company, consider projects that not only show technical prowess but also **solve practical business problems** (e.g., AI-driven tools for clients, automated content creation pipelines).
- Developing plugins or add-ons for **Blender or Unreal Engine** can demonstrate both research innovation and practical application, which is valuable for a portfolio.

### 5. Collaborations and Resources:

- Look into **open-source communities** around Blender and Unreal Engine; contributing or collaborating there can provide exposure.
- Research papers on **neural rendering** or **AI-driven content generation** would give you insights and help you frame your research within the context of current work.

Do you have any specific challenges or questions about the technical, academic, or business aspects?



I only enrolled in two classes of Computer Graphic and Intelligent System during my degree. SO I dont really have enough background in Computer Graphic and Intelligent System. Do you know what is the latest trend in both of these fields?



 **Memory updated**

It's great that you've taken classes in both **Computer Graphics** and **Intelligent Systems**, even if you feel your background isn't strong. To help bridge the gap and stay updated, here are some of the **latest trends** in both fields:

## Computer Graphics:

### 1. Real-time Ray Tracing:

- **Ray tracing** simulates light more realistically, producing stunning visual effects. NVIDIA's **RTX** technology has been pushing this, allowing real-time ray tracing in video games and interactive media.
- Explore how ray tracing integrates into game engines like **Unreal Engine**, which supports real-time rendering.

## 2. Neural Rendering:

- **Neural rendering** blends traditional computer graphics with AI to synthesize high-quality images, often using fewer computational resources.
- Companies like NVIDIA are leading with techniques such as **NeRFs (Neural Radiance Fields)**, which generate 3D scenes from 2D images.

## 3. Procedural Content Generation (PCG):

- **PCG** is about creating assets (textures, environments, models) algorithmically, often with minimal human intervention. AI is increasingly used to enhance procedural generation by creating more complex, natural, and varied assets.

## 4. Virtual Reality (VR) and Augmented Reality (AR):

- The combination of **real-time 3D rendering** and **interactive AI-driven environments** is a growing trend, especially with devices like **Meta's Quest 3** and **Apple's Vision Pro**.

## 5. Stylized Rendering:

- Non-photorealistic rendering (NPR) techniques, like **cel shading** or **toon shading**, are becoming more popular. These methods allow for artistic styles in games and movies, such as the visuals in the movie *Spider-Man: Into the Spider-Verse*.

# Intelligent Systems (AI and Machine Learning):

## 1. Generative AI:

- AI models like **GPT** and **DALL-E** (generating text and images) have inspired research into how generative models can be applied to graphics (e.g., generating textures or entire scenes).
- Generative AI also opens possibilities in creating content for virtual worlds in game engines like Unreal Engine.

## 2. Deep Reinforcement Learning (DRL):

- DRL is used in gaming and robotics for training AI agents to interact with environments. This is relevant if you're creating intelligent characters or autonomous systems within simulated 3D spaces.
- **OpenAI's Dota 2** and **DeepMind's AlphaStar** showcase how intelligent agents can learn complex tasks in real-time environments.

## 3. AI-Driven Content Creation:

- AI tools like **Runway ML** and **Artbreeder** allow designers and developers to generate images, animations, and even 3D assets with minimal input.
- AI-augmented pipelines for artists and developers are gaining ground, where AI assists or automates aspects of content creation.

## 4. Transformers in Vision:

- Originally used in NLP, **transformers** are now making their way into computer vision tasks, leading to advancements in object detection, image generation, and 3D reconstruction. Models like **Vision Transformers (ViT)** are pushing boundaries in image processing tasks.

## 5. AI in Edge Computing:

- With hardware becoming more powerful, there's a growing trend of deploying AI models on devices (e.g., smartphones, VR headsets) for real-time decision-making. This could be useful in VR/AR systems, where AI models process data in real time on-device.

## Learning Path Suggestions:

Since you mentioned feeling a lack of background, here's a suggestion:

- **For Computer Graphics:** Take courses or follow tutorials on **Blender**, **Unreal Engine**, or **Unity**. This will give you a practical understanding of the basics and let you start experimenting with rendering, modeling, and simulation.
- **For Intelligent Systems/AI:** Brush up on **machine learning basics** through free online platforms like **Coursera** or **Fast.ai**. Then, delve into **deep learning** with frameworks like **PyTorch** or **TensorFlow**.

This combination of trending topics and hands-on practice will help you quickly build the necessary foundation to pursue your Master's research.

Would you like more detailed resources or guidance on specific areas?



I plan to code in python scripting in Blender and c++ scripting in Unreal Engine



 **Memory updated**

That's a great plan! Using **Python scripting in Blender** and **C++ in Unreal Engine** will give you a lot of flexibility and control for your research. Here's how you can best approach both:

## Python Scripting in Blender:

- **Why Python?**
  - Blender's API is built with Python, making it a natural choice for scripting tasks like automating modeling, animation, texturing, and more. Python is also widely used for integrating machine learning models.
- **What to Focus On:**
  - **Automation:** You can automate workflows, generate procedural models, or optimize repetitive tasks.
  - **Data Import/Export:** Python can help with reading and writing data to and from different formats (e.g., OBJ, FBX, or even custom formats).
  - **Integration with AI:** You can integrate machine learning models to drive behavior in Blender (e.g., using trained models to generate procedural textures or 3D models).
  - **Simulation:** You can script physical simulations or modify them using AI (e.g., neural networks influencing object collisions, movement, or shape transformation).
- **Resources:**

- **Blender Scripting API:** Dive into Blender's Python API documentation to get a full understanding of how to manipulate Blender scenes.
- **GitHub Projects:** Look at existing open-source Blender Python scripts to see what others are doing in the AI and graphics space.

## C++ Scripting in Unreal Engine:

- **Why C++?**
  - C++ in Unreal Engine is powerful for creating optimized, real-time applications, especially when dealing with complex 3D environments and AI logic.
- **What to Focus On:**
  - **Performance:** C++ allows you to write highly performant code, which is essential for real-time rendering and AI. If you're working with AI for things like NPC behavior or environmental interaction, C++ can handle the heavy lifting.
  - **Gameplay Systems:** Develop more advanced, responsive AI systems that are tightly integrated with Unreal's engine. You can implement neural network decision systems or reinforcement learning agents for autonomous behaviors.
  - **Unreal Engine AI Tools:** Unreal Engine has built-in AI tools (Behavior Trees, AI Controllers, Navigation Mesh), which you can extend and customize with C++ to create intelligent systems for characters or procedural content generation.
- **Resources:**
  - **Unreal Engine Documentation:** Unreal's docs provide a wealth of information for C++ scripting.
  - **AI Integration:** Explore tutorials on integrating machine learning frameworks like **TensorFlow** or **PyTorch** into Unreal, especially if you're using AI to drive behaviors or automate content creation.

Combining these two skill sets will give you a comprehensive toolset for tackling cutting-edge projects in your Master's research. Would you like to explore any specific techniques or examples in these languages?



yes.



Great! Let's dive deeper into specific techniques and examples for both **Python scripting in Blender** and **C++ scripting in Unreal Engine**.

### 1. Python Scripting in Blender:

Here are some techniques you can focus on when scripting in Blender:

#### A. Procedural Mesh Generation

- **Example:** Automatically generate 3D objects using Python.

python

```
import bpy
from math import sin, cos

# Create a new mesh
mesh = bpy.data.meshes.new("GeneratedMesh")
obj = bpy.data.objects.new("GeneratedObject", mesh)
bpy.context.collection.objects.link(obj)

# Set up the vertices and faces
verts = [(sin(i), cos(i), 0) for i in range(16)]
faces = [(i, (i + 1) % 16, (i + 2) % 16) for i in range(16)]

# Build the mesh
mesh.from_pydata(verts, [], faces)
mesh.update()
```

- **Application:** You can use this for procedural content creation, building custom models or environments algorithmically.

## B. Integrating AI with Blender

- **Example:** Apply AI-generated textures to 3D models in Blender.
  - You can use a pre-trained neural network (like **StyleGAN**) to generate textures and then apply those textures directly to your models:

python

```
import bpy
from PIL import Image
import numpy as np

# Assume you have a pre-trained GAN that generates textures
generated_texture = np.random.rand(256, 256, 3) * 255 # Placeholder for a real GAN output
img = Image.fromarray(generated_texture.astype('uint8')).save("generated_texture.png")

# Apply the texture in Blender
bpy.data.images.load("generated_texture.png")
obj = bpy.context.active_object
mat = bpy.data.materials.new(name="GeneratedMaterial")
obj.data.materials.append(mat)
```

- **Application:** This is useful for automating the process of applying AI-generated textures to large numbers of assets, useful for procedural content generation in films or games.

## C. Animation and Simulation

- **Example:** Use Python to automate keyframe animation or physical simulations in Blender.

python

```
import bpy

# Create animation keyframes for an object
```

```
obj = bpy.context.active_object
obj.location = (0, 0, 0)
obj.keyframe_insert(data_path="location", frame=1)

obj.location = (5, 5, 5)
obj.keyframe_insert(data_path="location", frame=50)
```

- **Application:** You can use AI to control the parameters of these animations, for example, using reinforcement learning to have characters or objects "learn" how to move or behave naturally in the scene.

## 2. C++ Scripting in Unreal Engine:

Here's how you can leverage C++ to extend Unreal Engine's capabilities:

### A. Extending AI Behavior Trees

- **Example:** Write custom AI logic for NPC behavior using C++.
  - Unreal Engine uses **Behavior Trees** for AI decision-making. You can extend them by writing custom tasks in C++:

cpp

```
#include "YourTask.h"
#include "AIController.h"

UYourTask::ExecuteTask(UBehaviorTreeComponent& OwnerComp, uint8* NodeMemory)
{
    // Custom AI logic here
    AAController* AIController = OwnerComp.GetAIOwner();
    if(AIController)
    {
        // Move AI to a specific location
        FVector Location = FVector(100, 100, 0);
        AIController->MoveToLocation(Location);
    }
}
```

- **Application:** This is useful for controlling complex NPC behaviors like pathfinding, tactical movements, or reactions to player actions. You can also apply reinforcement learning to improve decision-making dynamically.

### B. Neural Network Integration

- **Example:** Integrating a pre-trained neural network into Unreal Engine using **C++**.
  - You can load a machine learning model from Python into C++ using libraries like **ONNX Runtime** for real-time inference:

cpp

```
#include <onnxruntime/core/session/onnxruntime_cxx_api.h>

// Load a pre-trained neural network model
Ort::Env env(ORT_LOGGING_LEVEL_WARNING, "AIModel");
Ort::SessionOptions session_options;
Ort::Session session(env, "model.onnx", session_options);
```

```
// Input and output tensors
std::vector<float> input_tensor_values = {...}; // Example input
std::vector<float> output_tensor_values;

Ort::Value input_tensor = Ort::Value::CreateTensor<float>(allocator,
input_tensor_values.data(), input_tensor_values.size(), input_shape.data(),
input_shape.size());
Ort::Value output_tensor = Ort::Value::CreateTensor<float>(allocator,
output_tensor_values.data(), output_tensor_values.size(), output_shape.data(),
output_shape.size());

session.Run(Ort::RunOptions{nullptr}, input_node_names.data(), &input_tensor, 1,
output_node_names.data(), &output_tensor, 1);
```

- **Application:** This allows you to use trained neural networks (for tasks like image recognition or decision-making) in real-time Unreal Engine environments. For example, you could apply it to procedural terrain generation or intelligent NPC decision-making.

## C. Procedural Content Generation (PCG) in Unreal

- **Example:** Generate procedural environments using C++.
  - You can create a procedural world generator, spawning items, landscapes, or levels at runtime.

cpp

```
void AProceduralWorld::GenerateTerrain()
{
    for(int x = 0; x < WorldSize; ++x)
    {
        for(int y = 0; y < WorldSize; ++y)
        {
            FVector Location(x * TileSize, y * TileSize, 0);
            SpawnActor<ATile>(TileClass, Location, FRotator::ZeroRotator);
        }
    }
}
```

- **Application:** Use this for generating large environments or levels on the fly, especially for open-world games or simulations. You can also apply AI to enhance the randomness and variation in content generation.

## Next Steps:

- **For Blender:** Start with small scripting projects (e.g., procedural model generation) and gradually integrate more AI components (like neural network-driven texture generation).
- **For Unreal Engine:** Learn the basics of C++ in Unreal, focusing on gameplay mechanics and AI. Then, look into integrating machine learning models to create more intelligent and dynamic content.

Would you like to explore more detailed tutorials or resources for any of these specific techniques?



yes





Here are detailed tutorials and resources for both **Python scripting in Blender** and **C++ scripting in Unreal Engine**, focusing on the techniques we discussed.

## 1. Python Scripting in Blender

### A. Procedural Mesh Generation Tutorial

- **Objective:** Learn how to create procedural 3D models using Python in Blender.
- **Tutorial:** [Blender Procedural Geometry with Python](#)
- **Documentation:** Blender's official Python API docs—[Blender API Mesh Class](#)
- **Steps:**
  1. Learn the basics of the **Blender API**, especially mesh creation (`bpy.data.meshes.new``).
  2. Create simple geometric shapes (cubes, spheres).
  3. Add complexity with loops to generate more detailed procedural objects.

### B. AI Texture Integration with Blender

- **Objective:** Integrate an AI model (e.g., a GAN) to generate textures and apply them to objects in Blender.
- **Tutorial:** [GAN Textures with Blender](#)
- **Steps:**
  1. Use pre-trained GAN models (e.g., **StyleGAN**) to generate textures.
  2. Export the textures and load them into Blender using Python.
  3. Apply these textures to 3D models automatically in Blender.

### C. Python Animation and Simulation Automation

- **Objective:** Automate keyframe animation and physics simulation with Python scripting.
- **Tutorial:** [Automating Blender with Python](#)
- **Steps:**
  1. Learn to insert keyframes using Python (`obj.keyframe_insert()`).
  2. Automate physics simulations, like fluid or cloth, by modifying parameters over time using scripts.
  3. Create custom animations by manipulating object locations and rotations programmatically.

## 2. C++ Scripting in Unreal Engine

### A. Extending AI Behavior Trees Tutorial

- **Objective:** Create custom AI tasks and behaviors in Unreal Engine with C++.
- **Tutorial:** [Unreal AI Programming with Behavior Trees](#)
- **Documentation:** [Unreal Engine AI and Behavior Trees](#)
- **Steps:**
  1. Set up a **Behavior Tree** in Unreal Engine and create a custom **C++ task**.
  2. Define custom logic inside the task (e.g., AI navigation, reactions to player actions).
  3. Use **blackboards** to store AI data and update the tree as actions are completed.

### B. Neural Network Integration in Unreal Engine

- **Objective:** Integrate a neural network (e.g., for decision-making) into Unreal Engine using C++.
- **Tutorial:** [Integrating ONNX in Unreal Engine](#)
- **Steps:**
  1. Install and configure **ONNX Runtime** in Unreal Engine.
  2. Load a pre-trained neural network (e.g., an image classification or reinforcement learning model).
  3. Use the neural network's outputs to influence in-game behaviors (e.g., intelligent NPCs, procedural content generation).

### C. Procedural Content Generation (PCG) in Unreal

- **Objective:** Use C++ in Unreal Engine to generate procedural landscapes and environments.
- **Tutorial:** [Procedural Generation in Unreal Engine](#)
- **Documentation:** [Procedural Mesh Component in Unreal](#)
- **Steps:**
  1. Set up a procedural world generator in Unreal Engine.
  2. Use C++ to randomly spawn terrain, buildings, and props at runtime.
  3. Add AI to dynamically change the environment based on player input or game events.

## 3. Additional Resources

### Blender Python Scripting

- **Blender Artists Forum:** This forum is an excellent place to ask questions and find Python scripts.
- **Blender's GitHub:** [Blender Python Templates](#)
- **Book:** *The Blender Python API* by Chris Conlan is a great resource for learning Python scripting in Blender.

### Unreal Engine C++ Scripting

- **Unreal Engine Forum:** Discuss and find solutions for C++ scripting.
- **Book:** *Learning C++ by Building Games with Unreal Engine 4* by Sharan Volin, for comprehensive C++ game development tutorials.
- **Video Series:** [Unreal C++ Tutorial Playlist](#) — Excellent video series on C++ in Unreal.

## Next Steps

I suggest following these tutorials step by step and starting small. First, automate a simple task (like procedural mesh generation in Blender) and then gradually move towards more complex integrations (like AI-driven procedural content in Unreal Engine).

Would you like further explanation or code examples on any specific area?



yes



Great! Let's dive into a bit more detail on specific areas. Please let me know which one you'd like to focus on, and I'll provide more in-depth explanation or a more detailed code example for that topic:

### 1. Procedural Mesh Generation in Blender (Python)

- Automating the creation of 3D models (basic or complex).
- Example: Creating a procedural city or terrain.

### 2. AI Texture Integration in Blender (Python)

- How to apply AI-generated textures to 3D objects.
- Example: Using StyleGAN to generate textures for procedural objects.

### 3. Custom AI Behavior Trees in Unreal Engine (C++)

- Writing C++ tasks for NPC behavior control.
- Example: AI navigation, decision-making, and interaction with the environment.

### 4. Neural Network Integration in Unreal Engine (C++)

- Using ONNX Runtime or TensorFlow for real-time AI decision-making.
- Example: Adding neural network-based intelligent NPC behavior or content generation.

### 5. Procedural Content Generation in Unreal Engine (C++)

- Writing C++ code to generate levels, terrain, or dynamic objects.
- Example: Random terrain generation or dungeon creation at runtime.

Let me know which one interests you the most!



all of them



Let's go through each topic one by one with more detailed explanations and example code!

## 1. Procedural Mesh Generation in Blender (Python)

**Goal: Automate the creation of 3D models like cities, landscapes, or custom objects.**

**Example: Procedural City Generation**

This example generates a grid of buildings with randomized heights, simulating a simple cityscape.

python

```
import bpy
import random

def create_building(x, y, width, depth, height):
    bpy.ops.mesh.primitive_cube_add(size=1, enter_editmode=False, align='WORLD',
                                     location=(x + width / 2, y + depth / 2, height / 2))
```

```
obj = bpy.context.active_object
obj.scale = (width / 2, depth / 2, height / 2)

# Parameters for city generation
city_width = 10
city_depth = 10
min_building_height = 2
max_building_height = 15

# Create buildings in a grid
for i in range(city_width):
    for j in range(city_depth):
        width = random.uniform(1.0, 3.0)
        depth = random.uniform(1.0, 3.0)
        height = random.uniform(min_building_height, max_building_height)
        create_building(i * 5, j * 5, width, depth, height)
```

### Explanation:

- **create\_building():** Creates a building at a given (x, y) location with specified dimensions.
- **Loop:** The city is generated by placing buildings in a grid, with randomized heights for variety.

### Use Case:

- **Procedural World Generation:** This can be used for games or simulations where entire cities or landscapes are created programmatically.

## 2. AI Texture Integration in Blender (Python)

**Goal:** Use AI (like StyleGAN) to generate textures and apply them to 3D models in Blender.

### Example: Apply AI-Generated Textures to a Cube

We'll simulate an AI-generated texture and apply it to a cube.

python

```
import bpy
from PIL import Image
import numpy as np

# Generate a random texture (this simulates output from an AI model like StyleGAN)
def generate_ai_texture():
    texture_data = np.random.rand(256, 256, 3) * 255 # Random texture for demo
    img = Image.fromarray(texture_data.astype('uint8'))
    img.save("ai_generated_texture.png")
    return "ai_generated_texture.png"

# Load the texture into Blender and apply to an object
def apply_texture_to_object(texture_path):
    # Create material
    material = bpy.data.materials.new(name="AI_Material")
    material.use_nodes = True
    bsdf = material.node_tree.nodes["Principled BSDF"]

    # Load texture image
    tex_image = material.node_tree.nodes.new('ShaderNodeTexImage')
    tex_image.image = bpy.data.images.load(texture_path)
    material.node_tree.links.new(bsdf.inputs['Base Color'], tex_image.outputs['Color'])

    # Assign material to active object
    obj = bpy.context.active_object
```

```

if obj.data.materials:
    obj.data.materials[0] = material
else:
    obj.data.materials.append(material)

# Create a cube and apply AI-generated texture
bpy.ops.mesh.primitive_cube_add(size=2, enter_editmode=False, align='WORLD', location=(0,
0, 0))
texture_path = generate_ai_texture()
apply_texture_to_object(texture_path)

```

### Explanation:

- **generate\_ai\_texture():** Generates a random image to simulate an AI-generated texture.
- **apply\_texture\_to\_object():** Loads the texture into Blender and applies it to the active object.

### Use Case:

- **AI-Assisted Design:** Use machine learning models to generate unique textures, automatically applying them to 3D models for faster creative workflows.

## 3. Custom AI Behavior Trees in Unreal Engine (C++)

**Goal:** Extend Unreal's AI Behavior Trees with custom C++ tasks.

### Example: Creating a Custom AI Task in C++

cpp

```

#include "MyCustomTask.h"
#include "AIController.h"
#include "BehaviorTree/BlackboardComponent.h"
#include "GameFramework/Actor.h"

UMyCustomTask::UMyCustomTask()
{
    NodeName = "Move To Random Location";
}

EBTNodeResult::Type UMyCustomTask::ExecuteTask(UBehaviorTreeComponent& OwnerComp, uint8*
NodeMemory)
{
    // Get AI Controller and Controlled Pawn
    AAIController* AIController = OwnerComp.GetAIOwner();
    APawn* ControlledPawn = AIController->GetPawn();

    if (ControlledPawn)
    {
        FVector RandomLocation = ControlledPawn->GetActorLocation() + FMath::VRand() *
500; // Random point 500 units away
        AIController->MoveToLocation(RandomLocation);
        return EBTNodeResult::Succeeded;
    }

    return EBTNodeResult::Failed;
}

```

### Explanation:

- **UMyCustomTask::ExecuteTask:** This function moves the AI-controlled pawn to a random location.
- **Behavior Trees:** Add this task as a node in the Behavior Tree editor, where it will execute custom C++ code.

#### Use Case:

- **Dynamic AI:** This custom task can make AI-controlled characters move randomly or respond to environmental factors.

## 4. Neural Network Integration in Unreal Engine (C++)

**Goal:** Integrate pre-trained neural networks into Unreal Engine for decision-making.

**Example:** Using ONNX to Run a Neural Network Model in Unreal Engine

cpp

```
#include <onnxruntime/core/session/onnxruntime_cxx_api.h>

void RunNeuralNetwork()
{
    Ort::Env env(ORT_LOGGING_LEVEL_WARNING, "UnrealNN");
    Ort::SessionOptions session_options;

    // Load the model
    Ort::Session session(env, "path_to_model.onnx", session_options);

    // Prepare input tensor
    std::vector<int64_t> input_shape = {1, 3, 224, 224}; // Example shape for an image
    input
    std::vector<float> input_tensor_values(224 * 224 * 3);

    // Create tensor
    Ort::Value input_tensor = Ort::Value::CreateTensor<float>(input_tensor_values.data(),
    input_tensor_values.size(), input_shape.data(), input_shape.size());

    // Run the model
    std::vector<const char*> input_node_names = {"input_node"};
    std::vector<const char*> output_node_names = {"output_node"};
    auto output_tensor = session.Run(Ort::RunOptions{nullptr}, input_node_names.data(),
    &input_tensor, 1, output_node_names.data(), 1);

    // Process the output (for example, control NPC movement based on network output)
}
```

#### Explanation:

- **ONNX Runtime:** This allows Unreal to load and execute a pre-trained neural network model.
- **Input/Output:** Input image data (or other data formats) can be fed into the neural network to influence game decisions, like NPC behavior or procedural generation.

#### Use Case:

- **Intelligent AI:** NPCs can make decisions based on the neural network's output, enabling more adaptive and realistic behaviors.

## 5. Procedural Content Generation in Unreal Engine (C++)

**Goal:** Generate dynamic environments using C++ in Unreal Engine.

**Example:** Procedural Terrain Generation

cpp

```
void AProceduralTerrain::GenerateTerrain()
{
    for (int x = 0; x < WorldSizeX; ++x)
    {
        for (int y = 0; y < WorldSizeY; ++y)
        {
            FVector Location(x * TileSize, y * TileSize, FMath::PerlinNoise2D(FVector2D(x,
y)) * MaxHeight);
            SpawnTileAt(Location);
        }
    }
}

void AProceduralTerrain::SpawnTileAt(FVector Location)
{
    FActorSpawnParameters SpawnParams;
    GetWorld()->SpawnActor<ATile>(TileClass, Location, FRotator::ZeroRotator,
SpawnParams);
}
```

### Explanation:

- **GenerateTerrain():** This function generates terrain tiles at random heights using Perlin Noise for a natural look.
- **SpawnTileAt():** Spawns a tile (terrain segment) at a given location.

### Use Case:

- **Dynamic Worlds:** This can be used to create endless procedural worlds for open-world games, where the terrain is generated as the player explores.

### Next Steps:

You can start experimenting with these examples and build on them as needed. Here's a suggestion for how to proceed:

1. **Start Small:** Try generating simple procedural meshes in Blender and applying textures.
2. **Move to AI:** Once you're comfortable, experiment with integrating neural networks into Unreal Engine or Blender.
3. **Expand Procedurally:** Finally, use C++ in Unreal Engine to create more complex AI behaviors and dynamic worlds.

Let me know if you'd like further clarification or additional resources for any of these!



I have two decisions of making use of my master. Firstly, I want to be a consultant for my own company where I want to teach people about subject programming computer graphic and intelligent system Second, I want to work with company Epic Games in Yokohama, Japan. What do you think about these option?



 **Memory updated**

Both options have great potential, and it depends on your long-term goals and personal preferences. Let's break down each one:

## Option 1: Consultant for Your Own Company

- **Advantages:**

1. **Independence:** You can design your own courses and programs, deciding how to teach and structure your work.
2. **Entrepreneurial Growth:** Running your own company can be highly rewarding. You can build a brand and scale your business over time.
3. **Flexibility:** You would have the freedom to work on your terms, potentially expanding to more clients and sectors over time.
4. **Portfolio Development:** Your Master's research could form the foundation for the content you teach, providing value to your clients.

- **Challenges:**

1. **Building a Client Base:** It might take time to establish a reputation and grow your client base. Marketing and networking will be key.
2. **Financial Risk:** Starting and maintaining a business involves financial risks, especially early on.
3. **Scaling:** Managing growth, employees, and projects as a sole proprietor can become demanding.

- **Considerations:** If you're passionate about sharing knowledge, building relationships with clients, and growing a business, this could be a rewarding path. You'll also need to be proactive in networking and marketing to make your consultancy successful.

---

## Option 2: Working with Epic Games in Yokohama, Japan

- **Advantages:**

1. **Industry Exposure:** Working with Epic Games would put you at the forefront of innovation in game development, especially with technologies like Unreal Engine.
2. **Skill Development:** You'd be exposed to cutting-edge tools and projects, learning from top professionals in the field.
3. **Stability:** A job at Epic Games would offer more financial security, benefits, and a structured work environment compared to running your own business.
4. **Networking:** You'd build connections within the industry that could open doors for future collaborations or career advancements.



5. **International Experience:** Living and working in Yokohama could be an exciting cultural experience, and Epic Games has a strong presence in the gaming industry worldwide.

- **Challenges:**

1. **Corporate Structure:** Unlike running your own company, you'll have less creative freedom and may need to work within specific boundaries or roles.
2. **Job Competition:** Landing a job at Epic Games can be competitive, especially at a prestigious location like Yokohama.
3. **Relocation:** Moving to Japan would mean adapting to a new culture, language, and work environment.

- **Considerations:** If you're interested in being part of a cutting-edge company and contributing to large-scale projects, this could be an exciting opportunity. It would also provide a stable income while letting you grow your skills in a highly respected company.

## Which Path to Choose?

- **If you value freedom, teaching, and entrepreneurship,** then the consulting route may be the better choice. It allows you to combine your research with hands-on teaching and opens up possibilities to grow a business.
- **If you want to immerse yourself in the gaming industry and work with a world-class company,** Epic Games would provide you with valuable experience, especially in computer graphics and AI, while also giving you international exposure.

You could also consider **starting as a consultant while applying to Epic Games** to see how things progress. That way, you could explore both options before fully committing.

What are your thoughts? Does one path resonate more with you right now?



I think I want to do both



 **Memory updated**

That's a great approach! Pursuing both paths can give you the flexibility to explore different opportunities while also building your expertise. Here are some strategies to help you balance both:

### 1. Start as a Consultant While Job Hunting

- **Freelance or Part-Time Work:** Consider starting your consultancy on a part-time basis while you apply for positions at Epic Games. This allows you to gain experience and start building your brand without fully committing to one path.
- **Offer Online Courses:** You could create online courses or workshops that fit around your schedule. This will allow you to teach while maintaining flexibility.

## 2. Use Your Experience to Enhance Both Roles

- **Industry Knowledge:** Working at Epic Games will provide you with valuable insights and experiences that you can incorporate into your consultancy. You can share real-world applications and examples in your teaching.
- **Networking:** The connections you make at Epic Games can help grow your consultancy. You may find potential clients or collaborators through industry events or company contacts.

## 3. Time Management

- **Set Clear Goals:** Define what you want to achieve with your consultancy and your role at Epic Games. This can help you stay focused and measure your progress.
- **Schedule Wisely:** Create a structured schedule that allows you to dedicate specific time to each pursuit. Make sure to include time for self-care to avoid burnout.

## 4. Transition Between Roles

- **Evaluate Opportunities:** If you find a fulfilling role at Epic Games, you can adjust your consultancy focus accordingly. Conversely, if your consultancy takes off, you might consider scaling back your job at Epic Games.
- **Consider Remote Work:** If your role at Epic Games allows for it, remote work could provide more flexibility to continue your consultancy.

## 5. Build a Strong Online Presence

- **Content Creation:** Share your journey, insights, and expertise through blogs, social media, or YouTube. This can help you attract clients for your consultancy while also showcasing your skills to potential employers at Epic Games.

By pursuing both paths, you'll not only gain diverse experiences but also create a unique position for yourself in the industry. It's a challenging but rewarding endeavor! If you need more specific advice on balancing these options or starting your consultancy, just let me know!



I am still planning for topics about my master I will try to read back on your suggestion



 **Memory updated**

That sounds like a solid plan! Taking the time to review and reflect on potential topics will help you find the best fit for your interests and goals. If you have any questions or need further assistance while you're exploring those topics, feel free to reach out. Good luck with your planning!



Thank you



You're welcome! If you need anything else, just let me know. Good luck with your Master's journey!



I just want to share I recently enrolled in Japanese language proficiency class and I have applied as a part-time programmer. I also apply to be a full-stack programmer and plan to migrate to another company as there are several reasons where, firstly, I don't like to be a characterization engineer for a semiconductor company. I just love to code for LabVIEW and web development other than that, I don't like it. Next, I don't like male-dominant industry where there are few female friends in the industry. Thirdly I don't like to commute to workplace everyday. I like to work from home



 **Memory updated**

It sounds like you're making some positive strides toward your goals! Here's a breakdown of your recent decisions and how they align with your interests:

## 1. Japanese Language Proficiency Class

- **Importance:** Learning Japanese will be incredibly beneficial, especially if you're planning to work in Japan with companies like Epic Games. It'll help with communication and integrating into the workplace and culture.

## 2. Part-Time Programming Role

- **Flexibility:** A part-time position can give you the experience you need while allowing time to continue your studies and explore other opportunities.
- **Relevant Experience:** This role will help you build your programming skills and possibly create networking opportunities.

## 3. Full-Stack Programmer Applications

- **Broader Skillset:** Full-stack programming allows you to work on both front-end and back-end development, which is highly valuable in today's tech landscape.
- **Career Growth:** This path can lead to more opportunities and a better understanding of the entire development process.

## 4. Migration to Another Company

- **Alignment with Interests:** Moving away from characterization engineering makes sense since it's not where your passion lies. Focus on roles that excite you, like LabVIEW and web development.
- **Work Environment:** Seeking a workplace that values diversity and offers a more balanced environment will be beneficial for your overall job satisfaction.

## 5. Preference for Remote Work

- **Work-Life Balance:** Remote work can enhance your productivity and allow for a better work-life balance. Many companies are increasingly offering remote options, especially in tech.

## Next Steps

- **Networking:** Engage with communities (online or offline) that share your interests in programming and web development. This can help you find new opportunities and support.
- **Skill Development:** Continue enhancing your programming skills, particularly in areas relevant to the jobs you're applying for.

It looks like you're taking thoughtful steps toward a career that aligns with your interests and lifestyle. If you need advice on any specific aspects, like job applications or learning resources, feel free to ask!



am I too late for a 27 years old?



Absolutely not! At 27, you're still very much in the prime of your career development. Many people change careers, pursue further education, or start new ventures at various stages of life. Here are a few points to consider:

### 1. Lifelong Learning

- **Continual Growth:** The tech industry, in particular, values skills and knowledge over age. Many people enter new fields or enhance their expertise later in life.

### 2. Diverse Experiences

- **Unique Perspective:** Your past experiences, even if they're in a different field, can provide valuable insights and skills that can be applied in your new endeavors.

### 3. Industry Trends

- **Demand for Skills:** There is a growing demand for skilled professionals in programming, AI, and computer graphics. Your decision to pursue a Master's degree and develop expertise in these areas is timely and relevant.

### 4. Networking and Community

- **Supportive Communities:** Many professionals in tech are welcoming and supportive of newcomers, regardless of age. Engaging with communities can help you build connections and find mentorship.

### 5. Successful Examples

- **Inspiring Stories:** There are countless examples of people who found success after 27, whether through education, career shifts, or entrepreneurship.

## 6. Focus on Your Goals

- **What Matters Most:** Your passion, motivation, and commitment to your career will make a bigger difference than your age. Focus on what you want to achieve, and take actionable steps towards those goals.

Starting or changing your career at 27 is not too late at all; in fact, it's a great time to set new goals and pursue what truly excites you. Keep moving forward, and don't hesitate to reach out for support or resources along the way!



I plan that when I have a good portfolio for my company, I want to join seminar or conference to promote my company and do networking.



 Memory updated

That's a fantastic strategy! Participating in seminars and conferences can be incredibly beneficial for promoting your company and expanding your network. Here are some tips to maximize your experience:

### 1. Choose Relevant Events

- **Focus on Your Niche:** Look for seminars and conferences that are specifically related to computer graphics, artificial intelligence, programming, or entrepreneurship.
- **Attend Industry Leaders:** Events hosted by or featuring leading companies can provide insights into current trends and allow you to network with key players in the field.

### 2. Prepare Your Portfolio

- **Showcase Your Work:** Ensure your portfolio highlights your best projects, particularly those that align with the themes of the events you plan to attend.
- **Elevator Pitch:** Prepare a concise and engaging elevator pitch to explain what your company does and the value you offer.

### 3. Networking Strategies

- **Connect Before the Event:** Use platforms like LinkedIn to connect with attendees or speakers ahead of time. This can lead to meaningful conversations during the event.
- **Engage in Workshops:** Participate in interactive workshops or discussions, as these settings often foster more personal connections.

### 4. Follow-Up After the Event

- **Maintain Connections:** After the conference, reach out to people you met with a personalized message, referencing your conversation to remind them of your connection.
- **Share Your Learnings:** Share insights from the event on your social media or blog to stay engaged with your network and demonstrate your expertise.

## 5. Consider Speaking Opportunities

- **Present Your Work:** As you grow your portfolio, consider applying to be a speaker or panelist at conferences. Sharing your expertise can raise your profile and attract potential clients or collaborators.

## 6. Continuous Engagement

- **Stay Involved:** Join relevant professional organizations or online communities to stay connected with industry trends and networking opportunities.

By actively engaging in these events, you'll not only promote your company but also establish valuable relationships that can support your growth. It sounds like you're on the right track! If you need help finding relevant events or preparing your materials, just let me know!



yes please



Great! Here are some steps to help you find relevant seminars and conferences, along with resources to prepare your materials:

### Finding Relevant Events

#### 1. Industry Conferences and Expos:

- **SIGGRAPH:** A leading conference on computer graphics and interactive techniques, offering opportunities to network and learn about the latest advancements in the field.
- **GDC (Game Developers Conference):** A major event for professionals in the gaming industry, covering everything from programming to design and marketing.
- **AI & Machine Learning Conferences:** Look for events like NeurIPS, CVPR, or local meetups that focus on AI and deep learning, as they often have tracks related to graphics and visualization.

#### 2. Local Meetups and Workshops:

- **Meetup.com:** Search for local groups focused on programming, AI, computer graphics, or entrepreneurship. These can provide smaller, more intimate networking opportunities.
- **Tech Hubs and Incubators:** Check if there are local tech hubs or incubators that host events or workshops related to your fields of interest.

#### 3. Online Events:

- **Webinars:** Many organizations host webinars on specific topics in tech, which can be great for learning and networking without needing to travel.
- **Virtual Conferences:** Explore platforms like Hopin or Eventbrite for virtual events that cater to your interests.

#### 4. Professional Organizations:

- Join organizations like the ACM (Association for Computing Machinery) or IEEE (Institute of Electrical and Electronics Engineers) to access event listings and networking opportunities.

## Preparing Your Portfolio and Materials

### 1. Portfolio Development:

- **Select Projects:** Choose a variety of projects that showcase your skills in computer graphics and AI, ideally highlighting practical applications or innovations.
- **Presentation Format:** Consider creating a digital portfolio (like a personal website) that's visually appealing and easy to navigate. Include descriptions, challenges, and results for each project.

### 2. Creating Marketing Materials:

- **Business Cards:** Design professional business cards to hand out during networking events. Include your name, company, contact information, and a QR code linking to your portfolio.
- **Brochures or Flyers:** Consider creating a brochure that outlines your services, skills, and notable projects. This can be distributed at events.

### 3. Practice Your Elevator Pitch:

- **Concise and Engaging:** Craft a 30-60 second pitch that succinctly explains who you are, what your company does, and what sets you apart. Practice delivering it until it feels natural.

### 4. Networking Follow-Up:

- **Create a Template:** Prepare a follow-up email template to send after events, reminding people of your conversation and expressing interest in staying in touch.

## Action Steps

- **Start by researching upcoming events** that align with your goals, and mark your calendar.
- **Begin developing your portfolio and materials**, focusing on clarity and visual appeal.
- **Network with professionals in your field** through platforms like LinkedIn to discover additional events and opportunities.

If you need help with specific resources or further guidance on any of these steps, just let me know!



yes please



Here are some specific resources and tools to help you find relevant events and prepare your materials:

## Finding Relevant Events

### 1. Conferences and Industry Events:

- **SIGGRAPH:**
  - [SIGGRAPH Conference Website](#) - Find the latest information on upcoming conferences and events in the computer graphics field.
- **GDC (Game Developers Conference):**
  - [GDC Website](#) - Check for the latest updates on the conference and sessions that align with your interests.
- **NeurIPS:**
  - [NeurIPS Conference](#) - A major conference focused on machine learning and AI.
- **CVPR:**
  - [CVPR Conference](#) - Focused on computer vision and pattern recognition.

## 2. Local Meetups and Workshops:

- **Meetup.com:**
  - [Meetup](#) - Search for tech-related meetups in your area. Use keywords like "computer graphics," "AI," or "programming."

## 3. Online Events:

- **Eventbrite:**
  - [Eventbrite](#) - Search for tech webinars, virtual conferences, and local events related to your fields of interest.
- **Webinars:**
  - Websites like [Coursera](#), [edX](#), and [Udacity](#) often have webinars and workshops on relevant topics.

## 4. Professional Organizations:

- **ACM:**
  - [ACM](#) - Join to access resources and events related to computing and information technology.
- **IEEE:**
  - [IEEE](#) - Consider becoming a member for access to conferences, publications, and networking opportunities.

# Preparing Your Portfolio and Materials

## 1. Portfolio Development:

- **Website Builders:**
  - [Wix](#) or [Squarespace](#) - These platforms allow you to create professional websites to showcase your portfolio.
- **GitHub:**
  - Create a GitHub repository to showcase your coding projects and technical skills.

## 2. Creating Marketing Materials:

- **Business Card Design:**
  - Use services like [Vistaprint](#) or [Canva](#) to design and print professional business cards.
- **Brochure or Flyer:**
  - Use [Canva](#) or [Adobe Express](#) for creating marketing materials. They offer templates and easy design tools.

## 3. Elevator Pitch Preparation:



- **Practice:**
  - Record yourself delivering your pitch and play it back to refine your delivery. You can also practice in front of friends or colleagues for feedback.

#### 4. Networking Follow-Up:

- **Email Templates:**
  - Create a follow-up email template like this:

```
vbnet

Subject: Great Connecting at [Event Name]

Hi [Name],

It was great meeting you at [Event Name]. I enjoyed our conversation about [specific topic]. I'd love to stay in touch and explore any potential collaboration.

Here's a link to my portfolio: [Your Portfolio Link].

Best,
[Your Name]
[Your Contact Information]
```

## Action Plan

1. **Research Events:** Start by checking the websites mentioned above for upcoming events that interest you.
2. **Develop Your Portfolio:** Begin building your online portfolio and preparing your marketing materials.
3. **Practice Networking:** Reach out to connections on LinkedIn or other platforms, and consider attending local meetups.

If you need more specific resources, tools, or examples in any area, let me know! I'm here to help.



also. I forgot to ask, since I want to do master combining Computer Graphic and Intelligent System, where should I find the reference? technical paper, journal ,organization and research paper? where should I publish the technical paper?



 **Memory updated**

Finding the right references and publication venues is crucial for your Master's research. Here's a guide to help you with both:

## Finding References

1. **Academic Journals:**

- **ACM Transactions on Graphics (TOG):** A premier journal focusing on computer graphics research.
- **IEEE Transactions on Visualization and Computer Graphics (TVCG):** Covers topics related to computer graphics and visualization.
- **Artificial Intelligence Journal:** Publishes articles on AI methodologies that can be applied in computer graphics.

## 2. Conferences:

- **SIGGRAPH:** This is one of the most significant conferences for computer graphics research. The proceedings can provide a wealth of current research.
- **Eurographics:** Focuses on computer graphics, including both applications and theoretical research.
- **ICCV (International Conference on Computer Vision):** A top conference in the field of computer vision and AI, where many graphics-related papers are presented.

## 3. Technical Papers:

- Use platforms like **Google Scholar**, **ResearchGate**, or **arXiv** to find technical papers related to your topic. You can search for keywords that relate to your interests in computer graphics and intelligent systems.

## 4. Books:

- Look for academic books on specific topics related to computer graphics and AI. Publishers like Springer, Wiley, and Elsevier often have comprehensive texts on these subjects.

## 5. University Libraries:

- Many university libraries have access to journals, papers, and databases like IEEE Xplore, ACM Digital Library, and JSTOR. If you have access to a university library, take advantage of these resources.

## 6. Professional Organizations:

- Organizations like ACM and IEEE provide access to publications, technical papers, and resources for members.

# Publishing Technical Papers

## 1. Conferences:

- Conferences like SIGGRAPH and Eurographics often have submission opportunities for technical papers. Presenting at these conferences can provide visibility and networking opportunities.
- Look for calls for papers (CFPs) for upcoming conferences in your field.

## 2. Journals:

- After your Master's research, consider submitting your findings to academic journals like those mentioned above. Be sure to review the submission guidelines for each journal, as they can vary significantly.

## 3. Workshops:

- Some conferences also host workshops where you can present preliminary findings or specific projects related to your research. This can be a good starting point for feedback and networking.

#### 4. Thesis/Dissertation:

- Depending on your university's requirements, you may publish your Master's thesis as a technical paper or submit it to a journal if the research is substantial.

#### 5. Open Access Platforms:

- Consider submitting your work to open-access platforms that focus on your research area, like PLOS ONE or MDPI journals, which have broader reach and accessibility.

## Next Steps

- **Research Events:** Check the websites of the conferences and journals for upcoming submission deadlines and calls for papers.
- **Start Building a Bibliography:** As you read relevant papers and articles, start compiling a bibliography to track your references.
- **Engage with the Community:** Follow researchers in your field on platforms like Twitter and LinkedIn to stay updated on recent publications and events.

If you have specific topics in mind or need help navigating any of these resources, feel free to ask!



Allright I think thats all. can you save this conversation?



I can't save the conversation in the way you're thinking, but I can remember key points and details you share for future reference. Just let me know what specific information you want me to keep, and I'll save that for you!



I want you to save the keypoints like Computer Grahic and AI



 **Memory updated**

Got it! I've saved that you're focusing on combining Computer Graphics and Artificial Intelligence for your Master's research. If there's anything else you'd like to add or remember, just let me know!



I think That's all.



Perfect! If you have any more questions or need assistance in the future, feel free to reach out. Good luck with your Master's research and all your plans ahead! Have a great day!

This conversation is archived. To continue, please unarchive it first.

 Unarchive