

Unreal Engine 5: Minesweeper AI Chat Plugin - Documentation

Version: 1.0

Author: Mukund Pareek

Date: 13th February 2025

Engine Version: Unreal Engine 5.3

1. Introduction

The **Minesweeper AI Chat Plugin** is a **Slate-based Unreal Engine 5 plugin** that allows users to generate **Minesweeper grids** using **AI-generated responses**. This plugin integrates with **Mistral AI** via HTTP requests to dynamically create Minesweeper grids based on user input.

Why This Plugin?

- **No UMG (User Widget Blueprint) dependency** → Fully built using **Slate UI**
 - **AI-powered grid generation** → Dynamically generates a **Minesweeper board** based on user instructions
 - **Interactive & Dynamic UI** → Users can regenerate grids by simply **re-entering a prompt**
-

2. Installation & Setup

2.1 Prerequisites

- ✓ Unreal Engine 5.3+
- ✓ C++ Unreal Engine Project (Blueprint-only projects are **not supported**)
- ✓ Mistral AI API Key (Replace in `MinesweeperAIRequest.cpp`)

2.2 Installation Steps

- 1 Copy the Plugin Files:

- Place the `MinesButton` plugin folder inside `/Plugins/` in your project directory.

2

Enable the Plugin:

- Open **Unreal Engine** → **Edit** → **Plugins**
- Search for `MinesButton` and enable it

3

Restart Unreal Engine

- This is required to load the plugin properly

2.3 Building the Plugin

- If using **Visual Studio 2022**:
 - Open the `.uproject` file in **Visual Studio**
 - Right-click → Select "**Regenerate Visual Studio Project Files**"
 - **Build the project** (`Ctrl + Shift + B`)
- If using **Unreal Engine**:
 - Open `JoinTask.uproject`
 - Click **Compile**

3. How to Use

3.1 Opening the Chat Window

1. Click the **Minesweeper AI** button in **Unreal Editor's** toolbar
2. A chat window appears

3.2 Generating a Minesweeper Grid

1. **Enter a request:**
 - Example: `"Generate a 3×3 Minesweeper grid with 1 bomb"`
2. **Press Enter**
3. **View the AI-generated grid**

3.3 Regenerating the Grid

- Simply **enter a new prompt** to get a fresh AI-generated grid.

4. Project Structure

MinesButton Plugin Structure

```
/MinesButton
| — /Public
|   | — MinesButton.h      # Main plugin module
|   | — SChatWidget.h      # Slate Chat UI
|   | — MinesweeperAIRequest.h # Handles AI requests
|   | — WindowHUD.h        # HUD for managing UI
| — /Private
|   | — MinesButton.cpp     # Plugin logic
|   | — SChatWidget.cpp     # Chat UI logic
|   | — MinesweeperAIRequest.cpp # HTTP request handler
|   | — WindowHUD.cpp       # HUD functionality
```

5. Code Architecture

5.1 Key Components

Slate UI (`SChatWidget`)

- Handles the chat input, AI response display, and UI updates
- Uses **Slate UI Widgets** (`SEditableTextBox` , `SButton` , `STextBlock`)

AI Requests (`MinesweeperAIRequest`)

- Uses **HTTP requests** to send user input to **Mistral AI**
- Receives a **Minesweeper grid** as **JSON** and processes the response

HUD (`WindowHUD`)

- Manages the **chat window** inside the **Unreal Engine viewport**
- Prevents **multiple UI instances** from overlapping

6. AI Request & Response Handling

6.1 Sending AI Requests (`MinesweeperAIRequest.cpp`)

```

void U MinesweeperAIRequest::SendAIRequest(FString UserQuery)
{
    FHttpModule* Http = &FHttpModule::Get();
    if (!Http) return;

    TSharedRef<IHttpRequest, ESPMode::ThreadSafe> Request = Http->CreateRequest();
    Request->SetURL("https://api.mistral.ai/v1/chat/completions");
    Request->SetVerb("POST");
    Request->SetHeader("Content-Type", "application/json");
    Request->SetHeader("Authorization", "Bearer <Your-API-Key>");

    FString RequestBody = FString::Printf(TEXT(R"({"model": "mistral-small",
    "messages": [
        {"role": "user", "content": "%s"}], "max_tokens": 100})"), *UserQuery);

    Request->SetContentAsString(RequestBody);
    Request->OnProcessRequestComplete().BindUObject(this, &UMinesweeperAIRequest::OnAIResponse);
    Request->ProcessRequest();
}

```

6.2 Handling AI Responses (**OnAIResponse**)

- Extracts JSON response
- Parses AI-generated grid
- Updates UI with the grid

7. UI Implementation

7.1 Chat UI (**SChatWidget.cpp**)

```

void SChatWidget::Construct(const FArguments& InArgs)
{
    AIRequest = NewObject<UMinesweeperAIRequest>();
    AIRequest->OnResponseReceived.BindRaw(this, &SChatWidget::HandleA

```

```

IResponse);

SAssignNew(GridBox, SVerticalBox);

ChildSlot
[
    SNew(SVerticalBox)
        + SVerticalBox::Slot().AutoHeight()
        [
            SNew(STextBlock).Text(FText::FromString("Ask AI for a Minesw
eeper Grid"))
        ]
        + SVerticalBox::Slot().AutoHeight()
        [
            SAssignNew(InputTextBox, SEditableTextBox)
                .HintText(FText::FromString("Enter your request..."))
                .OnTextCommitted(this, &SChatWidget::OnChatSubmitted)
            ]
        + SVerticalBox::Slot().AutoHeight()
        [
            SAssignNew(ResponseText, STextBlock).Text(FText::FromStrin
g("AI Response will appear here"))
        ]
        + SVerticalBox::Slot().AutoHeight()
        [
            SAssignNew(PlayButton, SButton)
                .Text(FText::FromString("Play"))
                .OnClicked(this, &SChatWidget::GenerateMinesweeperBoar
d)
                .IsEnabled(false)
            ]
        + SVerticalBox::Slot().AutoHeight()
        [
            GridBox.ToSharedRef()
        ]
    ];
}

```

8. Troubleshooting & Debugging

Issue	Solution
Plugin doesn't appear	Ensure plugin is enabled in Edit → Plugins
Build Fails	Run Regenerate Project Files in Visual Studio
API Key Error	Replace API key in <code>MinesweeperAIRequest.cpp</code>
AI Response Incorrect	Try a more specific prompt

9. Future Improvements

 **Upcoming Advanced Features:**

- ✓ **Better AI Grid Formatting** – Ensure cleaner, properly formatted grids
- ✓ **Interactive Minesweeper Gameplay** – Clickable buttons that allow playing directly
- ✓ **Grid Customization Options** – Set grid size & bomb count via UI

10. Conclusion

This plugin **demonstrates the power of AI-generated content** in Unreal Engine 5. By using **Mistral AI**, we can dynamically create **Minesweeper grids**, with a **fully custom UI built in Slate**. The current version provides a **solid foundation** for **future expansions** such as **interactive gameplay mechanics**.

 **Hope I get selected for the job!** 😊