```python
import os

from swarms_memory import ChromaDB

from swarms import Agent, OpenAIChat, AgentRearrange

# Initilaize the chromadb client
chromadb = ChromaDB(
    metric="cosine",
    output_dir="swarms_framework_onboardig_agent",
    docs_folder="docs",  # Folder of your documents
    n_results=1,
    limit_tokens=1000,
)

# Get the OpenAI API key from the environment variable
api_key = os.getenv("OPENAI_API_KEY")

# Create an instance of the OpenAIChat class
model = OpenAIChat(
    api_key=api_key,
    model_name="gpt-4o-mini",
    temperature=0.1,
)
```

```python
# Initialize the concept understanding agent
concept_agent = Agent(
    agent_name="Concept-Understanding-Agent",
    system_prompt="You're purpose is to understand the swarms framework conceptually and architecturally, you'll work with the code generation agent to generate code snippets",
    agent_description="Agent for understanding concepts",
    llm=model,
    max_loops="auto",
    autosave=True,
    verbose=True,
    saved_state_path="concept_agent.json",
    interactive=True,
    context_length=160000,
    memory_chunk_size=2000,
)

# Initialize the code generation agent
code_agent = Agent(
    agent_name="Code-Generation-Agent",
    system_prompt="You're purpose is to generate code snippets for the swarms framework, you'll work with the concept understanding agent to understand concepts.",
    agent_description="Agent for generating code",
    llm=model,
    max_loops="auto",
    autosave=True,
    verbose=True,
```

```python
    saved_state_path="code_agent.json",

    interactive=True,

    context_length=160000,

    memory_chunk_size=2000,

)




# Swarm

swarm = AgentRearrange(

    agents=[concept_agent, code_agent],

    flow=f"{concept_agent.agent_name} -> {code_agent.agent_name}",

    max_loops=1,

    memory_system=chromadb,

)


# Run

swarm.run(

    "Let's understand the agentrearrange class in the swarms framework"

)
```