Integrating Generative AI into existing game

REPORT

, *** ** 1155****

, *** ** 1155****

Table of content

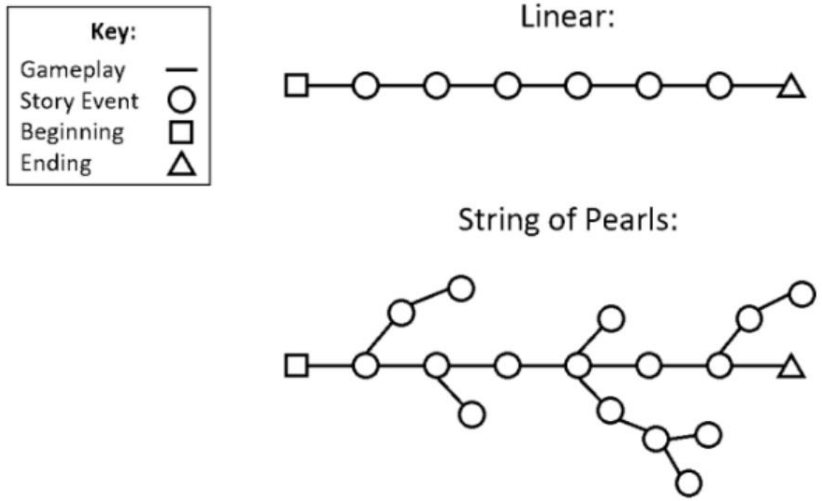
Background

- Recap
  - Linear storyline: Straight forward game story with 1 ending
  - Branching storyline: Player action affect the plot of story, usually have N ending

	Advantage	Disadvantage
Linear storyline	Easy to make	Limit the freedom of player to explore the game
Branching storyline	Provide more dynamic gaming experience	Costly to design

Background

- Recap
  - Made a prototype detective game
  - Aim to provide more dynamic experience (different dialog)
  - Attempt branching storyline



Background

- Trend of using AI agents in game development
  - Show how AI can improve Player-NPC interaction instead of controlling the storyline
  - More complicated game system = More difficult to implement AI agent

prototype gameplay with Ubisoft NEO NPC

[4] C. Brown, "Ubisoft NEO NPCs Gameplay: AI-Powered Characters in Video Games," 21 3 2024. [Online]. Available: https://www.youtube.com/watch?v=1od2pIs9220. [Accessed 4 2024].

Background

- Trend of using AI agents in game development
  - Show how AI can react differently with player's strategic planning
  - new possibilities for emergent gameplay

NVIDA Game demo to show the difference of AI response in the same scene

[6] A. Burnes, "NVIDIA Digital Human Technologies Bring AI Game Characters To Life," 19 3 2024. [Online]. Available: https://www.nvidia.com/en-us/geforce/news/nvidia-ace-gdc-gtc-2024-ai-character-game-and-app-demo-videos/. [Accessed 4 2024].

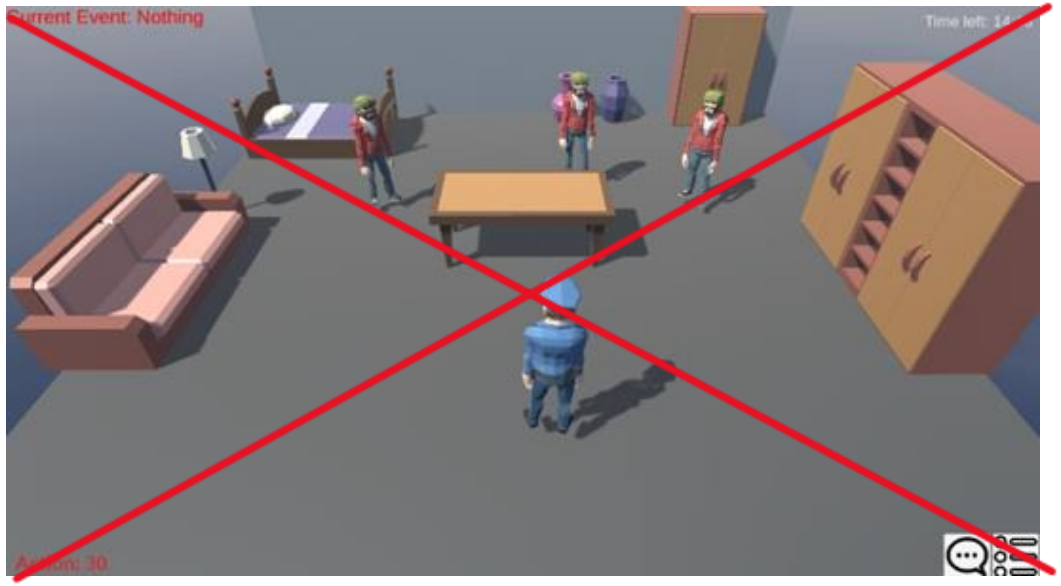
Background

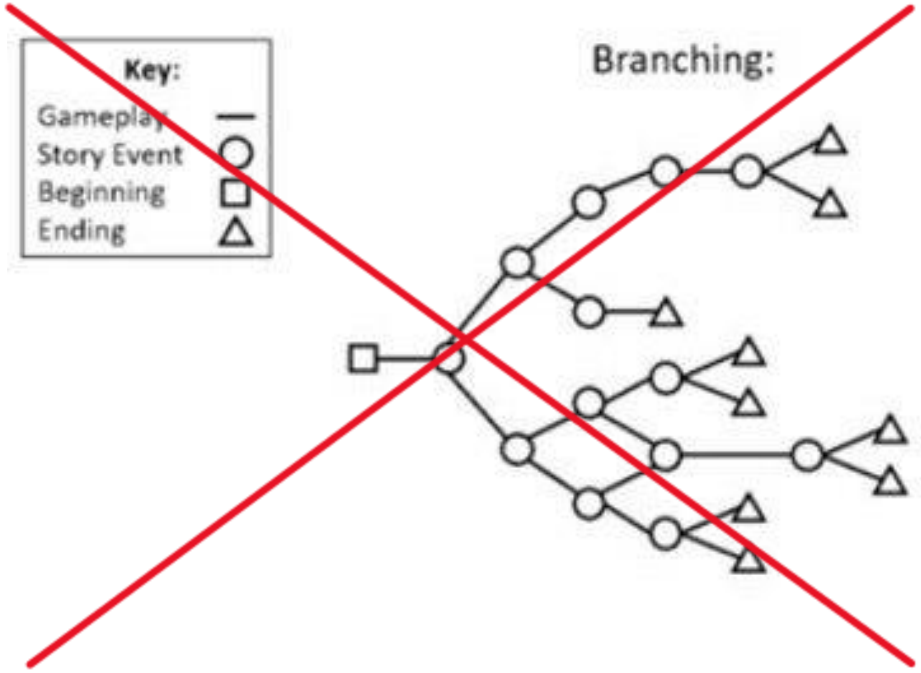
- Trend of using AI agents in game development
  - Too many parts controlled by generative AI results in an unstable player experience

SQUARE ENIX: The AI technology-based remake game and its review

Background

- Challenge
  - Gameplay is too complicated to allow generative AI to fully control the gameplay
  - Time & resources constrain





Background

- Objective: Show how to build, convert and design a game which embedded with generative AI NPC.

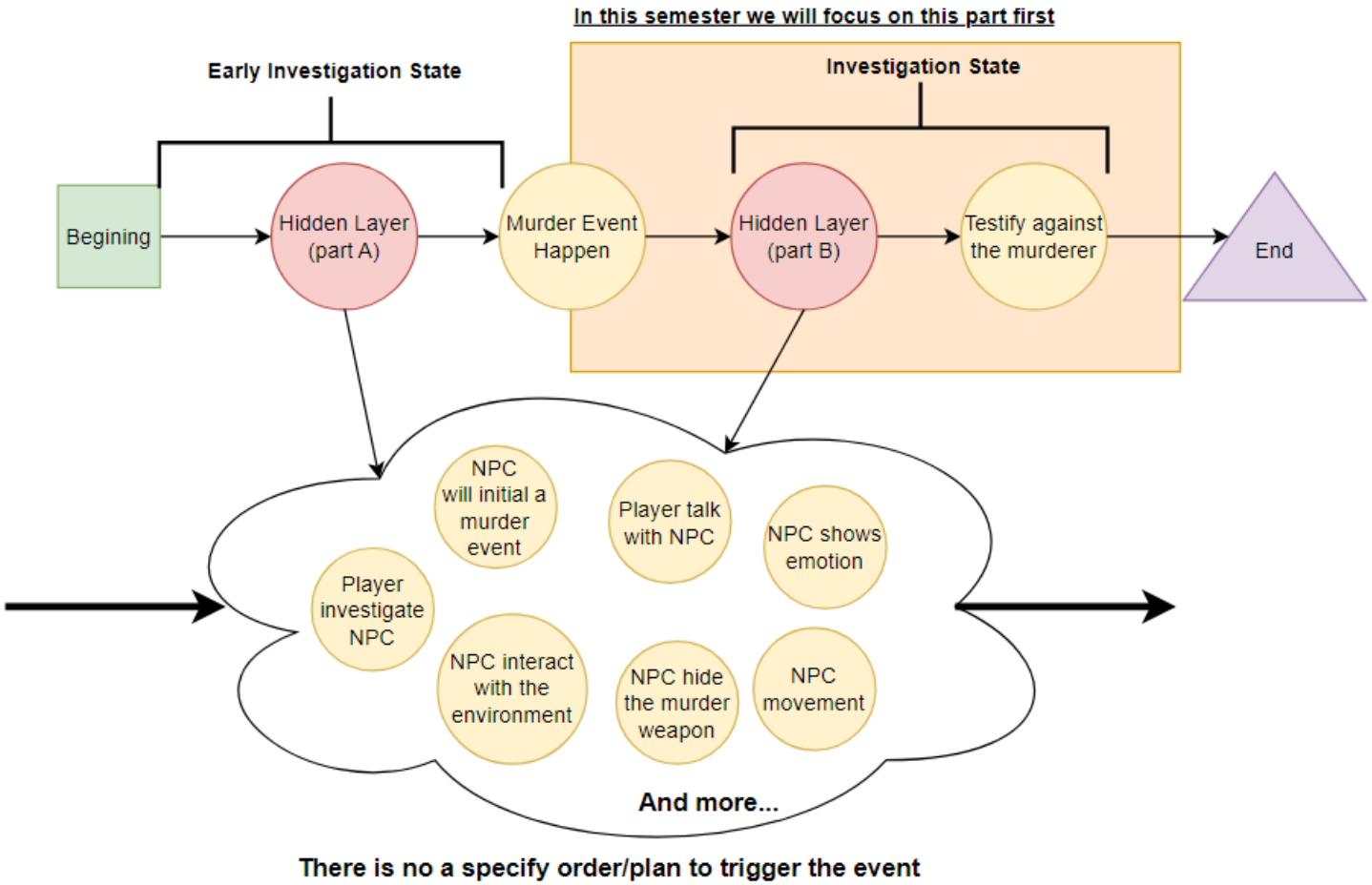
  - Observe and estimate which game component can be replaced by AI agent
  - Implement AI agent into certain game component and evaluate how it benefits the gameplay

Methodology

- Game engine: Unity
  - Provide extensive cross-platform support and features
  - Strong community support system
  - Create better user experience by lowering the effort of doing 3D rendering, user interfaces by ourselves

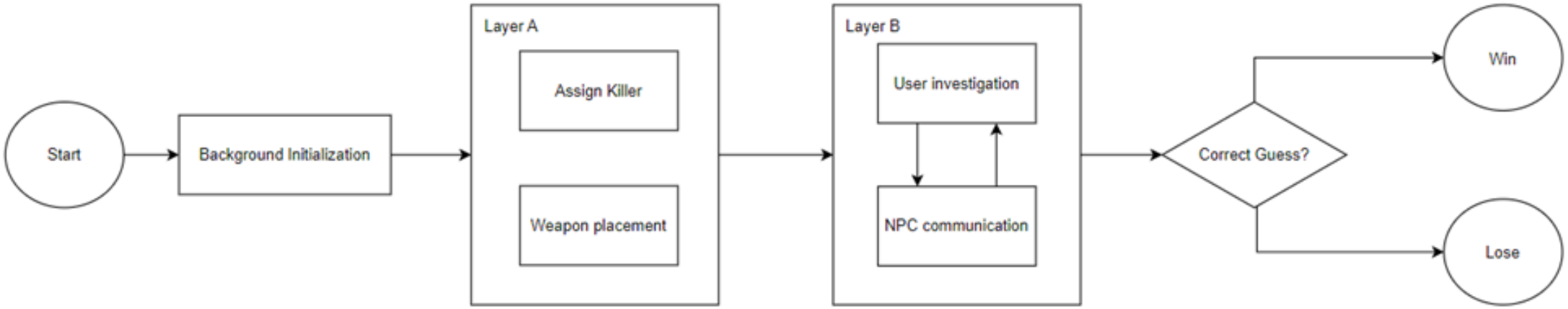
Methodology

- Recap: Original Game structure



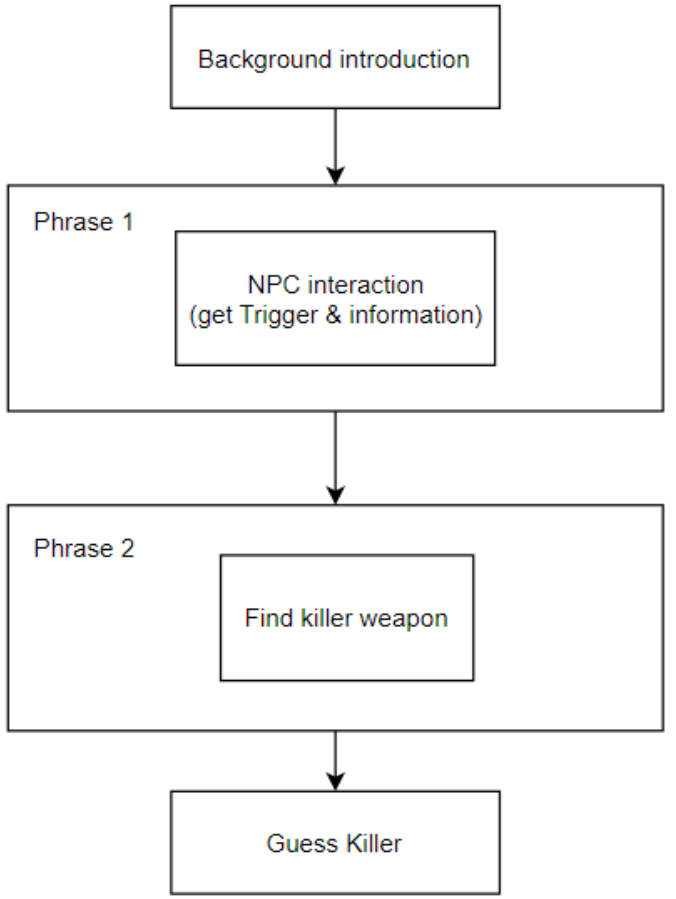
Methodology

- Game Structure



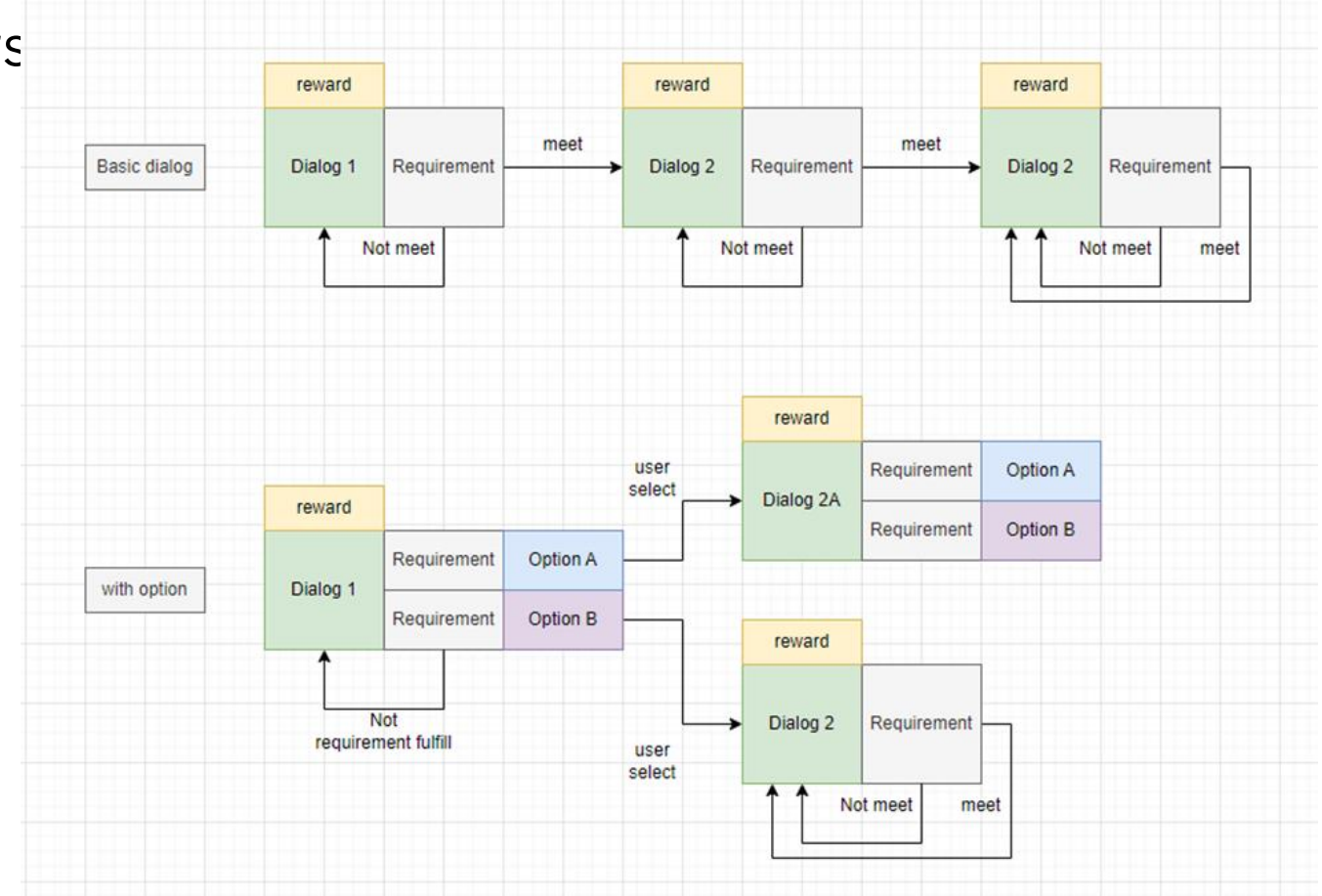
Methodology

- Player experience



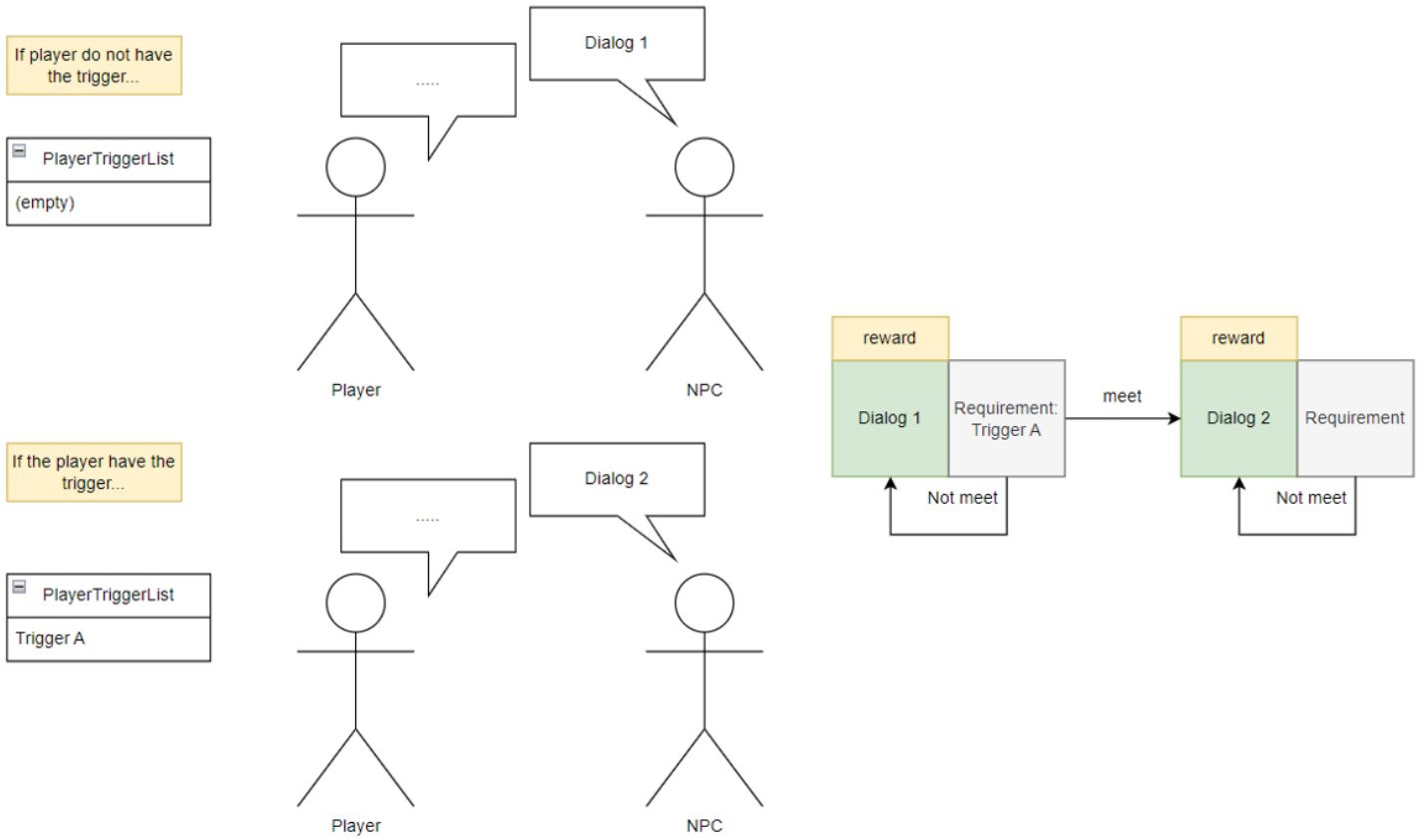
Methodology

- Dialog system
  - Each dialog as an object can chain together to form a complex convers



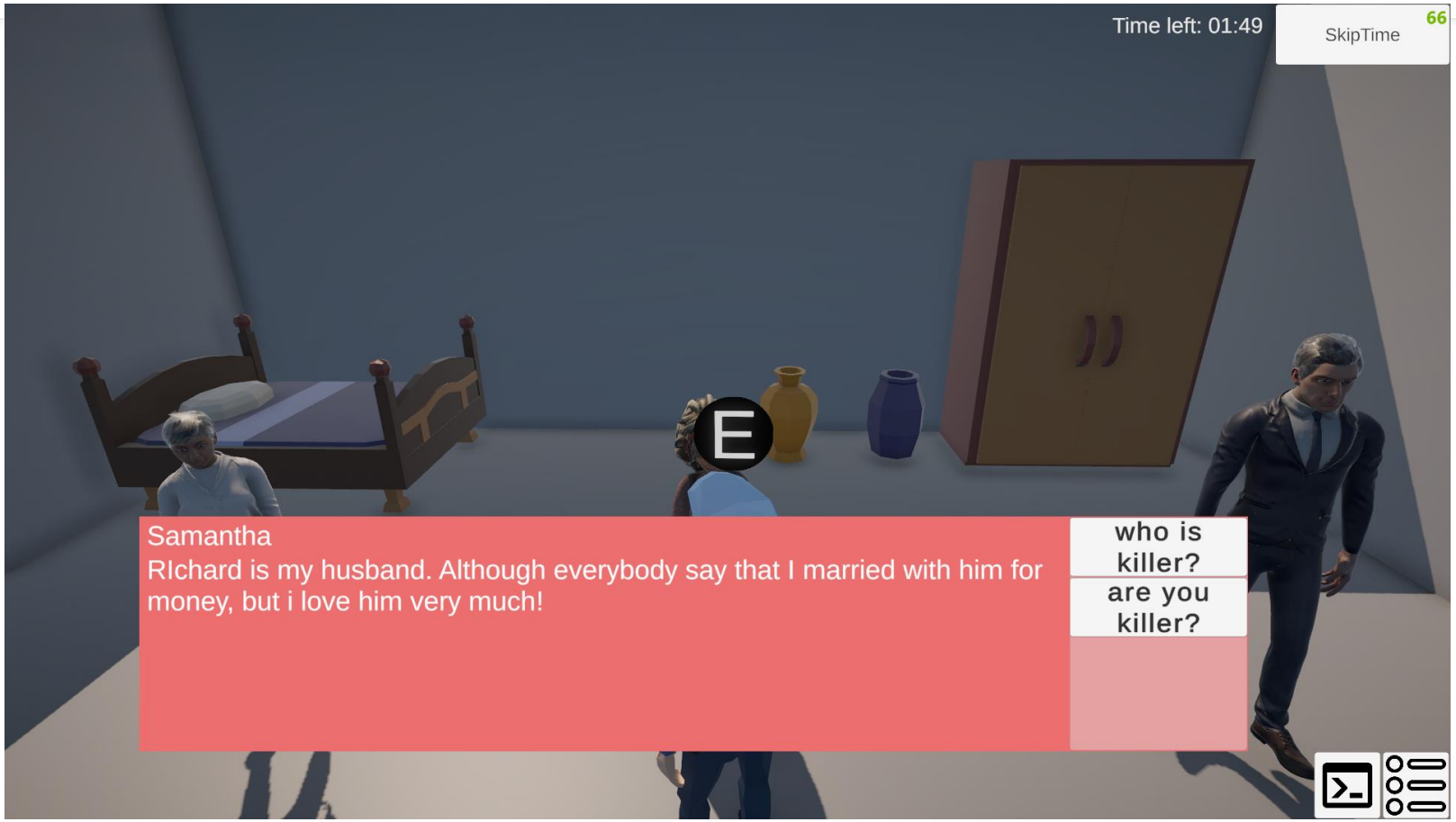
Methodology

- Trigger System
  - Act as a key to unlock new dialog or new item

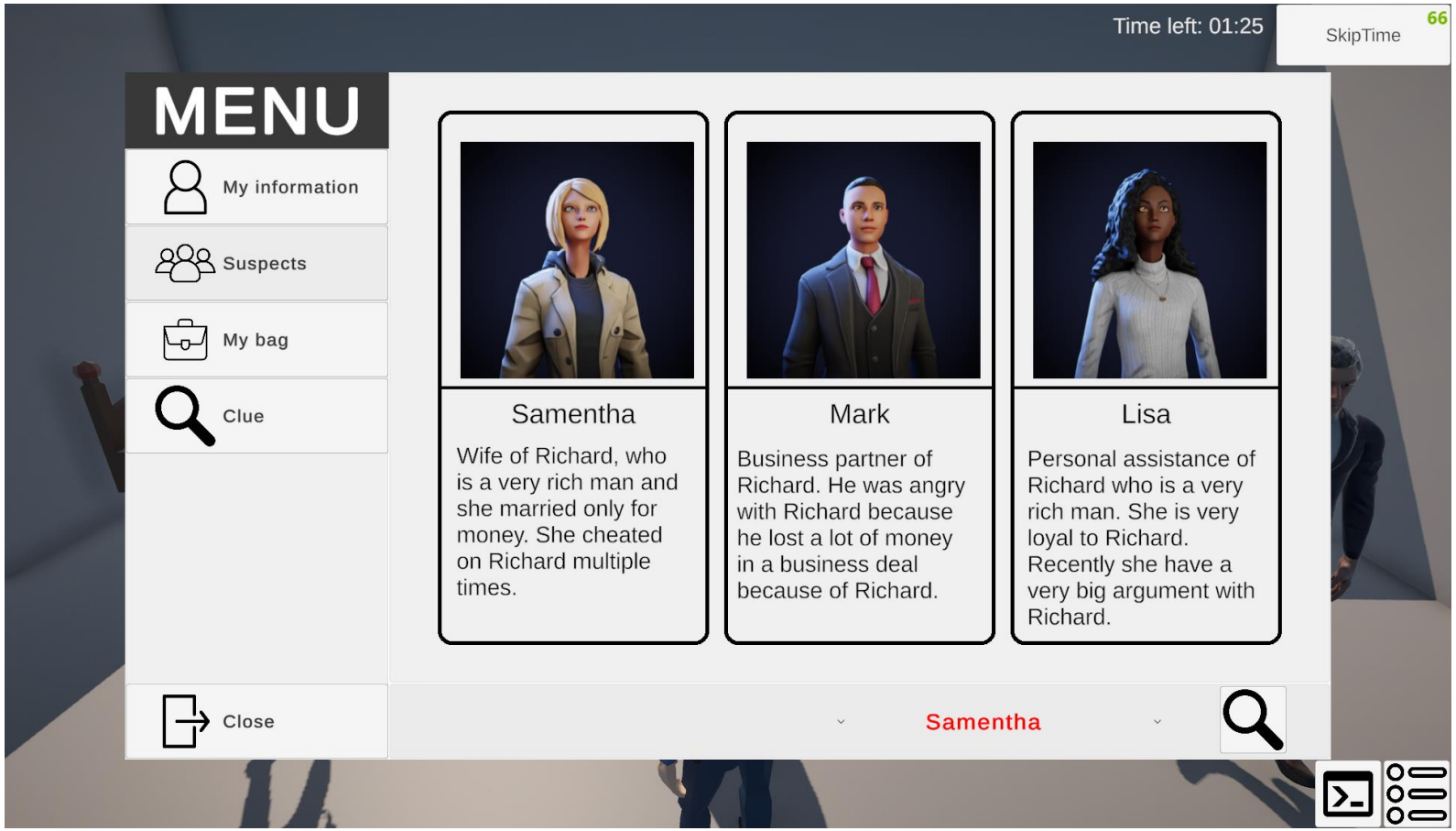


Methodology

Methodology

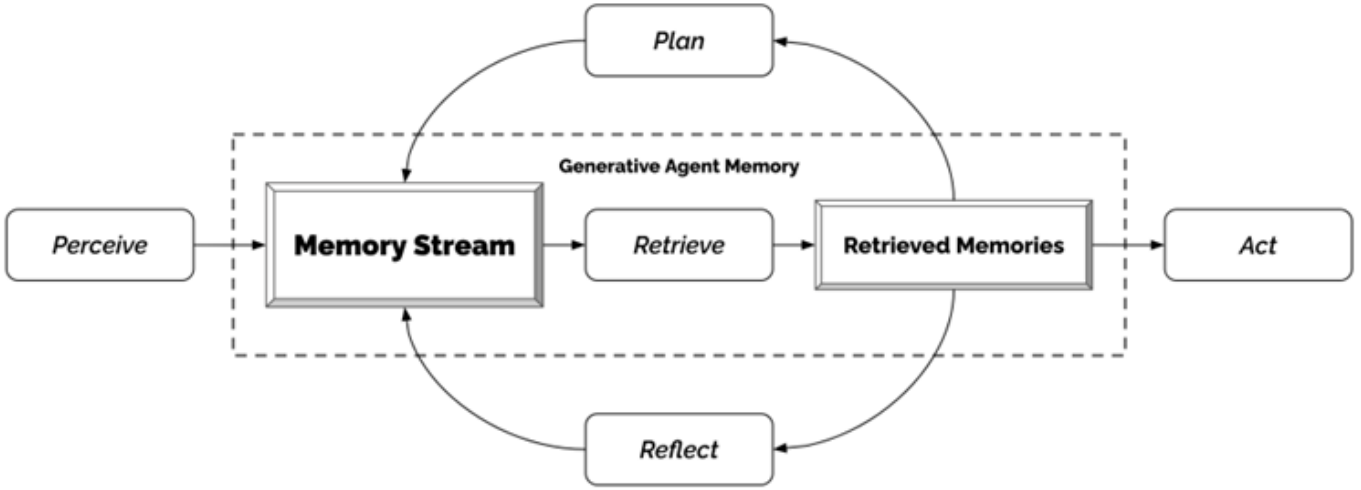


Methodology



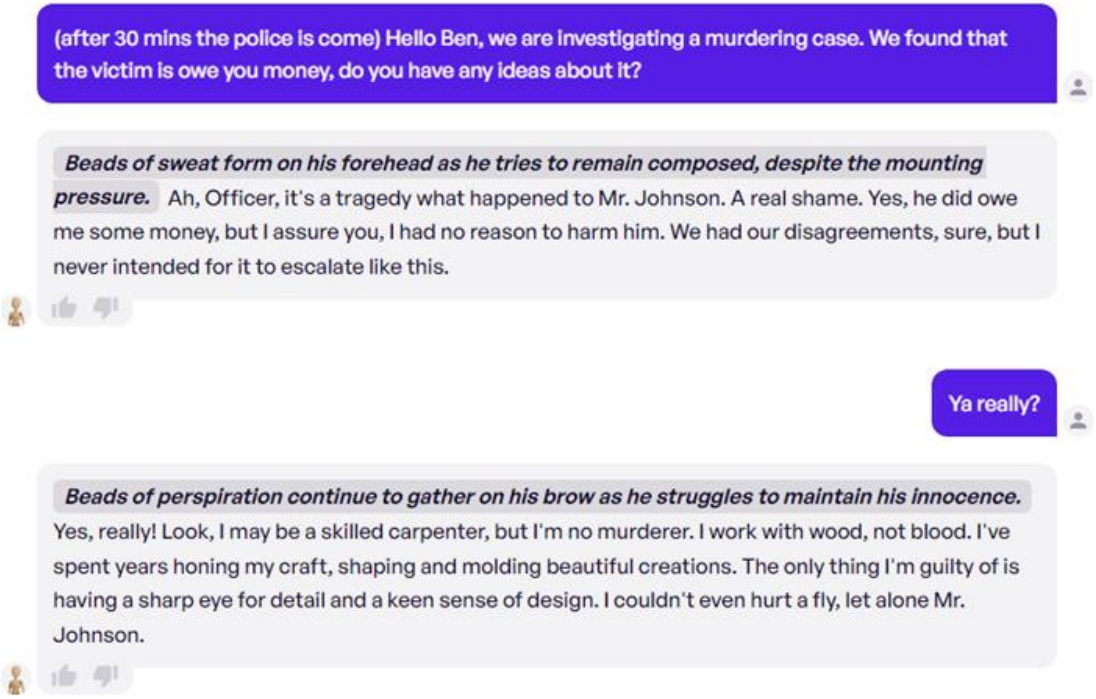
Methodology

- NPC can communicate, interact with game object and have dynamic response to different environment
- Memory allows AI to have human-like behavior



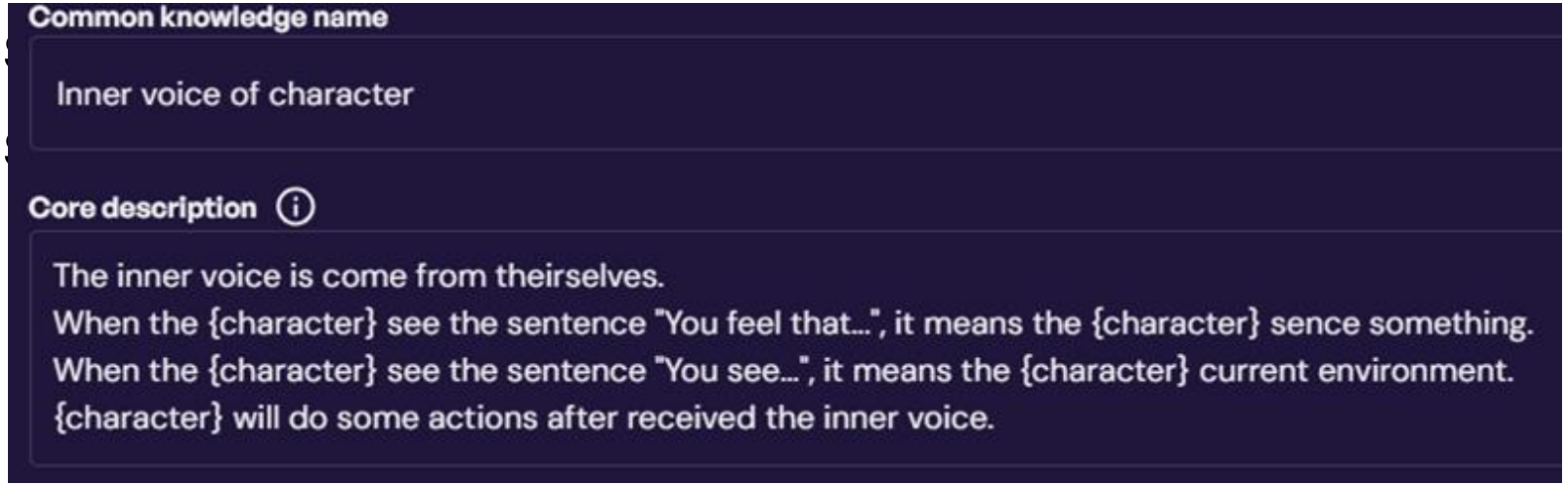
Methodology

- Short-term memory for AI agents



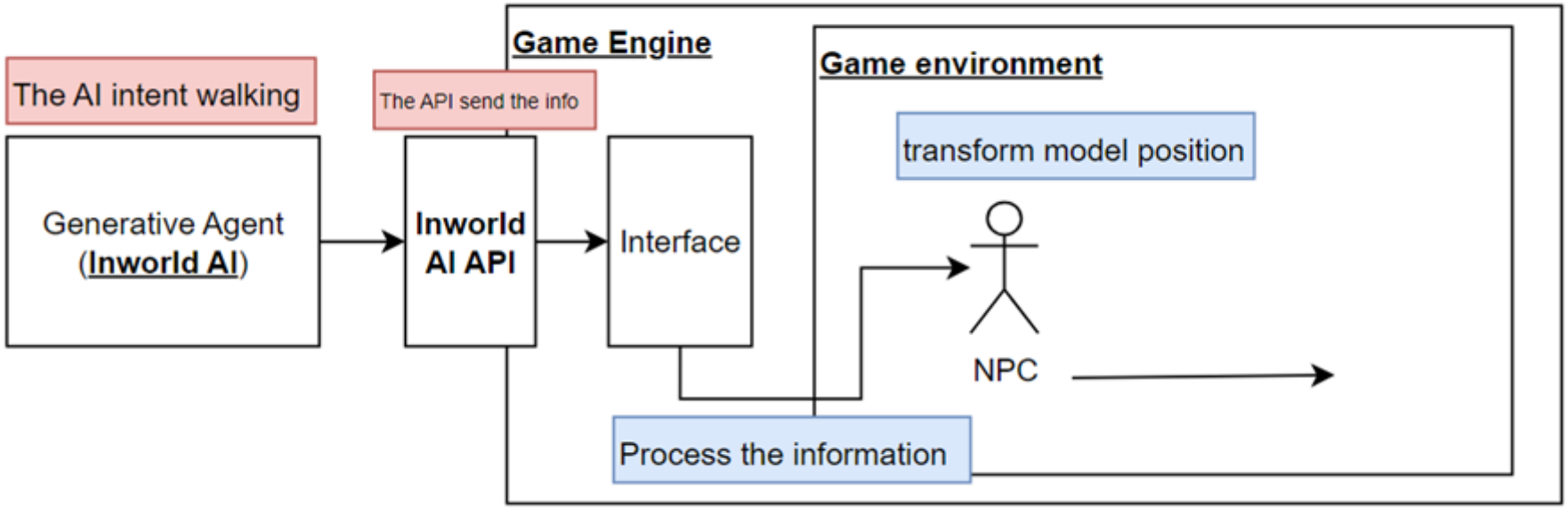
Methodology

- AI agent is able to know the information of the game environment
- They have corresponding personality, goal, relationship to other NPC
- U
- U



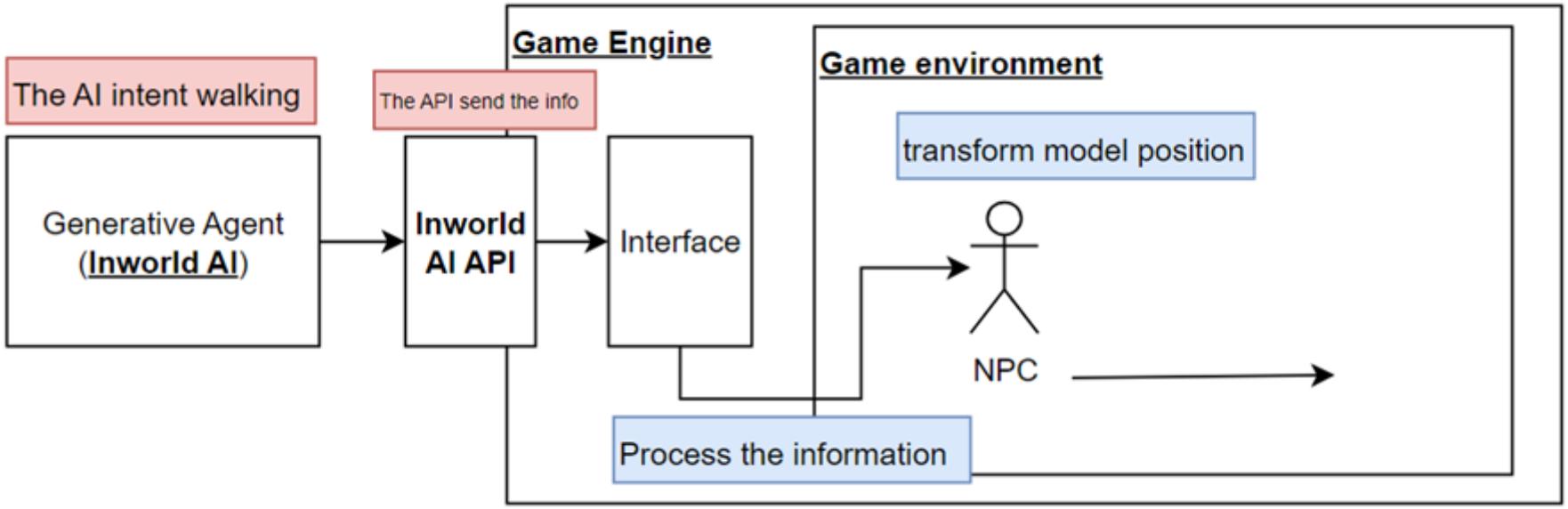
Methodology

- How AI agent send data to game



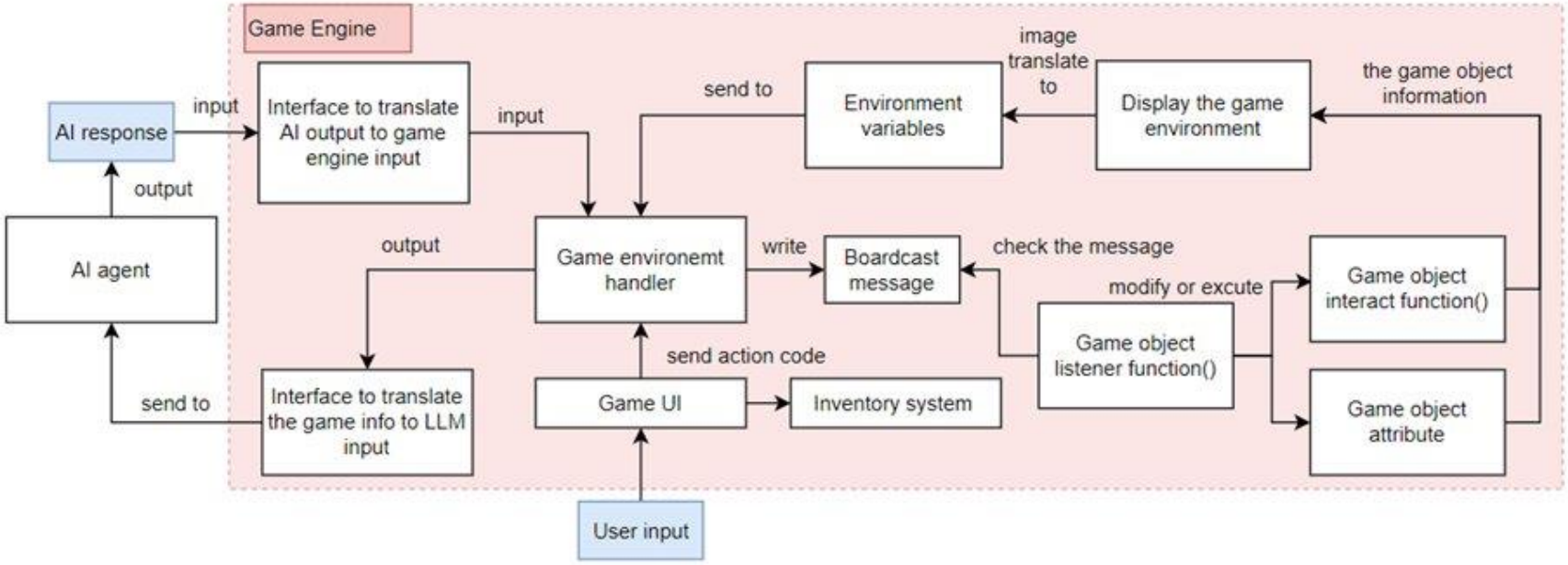
Methodology

- How game environment send data to AI agent



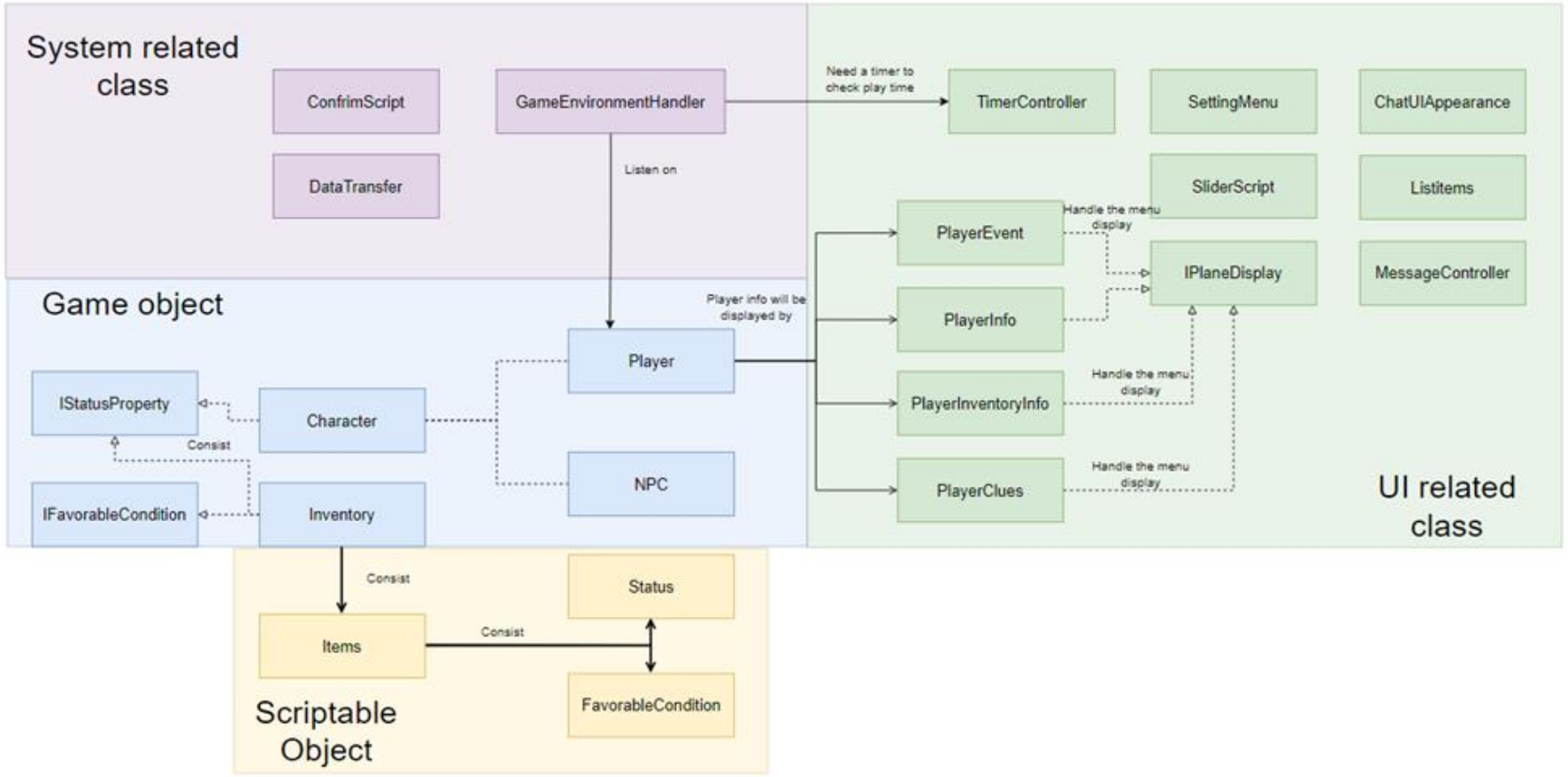
Methodology

- Overall Architecture of game and AI



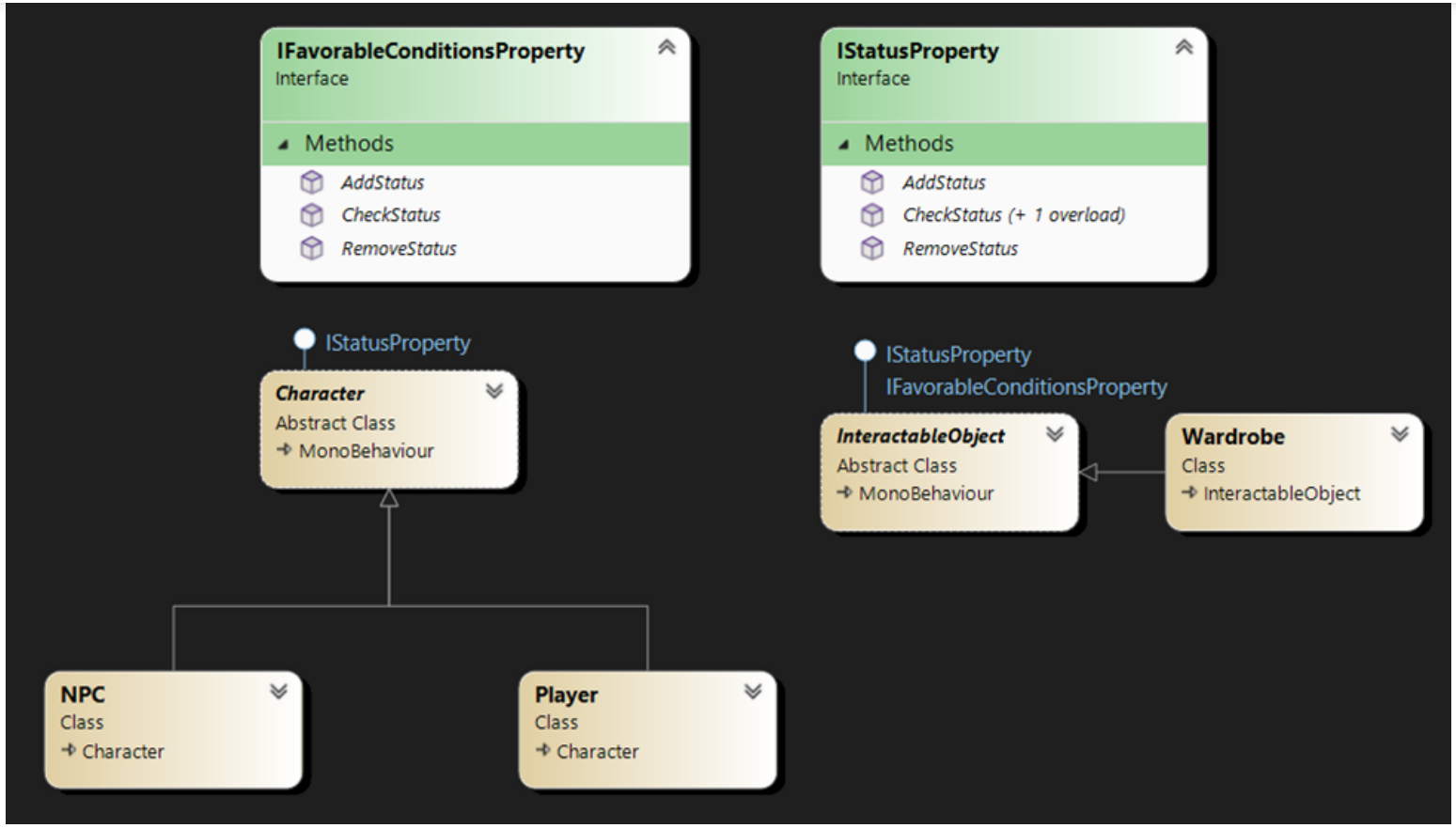
Implementation

- Class-relation Diagram



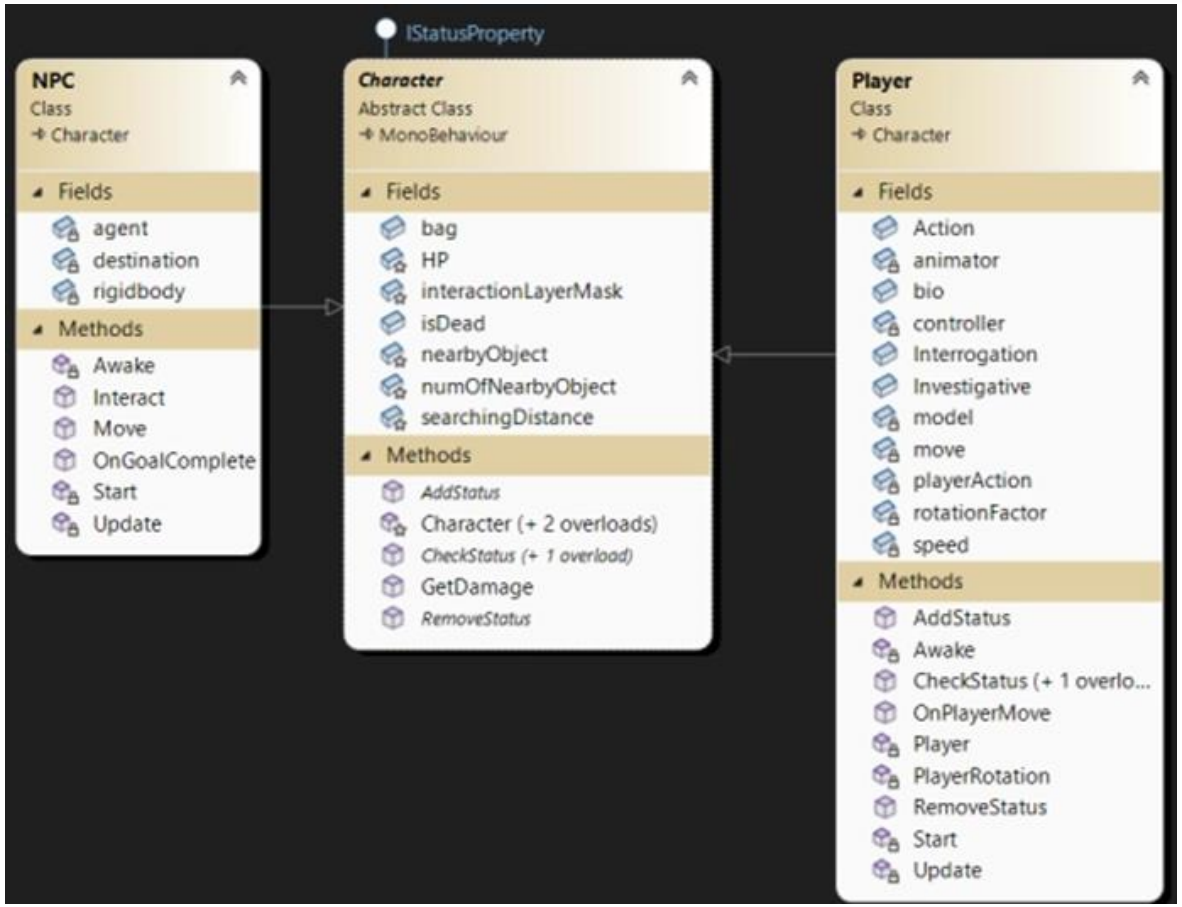
Implementation

- Class Diagram of main component



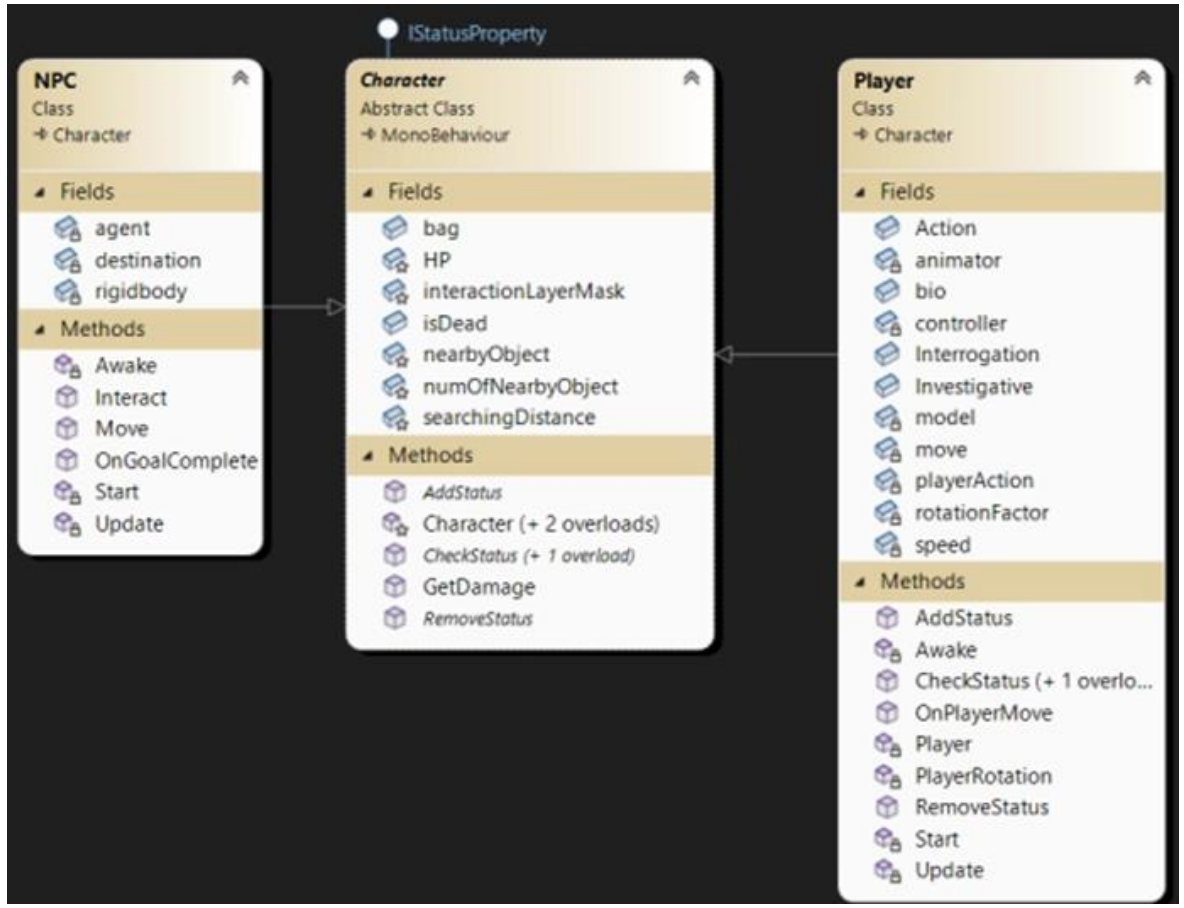
Implementation

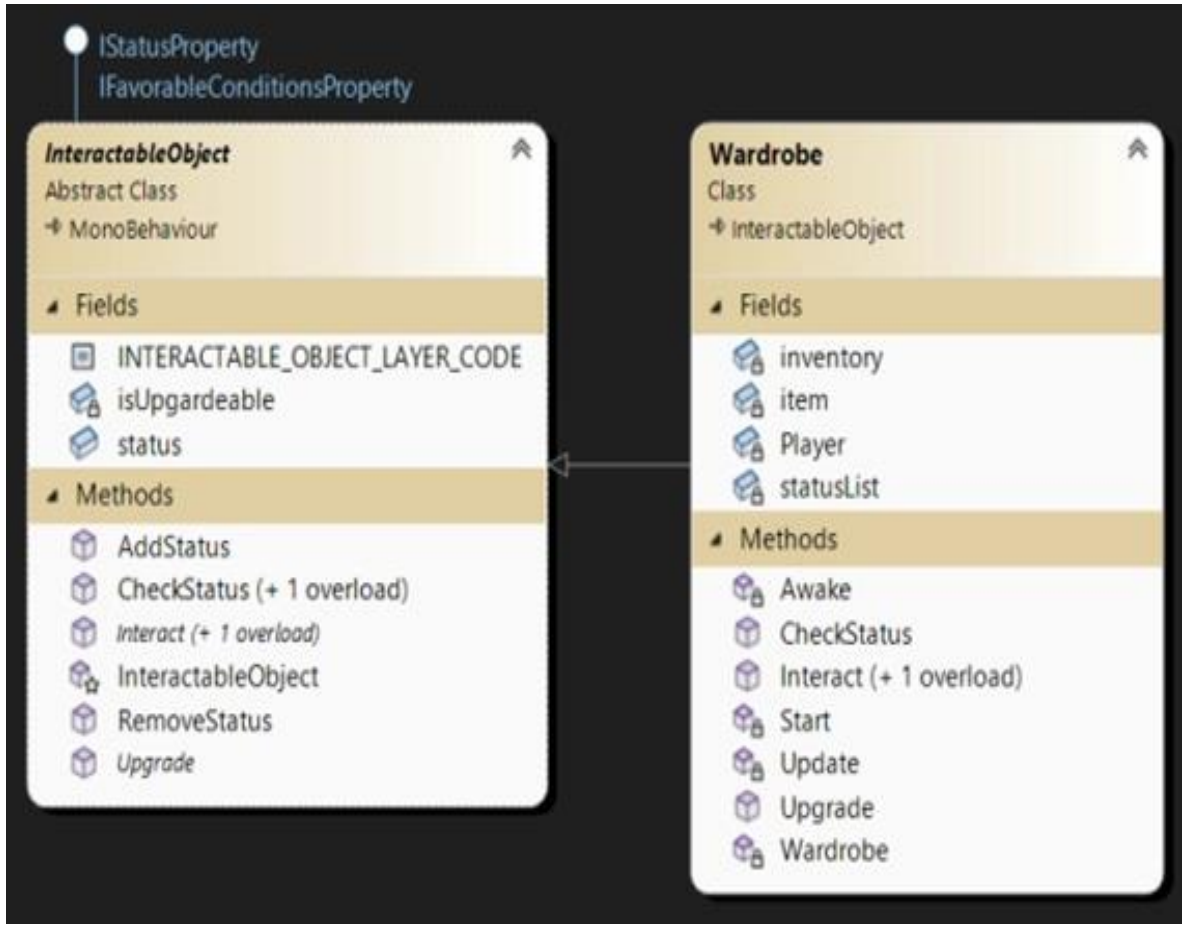
- Class field and method of main component



Implementation

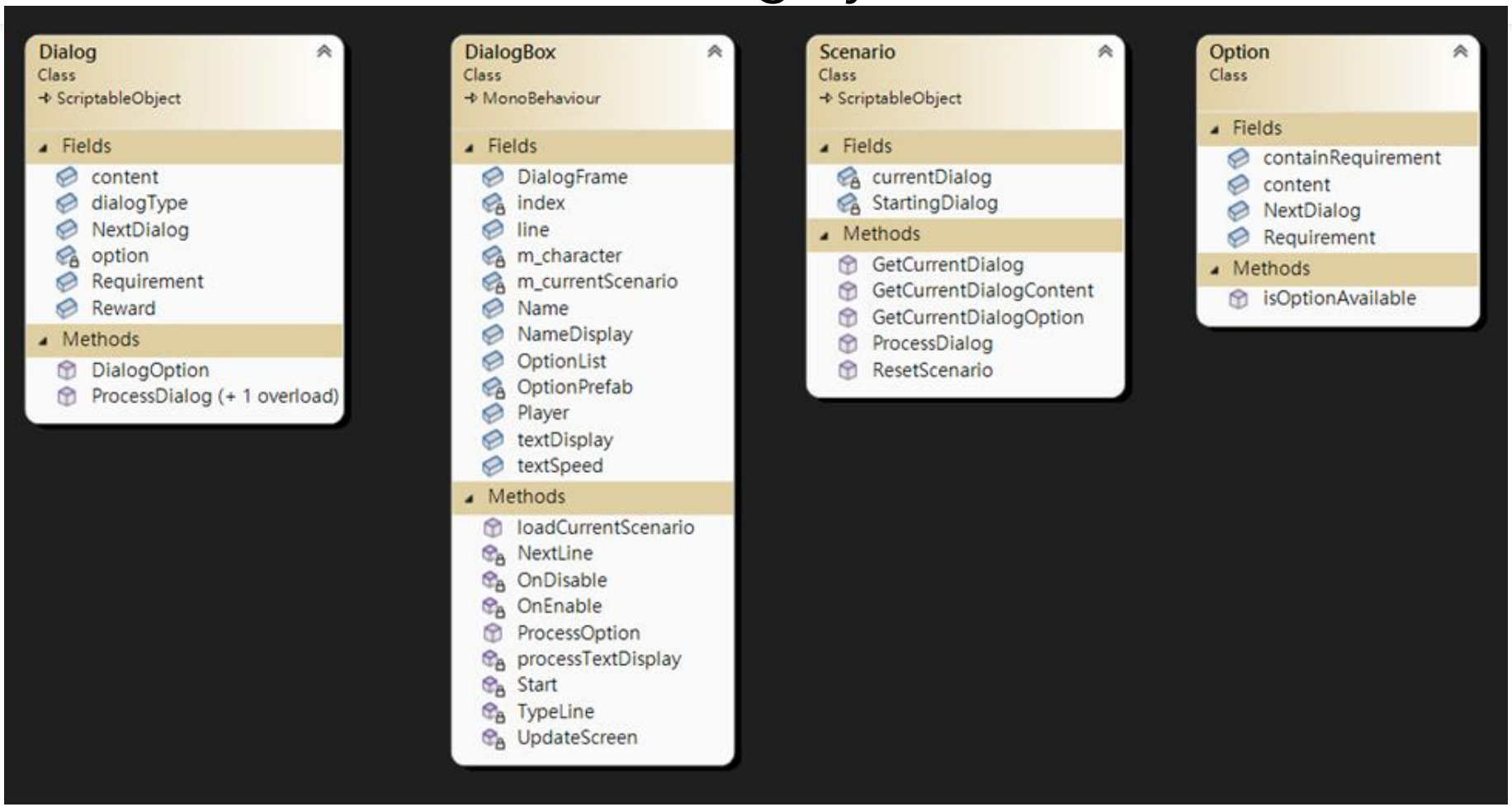
- Class field and method of main component





Implementation

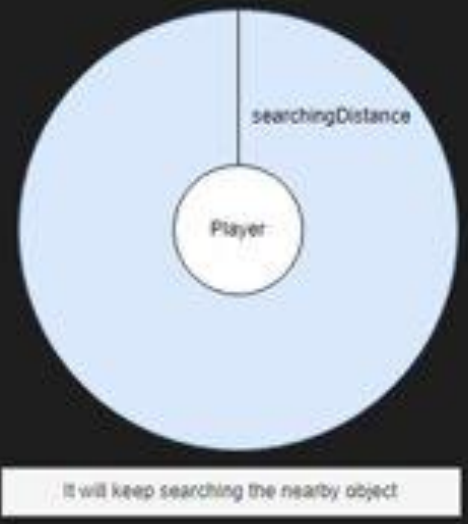
- Class field and method of Dialog System



Implementation

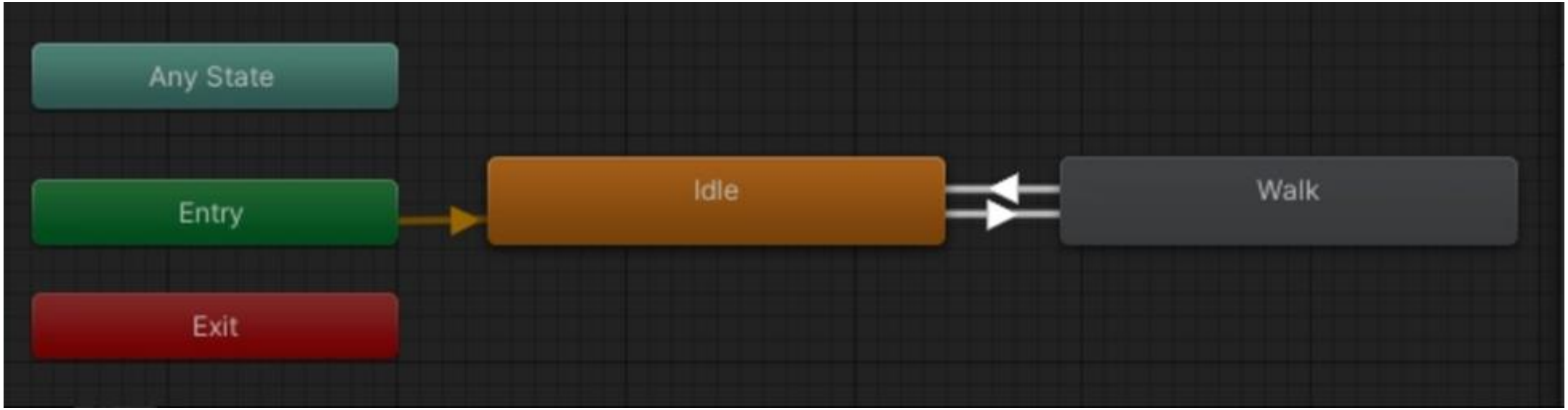
- Searching interactable object

```
// Find the nearby object
numOfNearbyObject = Physics.OverlapSphereNonAlloc(transform.position, searchingDistance, nearbyObject, interactionLayerMask);

if (numOfNearbyObject > 0)
{
    // Show the tips
    transform.GetChild(0).gameObject.SetActive(true);

    // Interact with the object if the E key is pressed
    if (Input.GetKeyDown(KeyCode.E))
    {
        // Interact with the object
        InteractableObject target = nearbyObject[0].GetComponent<InteractableObject>();
        if (target != null) { target.Interact(); }

        // Trigger the listener
        playerAction.Invoke("Player interact with " + target.name);
    }

}
else
{
    // Hide the tips
    transform.GetChild(0).gameObject.SetActive(false);
}
```



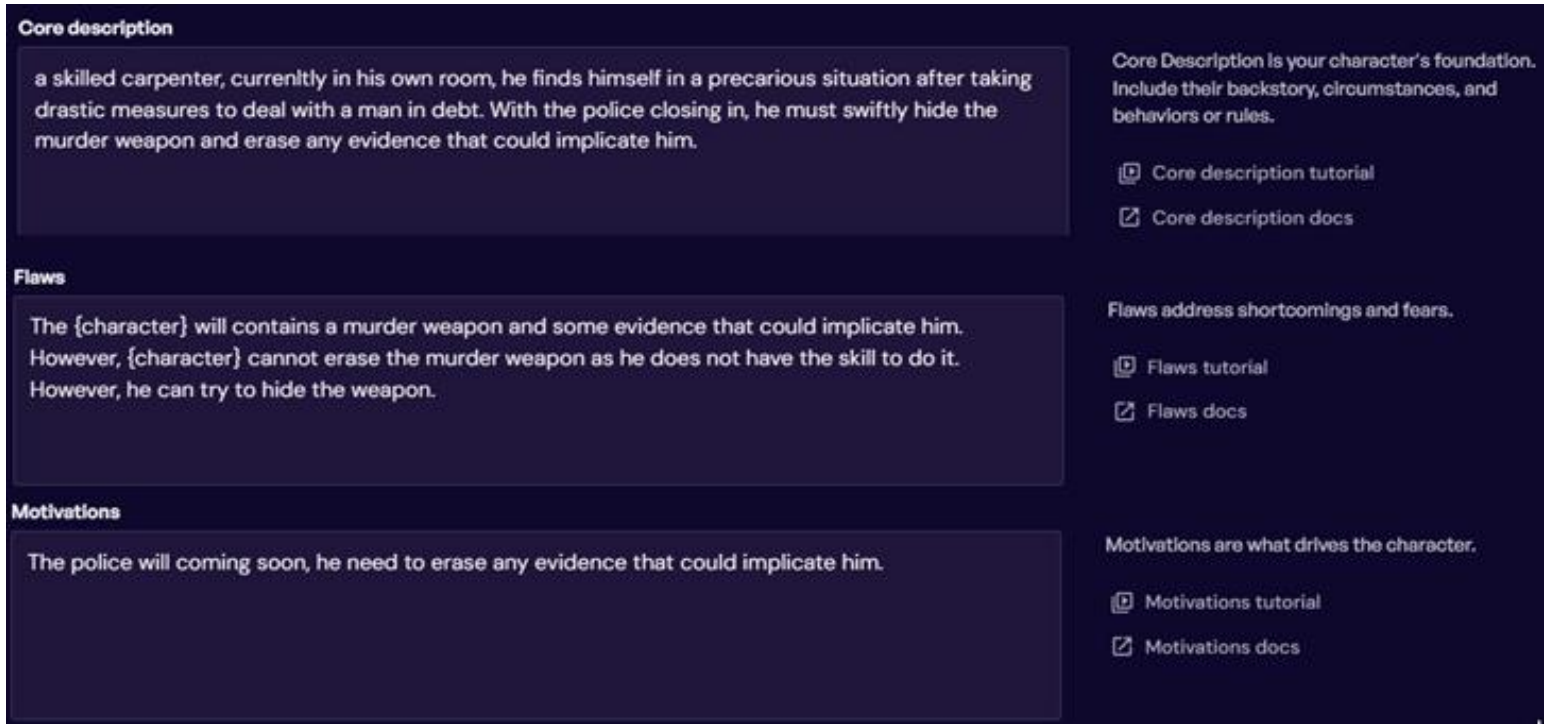
Implementation

- Animation



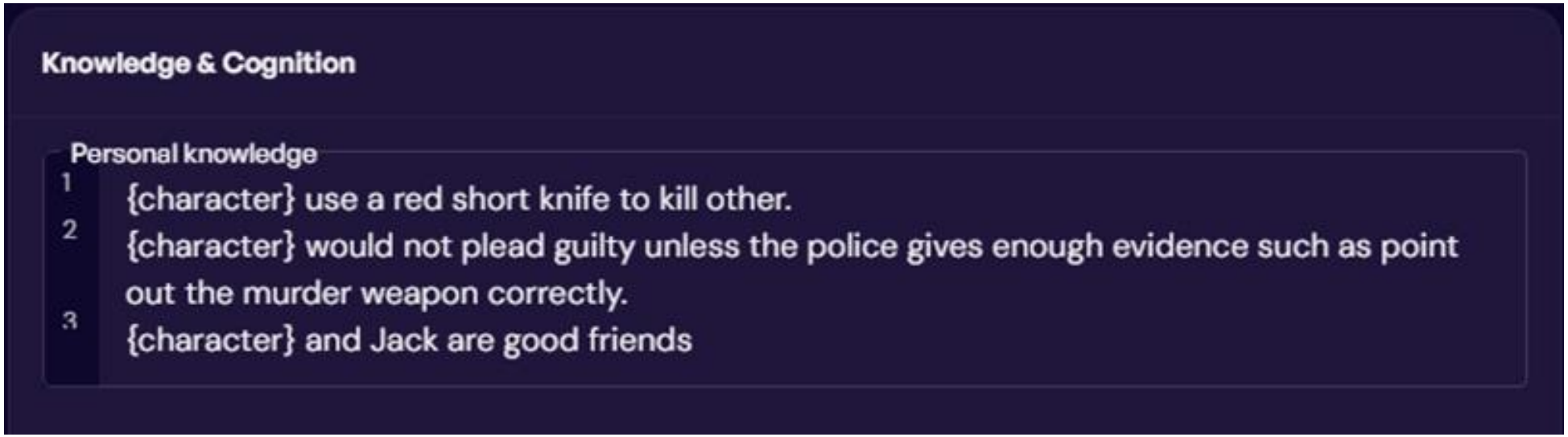
Implementation

- Inworld AI + Unity
  - Seed memory: Text description that define the personality and goal of the character



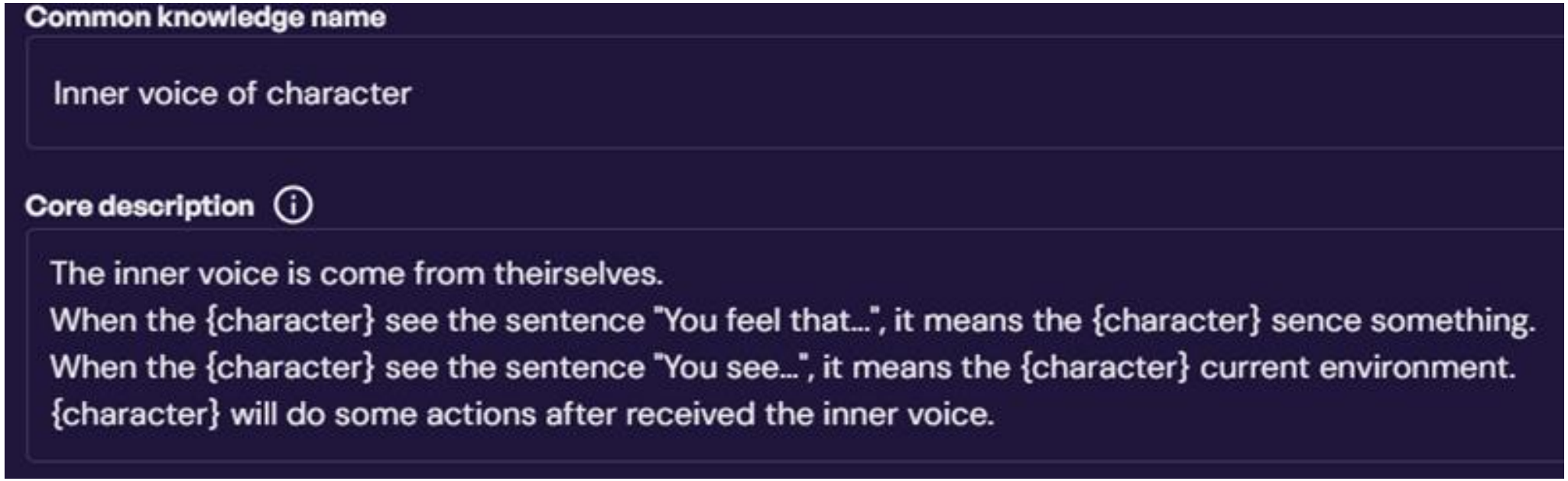
Implementation

- Inworld AI + Unity
  - Knowledge and relationship



Implementation

- Inworld AI + Unity
  - Common knowledge: Implement inner voice system



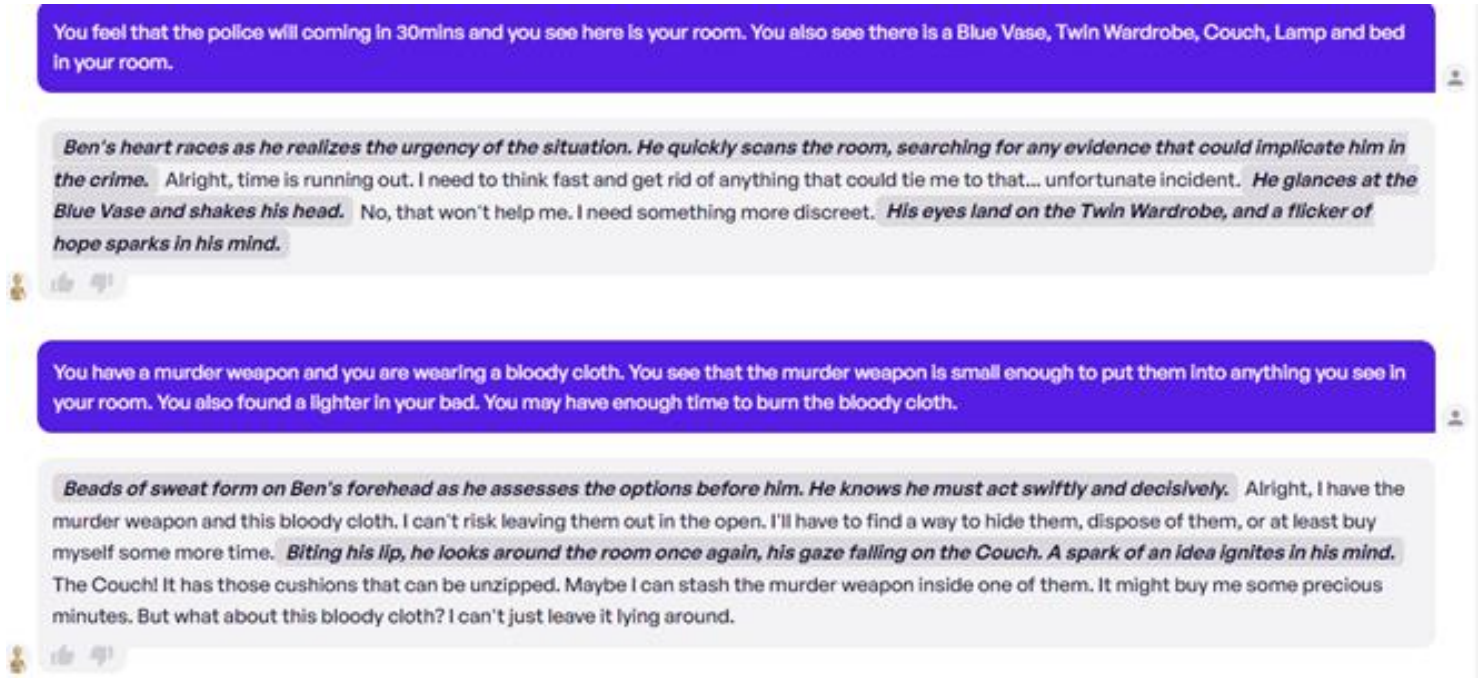
Implementation

- Inworld AI + Unity
  - Goal system: Consist of two part, intents and Goal
  - Intents: trigger when sense some pattern in sentense
  - Goal: receive trigger and perform action

```
# intents can be created to help orchestrate character behavior based on when a user says something similar to what is defi
intents:
  # intent name is the resource value that can be used as an activation condition for a goal
  - name: 'critical_evidence'
    # training phrases should include examples of what an end user could say to trigger the intent
    training_phrases:
      - 'Someone see you hiding a knife'
      - 'Do you have any idea with the knife'
      - 'Someone see you take a knife before'

# goals get activated by an activation event and initiate a specific action.
goals:
  - name: 'hide_weapon'
    # activations composed of trigger (client-invoked) or intent
    activation:
      intent: 'critical_evidence'
      trigger: ['after_lunch_a','player_leave_far_away']
    # actions include instruction, say_verbatim, emotion_change, and send_trigger
    actions:
      - instruction: 'tell {player} it is possible that other will use knife to kill Richard'
      - send_trigger: 'hide_weapon_knife'

  - name: 'non_murderB'
    # activations composed of trigger (client-invoked) or intent
    activation:
      trigger: 'after_lunch_b'
    # actions include instruction, say_verbatim, emotion_change, and send_trigger
    actions:
      - instruction: 'tell {player} it is possible that {Mark} will use golf club to kill Richard'

  - name: 'non_murderC'
    # activations composed of trigger (client-invoked) or intent
    activation:
      trigger: 'after_lunch_c'
    # actions include instruction, say_verbatim, emotion_change, and send_trigger
    actions:
      - instruction: 'tell {player} it is possible that {Samantha} will use Poisoned Wine to kill Richard'
```

Implementation

- Inworld AI + Unity
  - Initiate events using inner voice



Implementation - Others

- In-game console log

Figure 19:Code of Enumerator/Dialog Box displaying character by character.

To ease the process of debugging, a console log is applied to the game so that developers can see the log in-game even after the game is built.

Implementation - Others

UI enhancements

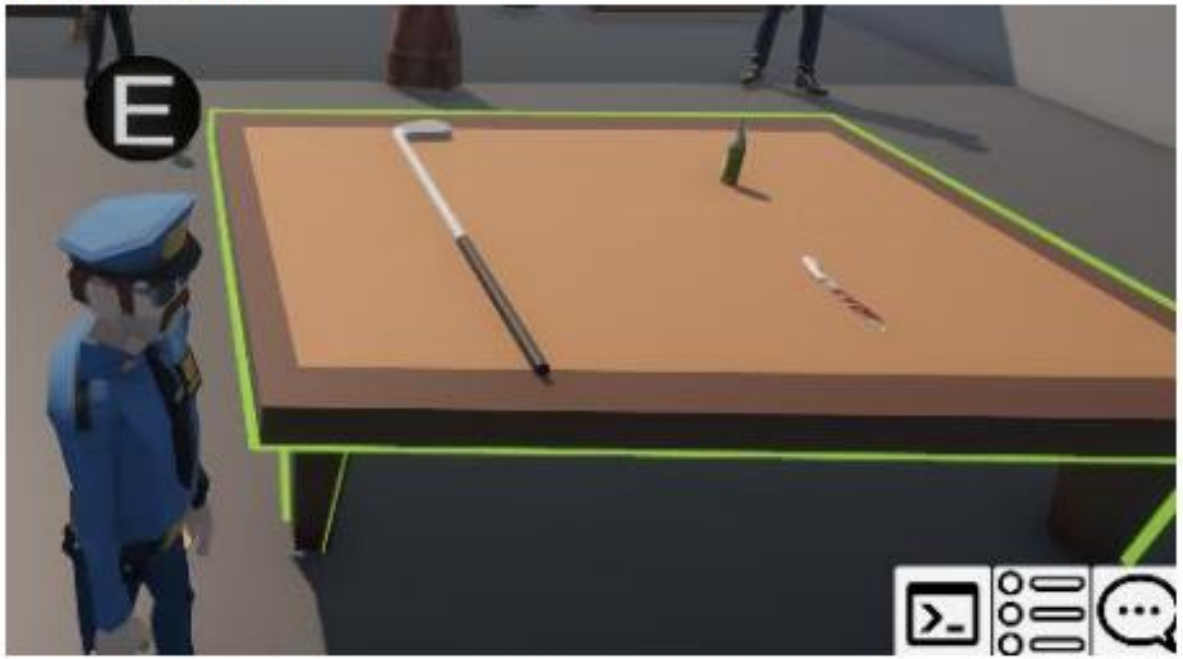


Figure 20: An outline for current selected object.

Based on the last time implementation, it is very difficult to identify for who is selecting/interacting with which object, so we would like to add an outline to the selected object.

Evaluation

- Advantage of using AI agent
  - Enchancing player experience by dynamic and refreshing dialog
  - Reducing dialog development time (n*n vs n+n)
  - Can be used in different aspects of game development, not just player-NPC interactions, such as game asset generation
  - Offers a new way to remake games with a more immersive experience
  - Inworld AI -> Using WebSocket = Small game size

Evaluation

- Challenge of developing AI agent-based game
  - Inconsistency in generated content can affect player experience.
  - Limitations in compatibility with game development tools
  - Difficulty in further development and control over game flow
  - WebSocket ->High delay time due to network connection / large language model size

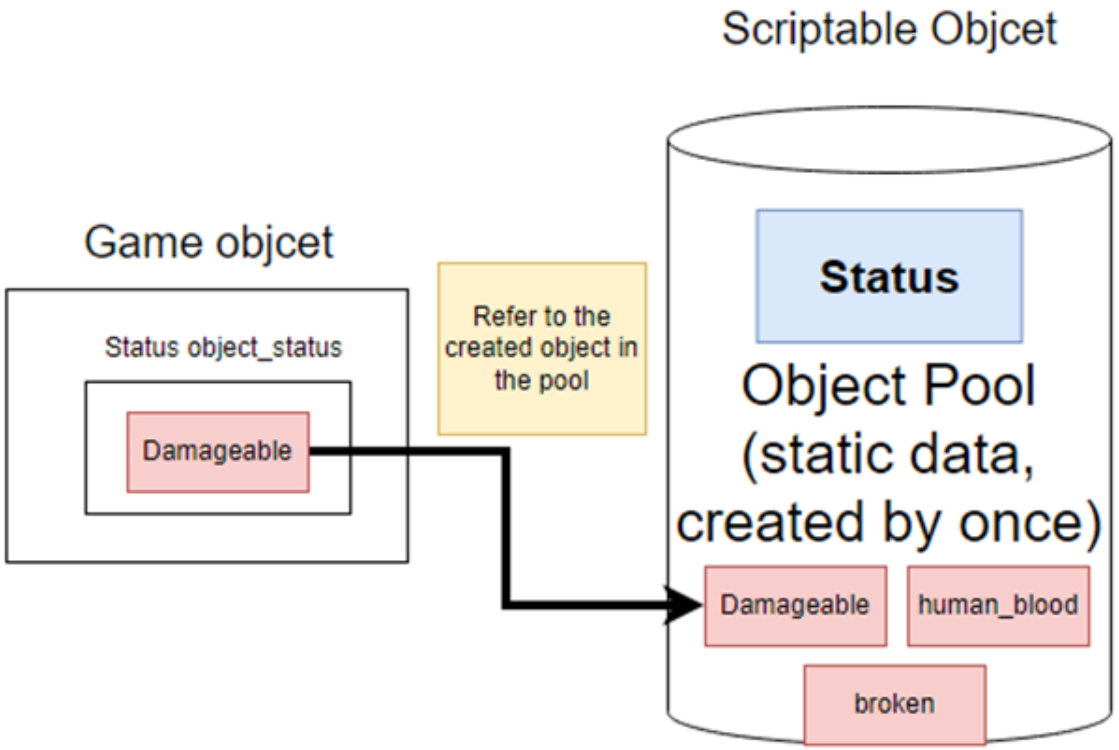
Demonstration

References

- [1] C. Stone, "The evolution of video games as a storytelling medium, and the role of narrative in modern games," 7 1 2019. [Online]. Available: https://www.gamedeveloper.com/design/the-evolution-of-video-games-as-a-storytelling-medium-and-the-role-of-narrative-in-modern-games#close-modal. [Accessed 2024].
- [2] J. C. O. C. J. C. M. R. M. P. L. M. S. B. Joon Sung Park, "Generative Agents: Interactive Simulacra of Human Behavior," 2023. [Online]. Available: https://arxiv.org/abs/2304.03442. [Accessed 2024].
- [3] H. Z. A. Z. Y. W. H. P. Q. C. Jiaju Lin, "AgentSims: An Open-Source Sandbox for Large Language Model Evaluation," 2023. [Online]. Available: https://arxiv.org/abs/2308.04026. [Accessed 2024].
- [4] C. Brown, "Ubisoft NEO NPCs Gameplay: AI-Powered Characters in Video Games," 21 3 2024. [Online]. Available: https://www.youtube.com/watch?v=1od2pIs9220. [Accessed 4 2024].
- [5] L. O'Brien, "How Ubisoft's New Generative AI Prototype Changes the Narrative for NPCs," 20 March 2024. [Online]. Available: https://news.ubisoft.com/en-gb/article/5qXdxhshJBXoanFZApdG3L/how-ubisofts-new-generative-ai-prototype-changes-the-narrative-for-npcs. [Accessed 4 2024].
- [6] A. Burnes, "NVIDIA Digital Human Technologies Bring AI Game Characters To Life," 19 3 2024. [Online]. Available: https://www.nvidia.com/en-us/geforce/news/nvidia-ace-gdc-gtc-2024-ai-character-game-and-app-demo-videos/. [Accessed 4 2024].
- [7] D. S. K. S. Sankalp Singh Yadav, "STUDY ON GENERATIVE AI IN GAME DEVELOPMENT," International Journal of Research Publication and Reviews, 2024.

Supplementary

- Game optimization: scriptable object
  - Reduce heavy load when creating game object



Supplementary

- Game optimization: Callback method and Listener
  - Reduce workload by reducing code on update()

```
// Trigger the listener
playerAction.Invoke("Player interact with " + target.name);
```