

Claude Plays Pokemon Review + Insights

Inspired by the ClaudePlaysPokemon Twitch Stream

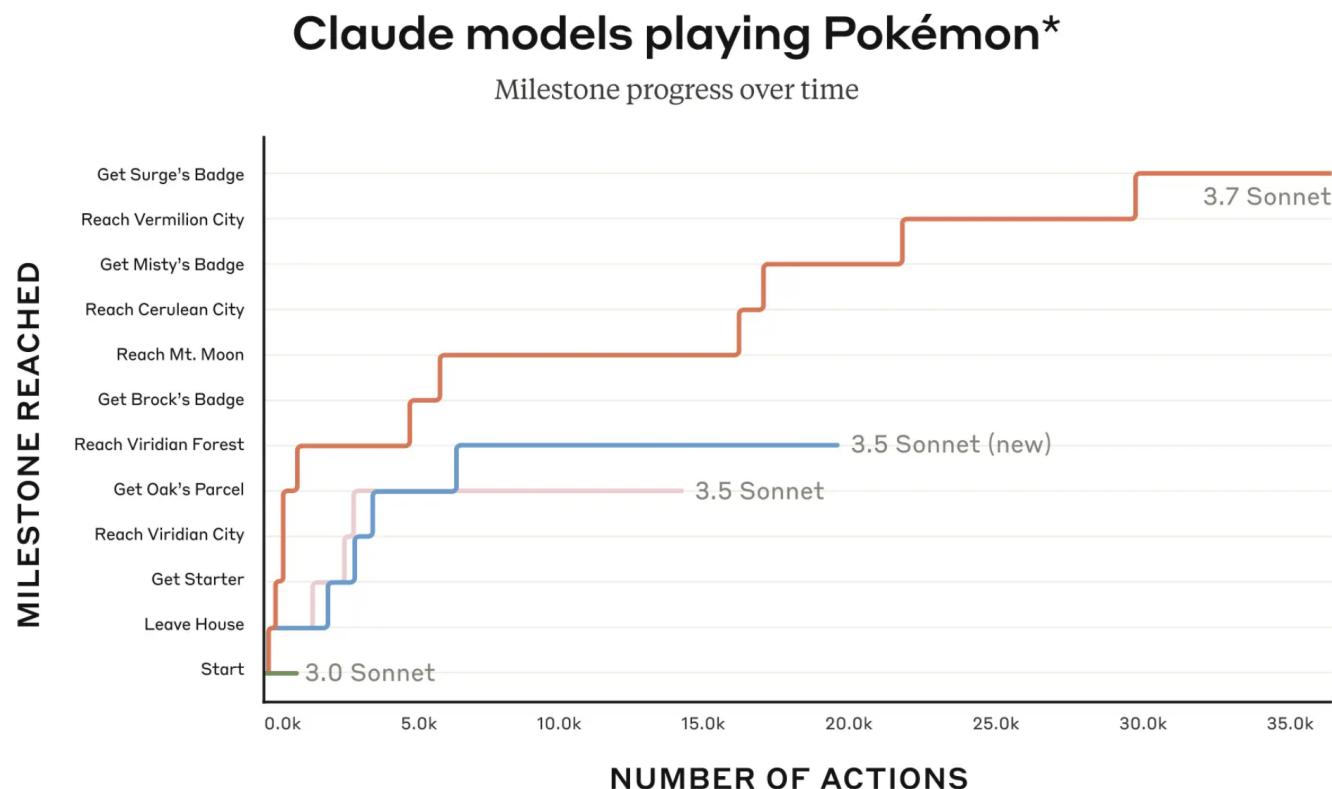
<https://www.twitch.tv/claudieplayspokemon>

10 Mar 2025

Presented by:

John Tan Chong Min

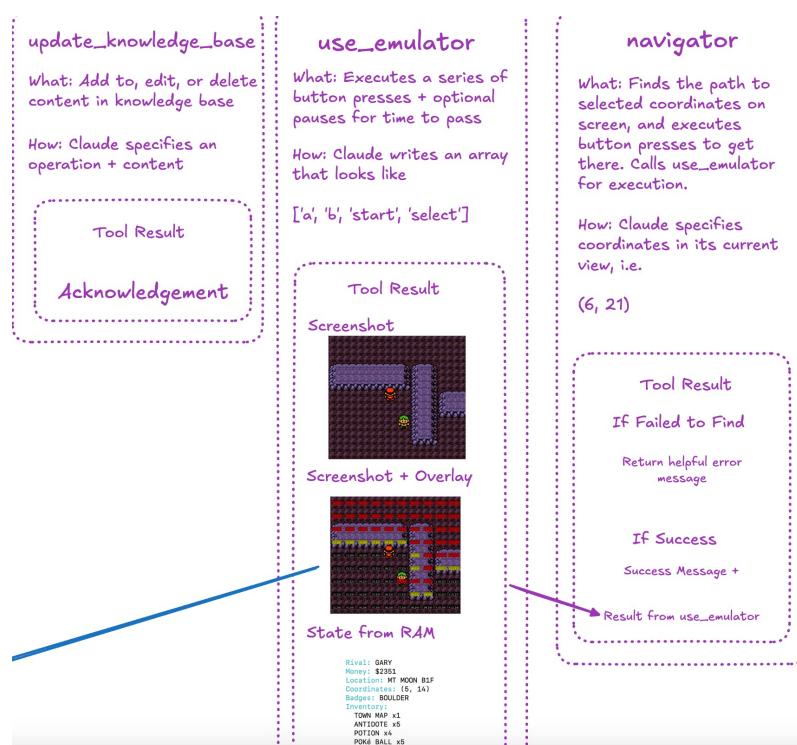
Thinking Model = Better Pokemon Playing?



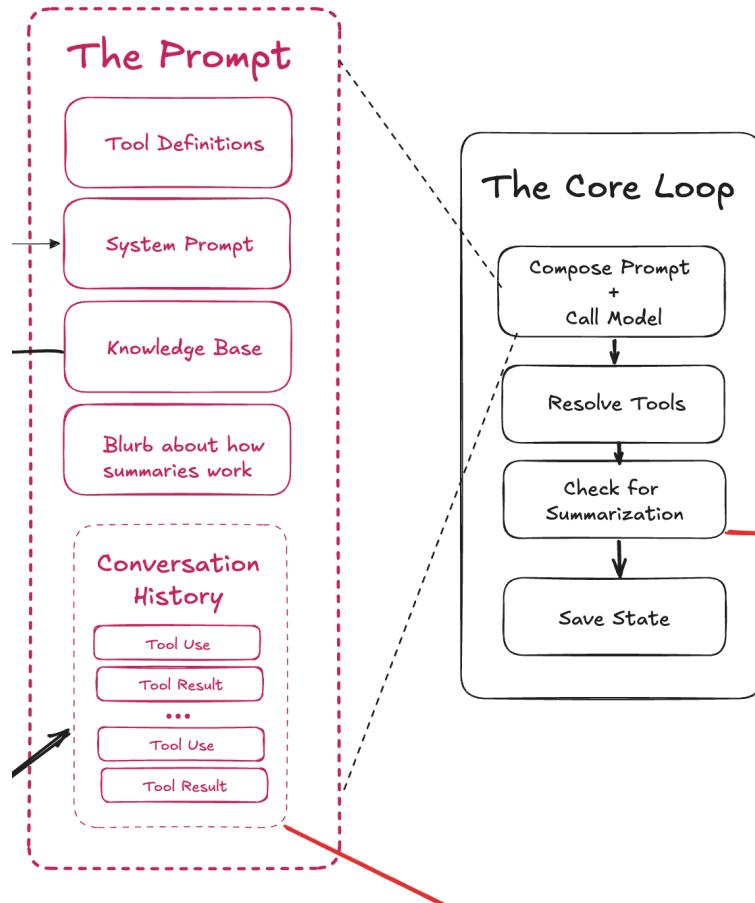
<https://www.anthropic.com/news/visible-extended-thinking>

Action Space - Tools

- When Claude Plays Pokemon (CPP) wants to act, it can:
 - Press game buttons in sequence (A, B, Up, Down, Left, Right, Start, Select)
 - Navigate to specific coordinates on the screen
 - Update its knowledge base with new information it discovers



Main Game Loop



Summarization / Managing Long Context

Rollouts are way too long for the 200k context window, so we need some way to manage context. I've found progressive summarization to work well

Once conversation history > max_turns we trigger a summarization event

This has Claude write a summary of their recent progress / what happened in the last max_turns.

Then we clear the full conversation history and insert the summary as the first assistant message, and Claude resumes its journey

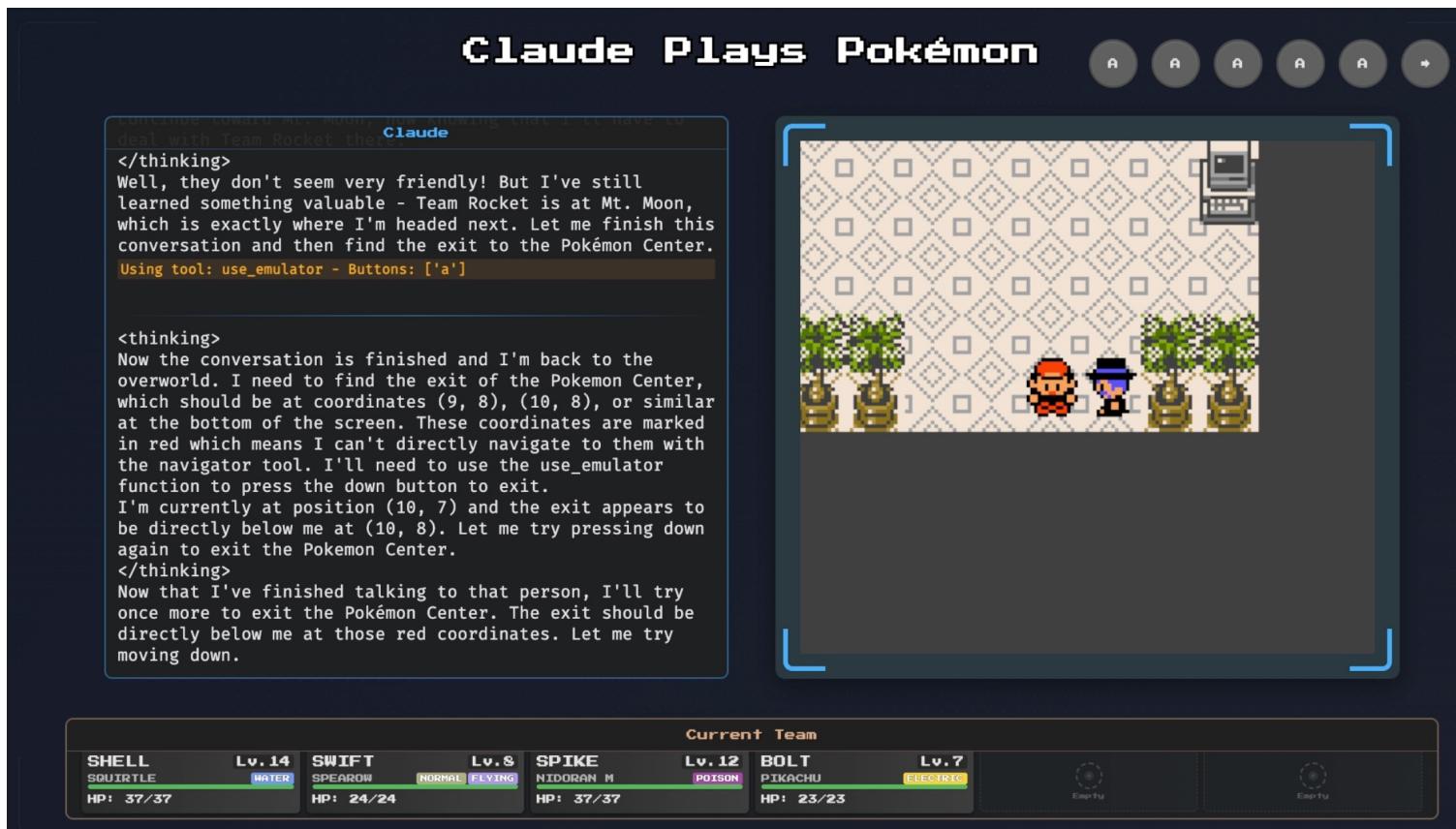
Finally, another LLM is called to inspect the first LLM's knowledge base and to provide feedback -- this helps ensure the agent does more frequent maintenance of its knowledge base

Combining it all together

The system combines several components for CPP:

- **Game Interface:** A custom interface that allows Claude to control the game by pressing virtual buttons.
- **Screenshot Analysis:** Claude looks at screenshots of the game and interprets what's happening on screen. Claude's ability to understand Game Boy screens isn't great—you'll often see it misinterpret what it's seeing.
- **Knowledge Base:** Claude maintains a dynamic set of notes about the game world, storing information about locations, Pokémon team status, and game mechanics.
- **Navigation System:** A pathfinding tool helps Claude navigate efficiently by finding paths to a spot on the screen.
- **Memory Reader:** Claude also gets a few tidbits of information from the memory of the game, like its current location and party. This helps prevent Claude from getting confused if it misinterprets the screen.

Overall Process:
Image + RAM State -> Think + Tool Use



World Map Overlay



(2,16)	(3,16)	(4,16)	(5,16)	(6,16)	(7,16)	(8,16)	(9,16)	(10,16)	(11,16)
(2,17)	(3,17)	(4,17)	(5,17)	(6,17)	(7,17)	(8,17)	(9,17)	(10,17)	(11,17)
(2,18)	(3,18)	(4,18)	(5,18)	(6,18)	(7,18)	(8,18)	(9,18)	(10,18)	(11,18)
(2,19)	(3,19)	(4,19)	(5,19)	(6,19)	(7,19)	(8,19)	(9,19)	(10,19)	(11,19)
(2,20)	(3,20)	(4,20)	(5,20)	(6,20)	(7,20)	(8,20)	(9,20)	(10,20)	(11,20)
(2,21)	(3,21)	(4,21)	(5,21)	(6,21)	(7,21)	(8,21)	(9,21)	(10,21)	(11,21)
(2,22)	(3,22)	(4,22)	(5,22)	(6,22)	(7,22)	(8,22)	(9,22)	(10,22)	(11,22)
(2,23)	(3,23)	(4,23)	(5,23)	(6,23)	(7,23)	(8,23)	(9,23)	(10,23)	(11,23)
(2,24)	(3,24)	(4,24)	(5,24)	(6,24)	(7,24)	(8,24)	(9,24)	(10,24)	(11,24)

Updating of Knowledge Base

Claude

```
<thinking>
Now let me update the inventory section with the accurate
information:
</thinking>
Using tool: update_knowledge_base - edit section: inventory

<thinking>
Now let me update the battle_log section to include the
recent LASS battle:
</thinking>
Using tool: update_knowledge_base - edit section: battle_log

<thinking>
Next, let me update the current_objectives section to
better separate immediate from longer-term goals:
</thinking>
Using tool: update_knowledge_base - edit section: current_objectives

<thinking>
```

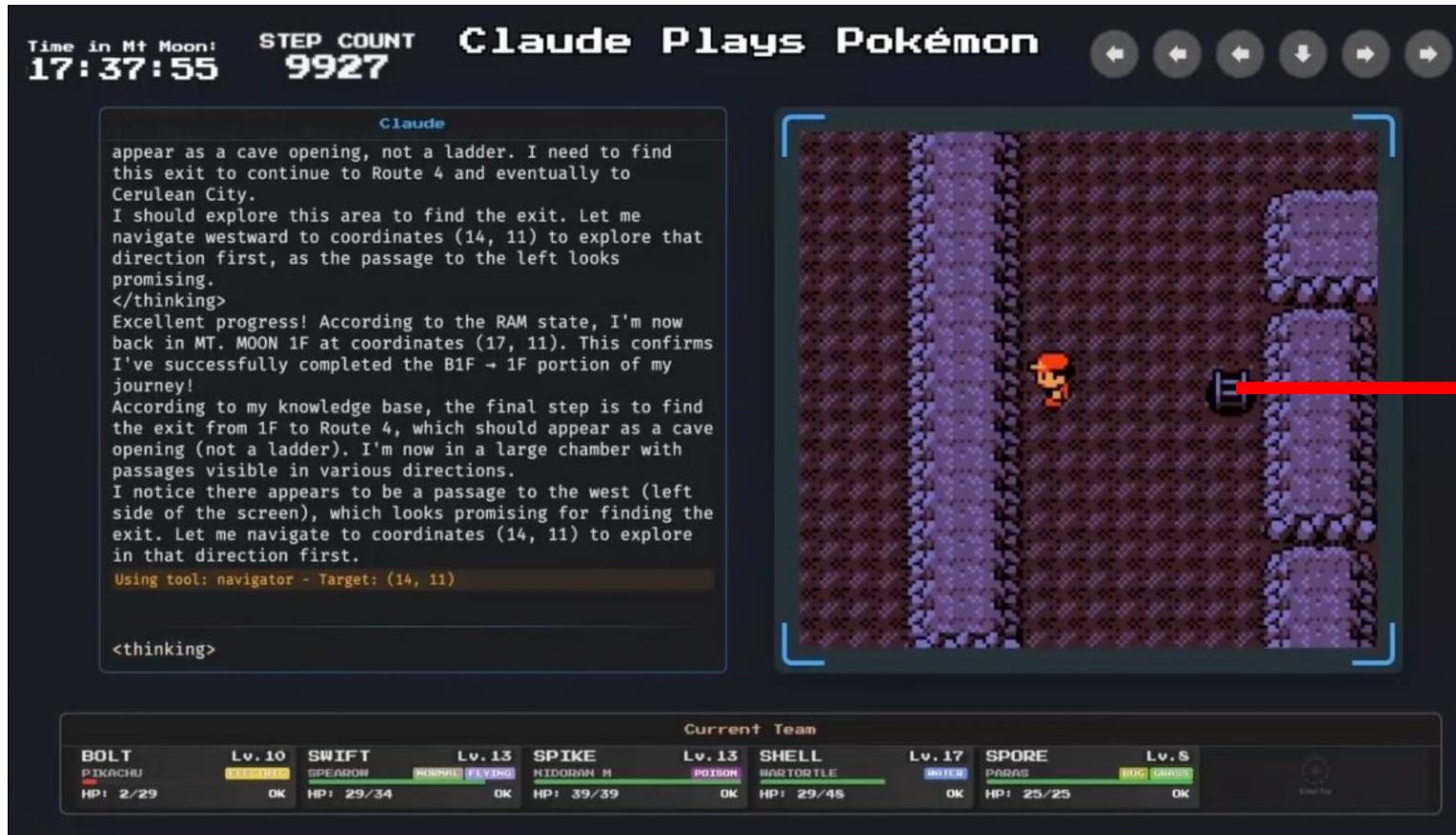


SHELL
L15
HP: 25 / 39

Knowledge Base Update

- 1. Continue east through Route 3 to reach the Pokémon Center near Mt. Moon entrance
 - URGENT: Multiple Pokémons need healing, especially BOLT (fainted)
 - Avoid unnecessary wild Pokémon encounters and trainers until team is healed

Ladder Trouble (leading to 72+ hours at Mt. Moon)

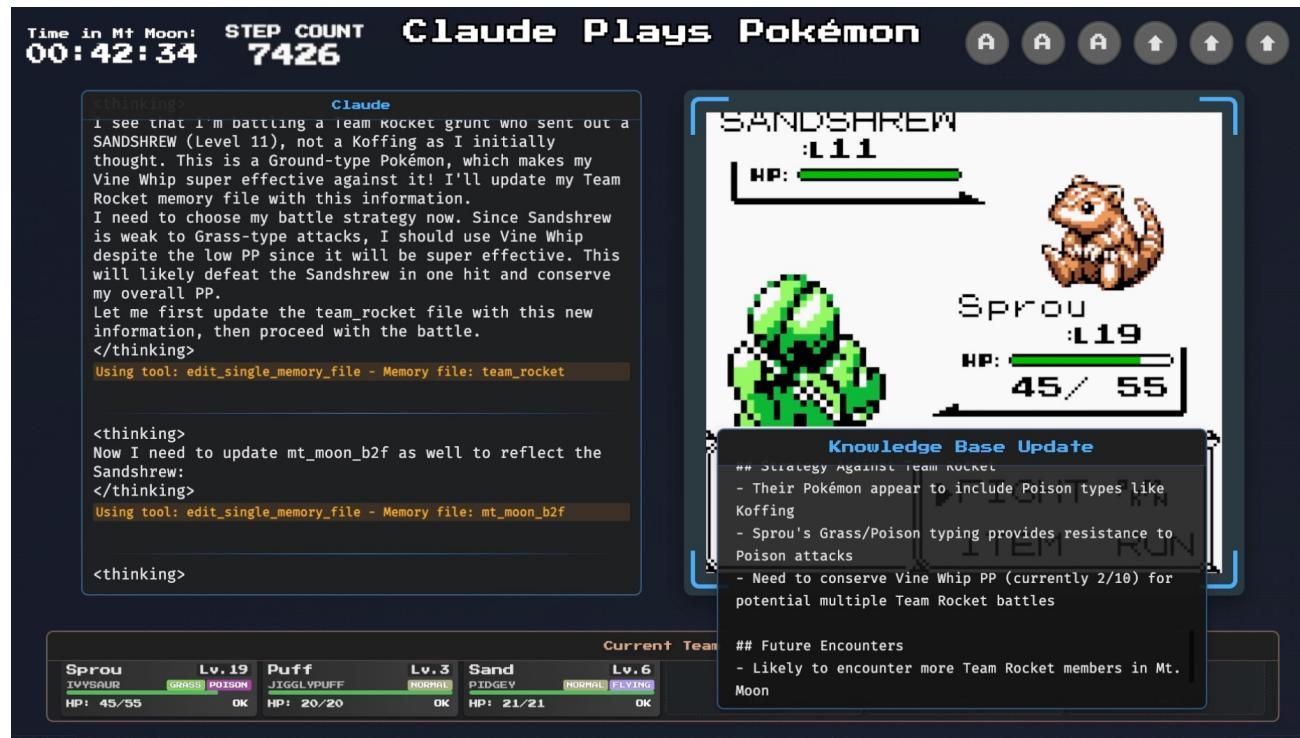


Keeps exploring this ladder, and after that, head north instead of going south

Agent with Memory Issues!

- CPP **keeps forgetting** whether it has seen something before or not
- In fact, CPP is very enthusiastic about “new” things, which were actually explored before but not stored in its available memory context
- CPP often overrides its memory when “cleaning context” that makes it forget important information, getting it stuck in cycles

Run 2: Updated memory with various files



Memory / Navigation Issues

- Even with separate files for each area, important information may not be stored
- Even with separate files for each area, important information is still lost when “cleaning context”
- With separate files for each area, the relevant file may not be retrieved, resulting in a loss of context
- **Navigation:** A long path will mean difficulty in navigating as it is not a direct pattern association from past context – some search is required

My Insights

Memory as streams

- Instead of a single document for memory, why not store it as chunks of memory that can be modified / replaced
- Can use presence of an existing similar memory to judge novelty

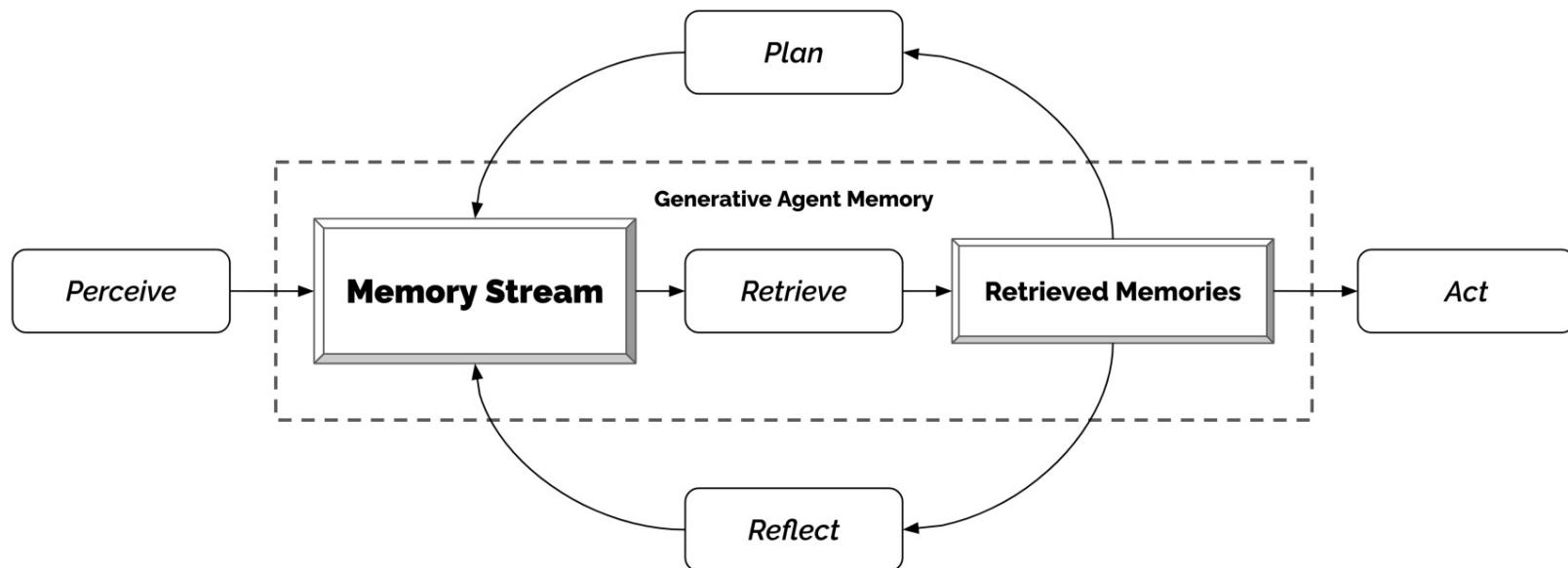


Image Extracted from **Generative Agents: Interactive Simulacra of Human Behavior**, Park et. al., 2023

How do humans prioritise? Emotions

- 4 deck game (Iowa Gambling Task) – Even without knowing the true distribution, people feel scared drawing from the deck that has large negative rewards
- In “**The Extended Mind**” by Annie Murphy Paul (2021), she mentions that bodily states can influence decisions, more than just the mind itself.



Iowa Gambling Task

Retrieving memory based on multiple factors, including emotion

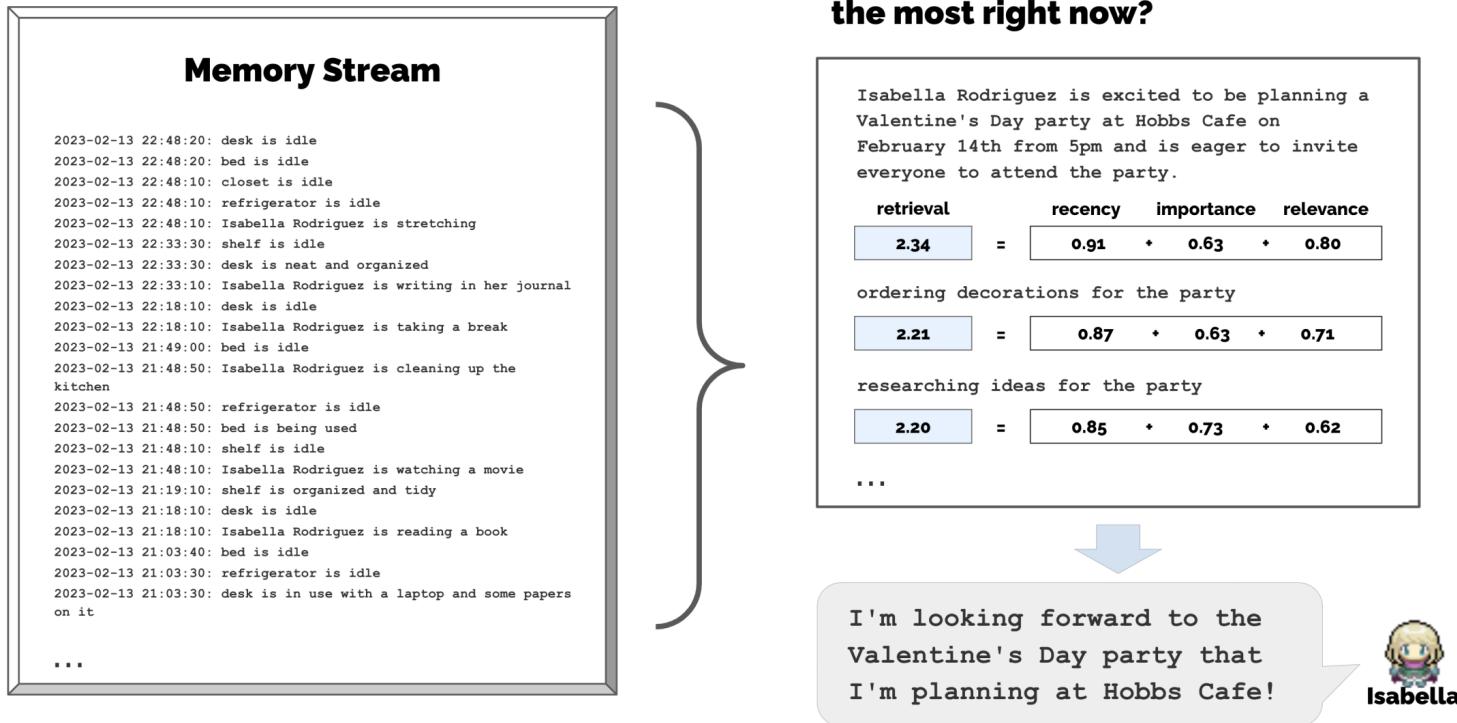


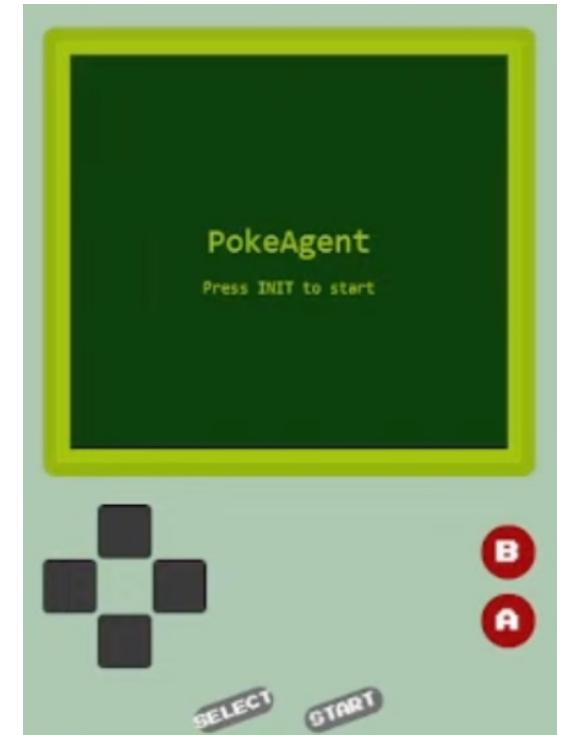
Image Extracted from Generative Agents: Interactive Simulacra of Human Behavior, Park et. al., 2023

Practical Implementation: Stay Tuned for “AgentJo Plays Pokemon!”

Multimodal Memory, Pathfinding Algorithms, Hierarchical Planning

Dylan Chia, Dylan Tan, Nicholas Tan, Nicholas Chen, John Tan Chong Min

<https://github.com/tanchongmin/agentjo/tree/main/projects/Pokemon>



Using Images as Memory



- Can we use **image inputs directly** to understand the state, without needing a conversion to text?
- May be easier to extract information on a need-to-know basis than compressing them all into text

```
res = strict_json("""You are a friendly assistant to play a game for the player.  
You are to describe what happened in previous game states'',  
    "Previous state 1: <>map1.png>, state 2: <>map2.png>",  
    output_format = {"State 1": "Describe first state",  
                    "State 2": "Describe second state",  
                    "Thoughts": "Describe what has happened"},  
    llm = llm)
```

```
res
```

```
{'State 1': 'The player character is in a cave with two other characters. One is in the bottom left corner and the other is in the top center.',  
 'State 2': 'The player character is in a cave with one other character in the top center. The character in the bottom left corner has disappeared.',  
 'Thoughts': 'The player character has moved slightly to the right. The character in the bottom left corner has disappeared, likely due to the player character moving further away from them.'}
```

Similar idea to frame concatenation in the **Deep Q Network paper (Playing Atari with Deep Reinforcement Learning, Mnih et. al., 2023)

Multimodal Input Processing

```
from agentjo import strict_json

res = strict_json("""You are a friendly assistant to play a game for the player.
The player is at the centre of the screen.
Coordinates are given as (x, y), x being the left to right axis, y being the up to down axis
Left moves in the negative x direction, e.g. (x, y) to (x-1, y)
Right moves in the positive x direction, e.g. (x, y) to (x+1, y),
Up moves in the negative y direction, e.g. (x, y) to (x, y-1)
Down moves in the positive y direction, e.g. (x, y) to (x, y+1)
A is typically used for confirming actions, selecting options, and interacting with elements in the game
B is typically used to cancel actions, go back to the previous screen, or exit menus
Start is used to open the main menu, allowing players to access options like the Pokémon menu, bag, items, and settings
Select is used to register key items, allowing players to quickly access them during gameplay""",
    "Based on the image <><map.png>>, what should the player do to talk to the NPC?",

    output_format = {"Thoughts": "'State where the player is, where the destination is,
and the actions needed to navigate there, type: str''',
                    "Action List": "type: list[Enum['Up', 'Down', 'Left', 'Right', 'A', 'B', 'Start', 'Select']]"},

    llm = llm)

res
```

```
{"Thoughts": "The player is at (6, 20) and the NPC is at (7, 22). The player needs to move down twice and then right once to be next to the NPC and then press A to talk to them.",
 "Action List": ["Down", "Down", "Right", "A"]}
```



Using algorithms to navigate

- Instead of relying on LLM to “plan” and navigate, why not create a map and then use known pathfinding algorithms (e.g. A* search) to navigate?
- Instead of a map, state nodes and action edges can also be stored, and some search used to do pathfinding based on these states (e.g. Dijkstra, Floyd Warshall)
- Hierarchical (broad to specific) navigation could also be used – first plan a sequence of overall places to visit using a broad planner, then use a specific planner to plan details of that place itself

Questions to Ponder

- How much of an Agent's performance is due to the LLM itself, and how much of it is due to external structures like tools, memory?
- Should emotions be used for memory retrieval and storage?
- Can we interpret images without the need to convert to text?
- What other forms of memory should we have?

AgentJo Logo Design Competition

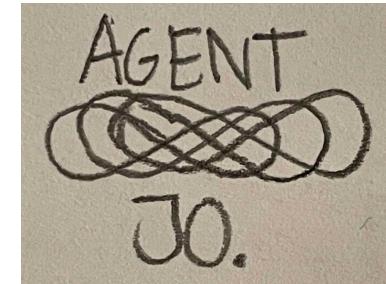
- AgentJo is my project for the next 5 – 10 years to create fast learning and adaptable agents!
- We have a logo design competition, now down to the final 3!
- Go to the discord group (<http://discord.com/invite/bzp87AHJy5>) to vote!
- Competition Ends: 11 Mar 2025, 2359hrs (GMT+8)



Boddu Sri Pavan



Randy



neoneye

<https://github.com/tanchongmin/agentjo>