Project Report : CS 5660
AI - TOWN

Keerthan Reddy Animireddigari
California State University, LA
Department of CS
kanimir@clastatela.edu

Sai Venkat Raghava Revanuri
California State University, LA
Department of CS
srevanu2@calstatela.edu

Harish Reddy Mekala
California State University, LA
Department of CS
hmekala@calstatela.edu

Abstract

The AI Town project envisions a virtual community where AI characters engage in conversations and social activities, inspired by the research paper "Generative Agents: Interactive Simulacra of Human Behavior." Our initiative offers a deployable starter kit for developers to customize AI Town, drawing from the intriguing question posed by Stanford and Google researchers: What happens when 25 AI agents are thrown into a simulated village? The outcomes mirror human behavior, utilizing technology similar to Chat GPT.

Going beyond a development experience, our project aims to provide a robust foundation for expansion. Utilizing 11 virtual agents, we explore AI-driven social dynamics in a simulated village, unraveling mysteries of artificial intelligence and fostering understanding in this evolving field.

1. Introduction

We aimed to create computer characters that behave like real people in digital worlds. Our goal was to make these characters remember their past actions and interactions, think and reflect on them, and then use this understanding to decide what to do next. This way, they can act and react in ways that feel real and believable, just like how people do in the real world. We wanted these characters to be able to handle different situations, interact with each other, and change their actions based on new experiences, just like humans do.

Today, in video games and virtual worlds, the characters you see are usually programmed with a set of rules or scripts that tell them how to act. They can perform certain actions or say specific things, but they don't really 'think' or 'remember' like humans. These characters often repeat the same actions and can't really learn from what's happening around them. They're not very good at acting differently based on new experiences or past interactions. So, their behavior can feel repetitive and predictable, not very much like real people.

If we manage to make these computer characters behave more like real people, it's going to change a lot of things. Imagine playing a video game where the charac- ters you meet can surprise you with their actions, making every gaming session unique. It's not just about games, though. Think about training programs for professionals like doctors or firefighters; they could practice in scenar- ios that are much closer to real life, helping them prepare better. This technology isn't only useful for entertainment and training. Researchers studying human behavior could gain insights without needing to involve real people, whichis safer and more convenient. And consider the last time you interacted with a customer service chatbot; if these bots could understand and respond more like a human, getting help with your problems online could be a lot smoother and more pleasant. So, in essence, making these advancements could make many aspects of our digital interactions more engaging, realistic, and helpful.

In the "AI Town" project, the main kind of data used was from the ChatGPT language model. This model was trained on a wide range of texts from books, websites, and other sources to understand and generate human-like lan- guage. This training allowed the generative agents to createrealistic and varied responses in their interactions. The keypart of the data was how it taught the agents to mimic hu- man conversations and behaviors, making them seem morelike real people in the digital world.

2. Approach

We combined the abilities of a powerful language model, like ChatGPT, with a new system to remember, think, and plan. This meant our characters could not only talk like humans but also remember past interactions, think about them, and use that information to decide what to do next. We made three key parts in our system: A memory part that lets the characters remember what they've done or seen. A thinking part that lets them make sense of their memories. A planning part that helps them decide on their future actions based on their memories and thoughts. We believed this approach would work because it mimics how people behave. People remember their experiences, think about them, and then make decisions. By giving these abilities to computer characters, we made them act more realistically. The new thing in our approach was how we combined memory, thinking, and planning with the language model. This hadn't been done before in this way. It allowed the characters to have more human-like, varied, and evolving behaviors in the digital world.

We tried to replicate the entire AI Town Project and tried to run the project and since we had all the code, environment and requirements in our system, so we anticipated that we could run the project successfully initially and We anticipated difficulties in establishing Clerk and Convex, predicting that we would face obstacles during this setup phase. but we can make the changes accordingly based on our understanding. We were not able to re-produce the project successfully because we were not able to make the interact button work in the system and we were struck there. Also during the setup of Clerk and Convex, we encountered difficulties as anticipated. Nevertheless, by thoroughly consulting their respective documentation, we were able to successfully overcome these challenges. We did try lot of possibilities and code changes but due to time crunch we were unable to move forward during this and we got stuck there.

Following are the documentations we have gonethrough:

Github Repository
Convex
Clerk

3. Experiments and Results

How Success Was Measured:

- **Behavioral Realism and Consistency:** The primary measure of success was how realistically and consistently the agents behaved in simulated scenarios. This involved evaluating their conversations, decision-making, and adaptability to different situations.

Experiments Used:

- **Interaction Scenarios:** Agents were placed in various social situations in the simulated environment of Smallville to observe their interactions and behaviors.

- **Comparative Analysis:** The full generative agent architecture was compared with ablated versions (where certain features like memory or planning were disabled) and human-authored responses.

Results:

- **Quantitative Results:** The full architecture outperformed the ablated versions, as evidenced by a higher believability score (= 29.89; = 0.72). This indicated the importance of integrating all components (memory, reflection, planning) for realistic behavior.
- **Qualitative Observations:** Anecdotal evidence from specific agent responses showed that with access to full features (like reflection), agents provided more contextually relevant and coherent responses.

Success or Failure:

- **Success:** The project was largely successful in creating agents with a high degree of behavioral realism and consistency. The agents demonstrated the ability to engage in complex social interactions and adapt their behavior based on past experiences.
- **Challenges:** Despite the success, there were limitations, such as occasional inconsistencies in behavior or the influence of the underlying language model's style on agent responses.

Justification:

- **Evidence and Data:** The success is supported by the quantitative data showing the full architecture's superiority over other conditions and the qualitative evidence of the agents' realistic interactions.

- **Statistical Significance:** The statistical tests (Kruskal-Wallis test and Dunn's post-hoc tests) confirmed the significant differences in performance between the various conditions, lending further credibility to the results.

In conclusion, the "Generative Agents" project suc- ceeded in its goal to create believable and dynamic agents, as demonstrated by both the quantitative and qualitative results. The integration of a sophisticated language model

Figure 1. Home Page

Figure 2. Agent's Description

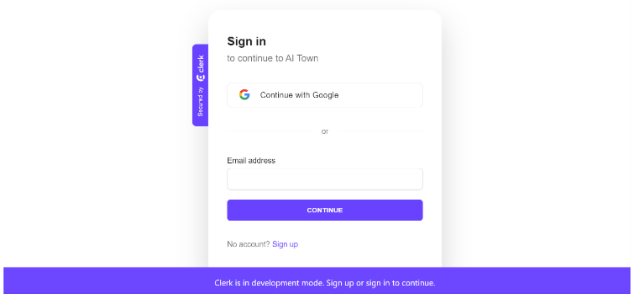

Figure 3. Login Page

with mechanisms for memory, reflection, and planning proved effective, although there is room for further improvement and refinement.

4. Understanding:

4.1. Overview of AI Town Model:

Inhabitants (AI Agents): Characters or agents within the town, each with unique behaviors, personalities, and roles. These agents could simulate the actions and interactions of real people in a town, using AI to make decisions, engage in conversations, and perform tasks.

Environmental Elements: This includes buildings (like homes, shops, schools), infrastructure (roads, parks), and other physical elements of a town. The model would ac-

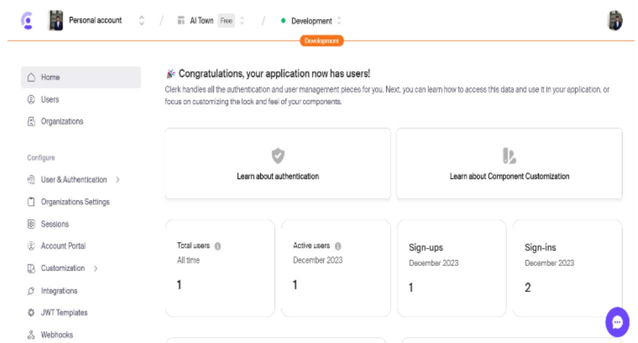

Figure 4. Clerk Dashboard

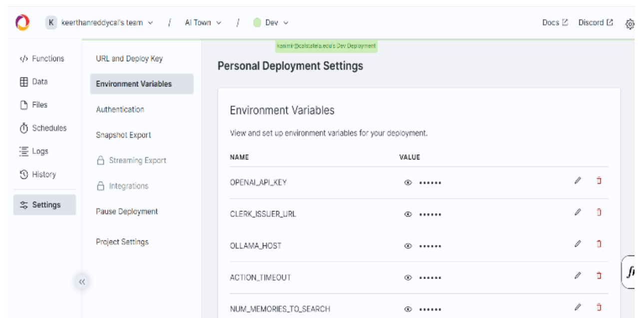

Figure 5. Convex Dashboard

count for how these elements are used and interacted with by the AI agents.

Economic and Social Systems: This could involve simulating aspects such as local economy, social interactions, and community events. AI agents would participate in these systems, contributing to and being influenced by them.

Dynamic Events and Scenarios: The model could include a range of events (like festivals, emergencies) and changing scenarios (like weather changes, economic shifts) to add realism and variability to the simulation.

Data and Analytics Layer: This layer would collect and analyze data on how the AI agents and various elements of the town interact, providing insights into the functioning and evolution of the simulated environment.

Learning and Adaptation Mechanisms: AI agents would be capable of learning from experiences and adapting their behavior, allowing for the evolution of the town environment over time.

User Interaction Interface: If the model is designed for user interaction (like in a game or educational tool), there would be an interface through which users can interact with the AI town and its inhabitants.

In an AI town data model, AI algorithms, particularly in the realm of machine learning and natural language processing, would be key in driving the behaviors and interactions within the simulation, creating an immersive and dynamic virtual town.

The AI Town Project is quite clear and should be easy to understand for someone who has studied **Deep Learn- ing**. It covers all the necessary details like what the projectis about, how it was done, what challenges were expected and faced, and what the results were. The project explains things well and uses pictures and graphs to make its pointsclearer, especially when talking about how well the different versions of the agents worked. So, if you've learned about Deep Learning, you should be able to understand most of the project, although some parts might need a bit more spe-cific knowledge about how these types of agents are built.

In the AI Town Project, we had a specific problem: making computer characters that can act like real people. This problem had a few main parts:

Talking and Acting Real: The computer characters needed to talk and act in ways that seem real, like how people do.

Remembering Stuff: They needed to remember things that happened before and use those memories to make decisions.

Changing Over Time: The characters should be able to change their actions based on new things they experience or learn.

Our model, or the way we built these characters, was made to match these parts of the problem:

Using a Chat Model: We used a special chat program (like ChatGPT) to help the characters talk and respond in a real way.

Memory and Thinking Parts:

- **Memory:** We gave the characters a way to remember things that happened.
- **Thinking:** We also made a part that lets them think about their memories and learn from them.
- **Planning:** Based on what they remember and think, the characters can make plans on what to do next.

So, the way we built the model matched the problem we were trying to solve. This helped us create characters that could talk, remember, learn, and change just like real people.

Learned Parameters:

- **Language Model:** The core component with learned parameters was the language model, like ChatGPT. This model was pre-trained on a large dataset, which means it learned from tons of text to understand and generate human-like language. It's similar to havinga lot of experience in various conversations, which it uses to respond in realistic ways.

Parts Without Learned Parameters:

- **Memory Stream:** This part didn't learn in the traditional sense. It was more like a database storing the agents' experiences. The agents used this stored information to recall past events, but the memory itself didn't 'learn' or change its behavior based on data.
- **Reflection and Planning:** These components processed the information from the memory and current situation to make decisions and plan future actions. While they used the output from the learned language model, the process of reflecting (thinking over memories) and planning (deciding future actions) did not involve learning parameters. They were more about applying rules or logic to the information at hand.

In summary, the AI Town' model had a mix of learned and non-learned parts. The learned part (language model) provided the capability to understand and generate language, while the non-learned parts (memory, reflection, planning) dealt with storing experiences, thinking about them, and making decisions.

In the "AI Town" project, where we integrated a neural network (like ChatGPT) with other components, here's how we handled the input and output, and the data processing:

Input and Output Representation:

- **Input:** The neural network expected input in the form of text. This included conversations, descriptions of past events, and current scenarios. Basically, anything the agents experienced or thought about was turned into words and sentences, like writing a diary.
- **Output:** The output from the neural network was also text. It provided responses to conversations, reflections on past events, or suggestions for future actions.

Data Pre-Processing:

- **Memory Stream:** Before feeding data into the neu- ral network, we organized the agents' past experiencesand current context. This was like summarizing what has happened and what's going on now, so the network could understand it better.
- **Formatting:** We also made sure the text data was clean and in a format that the network could process, like fixing any errors or organizing it into a consistent style.

Data Post-Processing:

- **Decisions and Actions:** After the neural network provided its output, we converted these text responses into decisions or actions for the agents. This was like reading advice and then deciding what to actually do.

Loss Function: The neural network (like ChatGPT) would have been originally trained with a loss function suited for language understanding and generation. This typically involves measuring how well the model's predictions match the expected output. However, the specific details of the loss function used in training ChatGPT aren't described in the paper.

For the AI Town part, The 'success' wasn't about minimizing a traditional loss function but rather about how believable and coherent the agents' behavior was.

Our computer characters were smart enough to handle a bunch of different situations without just repeating the same old responses. They could adapt and change their behavior in new and varied scenarios, which was exactly what we wanted them to do.

Hyperparameters: These are like settings for the model, such as how much data it should consider at once. They were likely set based on what works best for language models like ChatGPT.
Choosing Them: These settings are usually picked by trying out what works well and making adjustments.
Impact: The right settings can really help the model perform better, like making it learn faster or more accurately.
Optimizer: This is a tool that helps the model learn from its mistakes. Common ones are Adam or SGD, and they're chosen based on what's best for the model's learning style.

So, the specifics weren't detailed, but these elements are crucial for making sure the model learns effectively.

The primary Natural Language Processing (NLP) framework used was the language model ChatGPT, developed by OpenAI. ChatGPT is a variant of the GPT (Generative Pre-trained Transformer) models, which are known for their effectiveness in understanding and generating human-like text. This choice of NLP framework provided the capability for the generative agents to engage in realistic conversations and to process and generate natural language responses in a variety of interactive scenarios.

Using ChatGPT as a starting point provided a significant advantage, as it meant not having to build a complex language processing system from scratch. It allowed the focus to be on developing the unique aspects of the project, such as integrating the language model with memory, reflection, and planning mechanisms to create believable and dynamic agent behaviors.

5. Conclusion:

In conclusion, the "AI Town" project successfully pioneers a virtual community where AI characters emulate human-like behaviors, driven by the ChatGPT language model. The initiative provides a deployable toolkit for developers and explores the impact of introducing AI agents into a simulated village.

The project's approach, integrating ChatGPT with memory, thinking, and planning components, produces AI characters with human-like adaptability and realism. While the experiments demonstrate success in complex social interactions, challenges like occasional inconsistencies are acknowledged.

The AI Town model, encompassing AI agents, environmental elements, and dynamic systems, offers a versatile tool for creating immersive virtual environments. The project's success, supported by quantitative and qualitative data, opens possibilities for applications in gaming, training, and behavioral research.

In summary, the "AI Town" project makes notable strides in AI-driven social simulations, laying a foundation for future enhancements. Despite acknowledged challenges, the integration of language models with memory mechanisms proves effective, contributing to the advancement of AI research and development.

6. Work Division

Student Name	Contributed Aspects	Details
Keerthan Reddy	Environment Setup, Code Replication and Project Report	Installed all required dependencies and completed the initial setup of environment variables. Setup of Convex, Clerk, and other API-related setup.
Sai Venkat Raghava	Simulation Changes and Project Report	Implemented Implemented simulation changes such as background music variations and frequency adjustments in the convex/crons.ts file. Contributed to the Abstract and introduction sections of the Project report by reviewing research papers and analyzing code.
Harish Reddy	Code Analysis of Convex Folder, Research, and Project Report	Conducted in-depth code analysis of the convex folder. Contributed valuable insights to the project report. Conducted additional research on related papers to enhance project understanding.

Table 1. Contributions of team members.

7. References:

[1] Gavin Abercrombie, Amanda Cercas Curry, Tanvi Dinkar, and Zeerak Talat. 2023. Mirages: On Anthropomorphism in Dialogue Systems. arXiv:2305.09800 [cs.CL]

[2] Robert Ackland, Jamsheed Shorish, Paul Thomas, and Lexing Xie. 2013. How dense is a network? http://users.cecs.anu.edu.au/~xlx/teaching/css2013/ network-density.html.

[3] Eytan Adar, Mira Dontcheva, and Gierad Laput. 2014. CommandSpace: Modeling the Relationships between Tasks, Descriptions and Features. In Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (Honolulu, Hawaii, USA) (UIST ’14). Association for Computing Machinery, New York, NY, USA, 167–176. https://doi.org/10.1145/2642918.2647395

[4] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. 2014. Power to the people: The role of humans in interactive machine learning. AI Magazine 35, 4 (2014), 105–120.

[5] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N Bennett, Kori Inkpen, et al. 2019. Guidelines for human-AI interaction. In Proceedings of the 2019 chi conference on human factors in computing systems. 1–13.

[6] John R. Anderson. 1993. Rules of the Mind. Lawrence Erlbaum Associates, Hillsdale, NJ.

[7] Electronic Arts. 2009. The Sims 3. Video game.

[8] Ruth Aylett. 1999. Narrative in virtual environments—towards emergent narra- tive. In Narrative Intelligence: Papers from the AAAI Fall Symposium (Technical Report FS-99-01). AAAI Press, 83–86.

[9] Christoph Bartneck and Jodi Forlizzi. 2004. A design-centered framework for social human-robot interaction. In Proceedings of the 13th IEEE International Workshop on Robot and Human Interactive Communication (RO-MAN’04). 591– 594. https://doi.org/10.1109/ROMAN.2004.1374827

[10] Joseph Bates. 1994. The Role of Emotion in Believable Agents. Commun. ACM

37, 7 (1994), 122–125. https://doi.org/10.1145/176789.176803

[11] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique P. d.O. Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang. 2019. Dota 2 with Large Scale Deep Reinforcement Learning. arXiv preprint arXiv:1912.06680 (2019).

[12] Marcel Binz and Eric Schulz. 2023. Using cognitive psychology to under- stand GPT-3. Proceedings of the National Academy of Sciences 120, 6 (2023), e2218523120.

[13] BioWare. 2007. Mass Effect. Video game.

[14] Woody Bledsoe. 1986. I had a dream: AAAI presidential address. AI Magazine 7, 1 (1986), 57–61.

[15] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, and et al. 2022. On the Opportunities and Risks of Foundation Models. arXiv:2108.07258 [cs.LG]

[16] Michael Brenner. 2010. Creating dynamic story plots with continual multiagent planning. In Proceedings of the 24th AAAI Conference on Artificial Intelligence.

[17] Rodney A. Brooks, Cynthia Breazeal, Marko Marjanovic, Brian Scassellati, and Matthew Williamson. 2000. The Cog Project: Building a Humanoid Robot. In Computation for Metaphors, Analogy, and Agents (Lecture Notes on Artificial Intelligence, 1562), Chrystopher Nehaniv (Ed.). Springer-Verlag, Berlin, 52–87.

[18] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie

Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. arXiv:2005.14165 [cs.CL]

[19] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric

Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. arXiv preprint arXiv:2303.12712 (2023).

[20] Robin Burkinshaw. 2009. Alice and Kev: The Story of Being Homeless in The Sims 3.

[21] Chris Callison-Burch, Gaurav Singh Tomar, Lara Martin, Daphne Ippolito, Suma Bailis, and David Reitter. 2022. Dungeons and Dragons as a Dialog Challenge for Artificial Intelligence. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 9379–9393. https://aclanthology.org/2022.emnlp- main.637

[22] Stuart K Card, Thomas P Moran, and Allen Newell. 1980. The keystroke- level model for user performance time with interactive systems. Com- mun. ACM 23, 7 (1980), 396–410. https://doi.org/10.1145/358886.358895 arXiv:https://doi.org/10.1145/358886.358895

[23] Stuart K Card, Thomas P Moran, and Alan Newell. 1983. The psychology of human-computer interaction. (1983).

[24] Alex Champandard. 2012. Tutorial presentation. In IEEE Conference on Compu- tational Intelligence and Games.

[25] Dong kyu Choi, Tolga Konik, Negin Nejati, Chunki Park, and Pat Langley. 2021. A Believable Agent for First-Person Shooter Games. In Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, Vol. 3. 71–73.

[26] Anind K Dey. 2001. Understanding and using context. Personal and ubiquitous computing 5 (2001), 4–7.

[27] Kevin Dill and L Martin. 2011. A Game AI Approach to Autonomous Con- trol of Virtual Characters. In Proceedings of the Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC'11). Orlando, FL, USA.

[28] David Easley and Jon Kleinberg. 2010. Networks, crowds, and markets: Reasoning about a highly connected world. Cambridge university press.

[29] Arpad E Elo. 1967. The Proposed USCF Rating System, Its Development, Theory, and Applications. Chess Life XXII, 8 (August 1967), 242–247.

[30] Jerry Alan Fails and Dan R Olsen Jr. 2003. Interactive machine learning. In Proceedings of the 8th international conference on Intelligent user interfaces. ACM, 39–45.

[31] Ethan Fast, William McGrath, Pranav Rajpurkar, and Michael S Bernstein. 2016. Augur: Mining human behaviors from fiction to power interactive systems. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems. 237–247.

[32] Rebecca Fiebrink and Perry R Cook. 2010. The Wekinator: a system for real-time, interactive machine learning in music. In Proceedings of The Eleventh Interna- tional Society for Music Information Retrieval Conference (ISMIR 2010)(Utrecht), Vol. 3. Citeseer, 2–1.

[33] Uwe Flick. 2009. An Introduction to Qualitative Research. SAGE.

[34] James Fogarty, Desney Tan, Ashish Kapoor, and Simon Winder. 2008. CueFlik: Interactive Concept Learning in Image Search. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Florence, Italy) (CHI '08). Association for Computing Machinery, New York, NY, USA, 29–38. https:

//doi.org/10.1145/1357054.1357061

[35] Adam Fourney, Richard Mann, and Michael Terry. 2011. Query-feature graphs: bridging user vocabulary and system functionality. In Proceedings of the ACM Symposium on User Interface Software and Technology (UIST) (Santa Barbara, California, USA). ACM.

[36] Tom Francis. 2010. The Minecraft Experiment, day 1: Chasing Water- falls. http://www.pcgamer.com/2010/11/20/the-minecraft-experiment-day-1-chasing-waterfalls/

[37] Jonas Freiknecht and Wolfgang Effelsberg. 2020. Procedural Generation of Interactive Stories using Language Models. In International Conference on the Foundations of Digital Games (FDG '20). ACM, Bugibba, Malta, 8. https://doi. org/10.1145/3402942.3409599

[38] Tianyu Gao, Adam Fisch, and Danqi Chen. 2020. Making Pre-trained Language Models Better Few-shot Learners. CoRR abs/2012.15723 (2020). arXiv:2012.15723 https://arxiv.org/abs/2012.15723

[39] Perttu Hämäläinen, Mikke Tavast, and Anton Kunnari. 2023. Evaluating Large Language Models in Generating Synthetic HCI Research Data: a Case Study. In Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems. ACM.

[40] Matthew Hausknecht, Prithviraj Ammanabrolu, Marc-Alexandre Cote, and Xinyu Yuan. 2020. Interactive Fiction Games: A Colossal Adventure. In Pro- ceedings of the AAAI Conference on Artificial Intelligence, Vol. 34. 7903–7910. https://doi.org/10.1609/aaai.v34i05.6297

[41] Chris Hecker. 2011. My Liner Notes for Spore. http://chrishecker.com/My_liner_ notes_for_spore