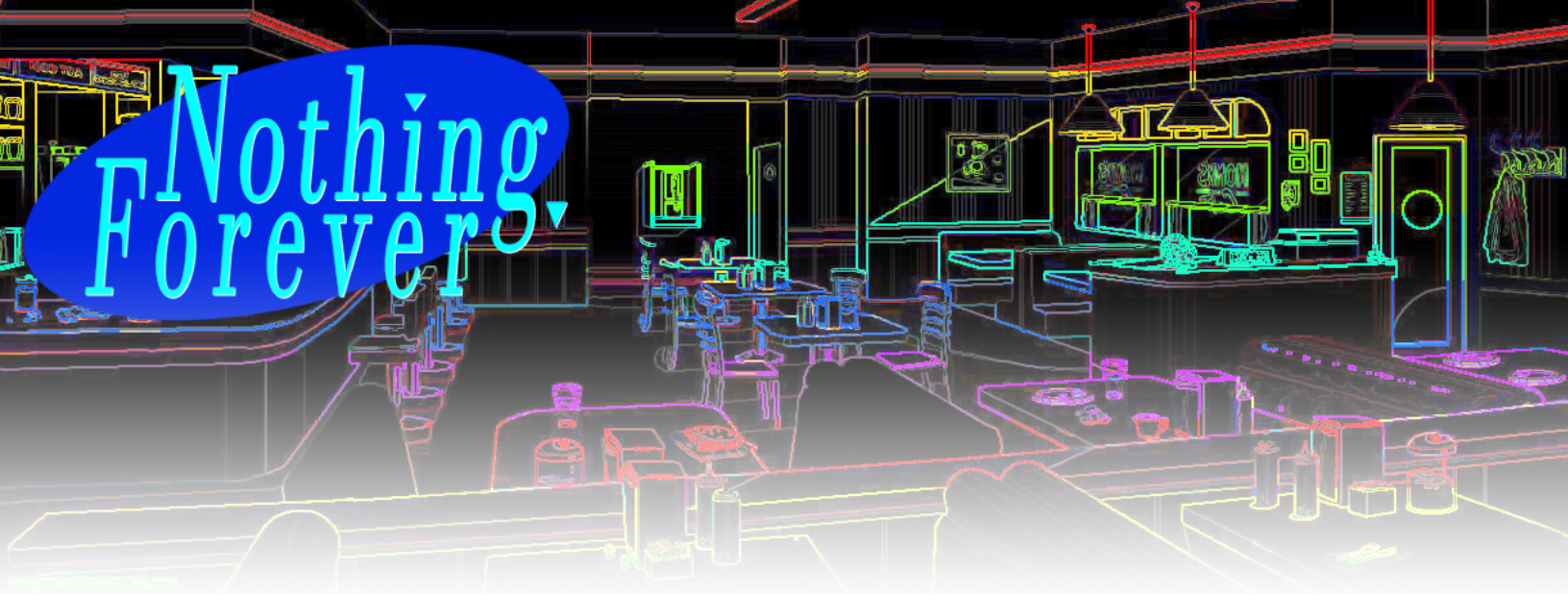


Executive summary

"Nothing, Forever" is an algorithmically-generated live-streamed parody of Seinfeld incorporating other references and elements of 90s media and culture. The script is generated via machine learning (ML), which enables "Nothing, Forever" to be broadcast perpetually – it is always on. Additionally, "Nothing, Forever" interfaces with live-streaming platforms such as Twitch.tv, allowing for audience participation in elements of the show through Follows, Subscriptions, and chat interaction.

"Nothing, Forever" is the first example of a potential new media format that leverages the power of AI and ML technology to create compelling interactive content at a scale that would be impossible for traditional media. It's always on, and it's always new.

Creative

A short concept video of "Nothing, Forever" has been produced to demonstrate aspects of the project. Art from Jacob Janerka was used as a stand-in (unlicensed, video is unlisted and not for broad distribution). All dialogue in the demo was actually produced via machine learning. See video here.

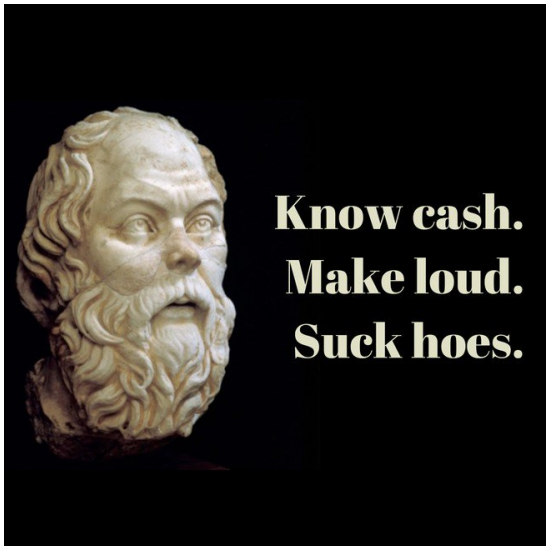
Overview



"Nothing, Forever" features as a primary element a 90s-style sitcom based on Seinfeld. The visuals for this element are in the style of a 90s PC point-and-click adventure game, such as "Day of the Tentacle" or "Monkey Island". Elements of the scene are randomly generated on the fly as the

show is live-streamed. Art assets, including sets (e.g. Jerry's apartment, the diner, the club, etc.) and characters (the cast of Seinfeld plus crossover guests) are chosen from an asset library. Scrips are generated by character-specific ML "screenwriter" bots, which have been trained using the scripts from Seinfeld. Scripts are converted to audio using custom text-to-speech programs generated from the voices of Seinfeld character impersonators. Other audio elements, such as music and laugh track, are controlled by an AI "director" program, following rules that mimic the conventions of American sitcoms.

While the content generated via the above approach will be absurd, the format of the sitcom will enable audiences to relate to the show. As with other examples of ML-generated content, such as Inspirobot (https://www.thisamericanlife.org/extras/why-i-love-inspirobot, http://inspirobot.me/), placing absurd or nonsensical language into a highly familiar and structured format facilitates interpretation of the absurdity as humorous and occasionally profound.



Interactivity and familiarity are key to the success of modern social media. Famous Youtubers appear side-by-side with your grandmother on your Facebook feed, sharing recipes and makeup tips. Watching a Twitch.tv stream of someone playing a videogame while interacting with viewers in chat evokes similar feelings as hanging out in your childhood friend's basement, playing video games while hoping his parents don't notice how late it's gotten. At first blush, algorithmically-generated media lacks these features.

However, consider the success of "Twitch Plays Pokemon", a live-streaming project in which Twitch chat participants directly controlled the controller inputs to the Nintendo classic. This project similarly combined the familiar with the absurd, as game tasks that are trivial for a single mind became monumental feats for the wild crowd, which contained a combination of viewers acting for and against the game objectives. Meta-content emerged, with efforts to organize and execute strategies that lead the hive mind to complete what had been thought impossible.

"Nothing, Forever" will similarly feature autonomous audience interaction and participation events. Content will be unlocked as certain viewer thresholds are met. For example, the show might begin with Jerry alone in the comedy club, monologuing to a small livestream audience. At programmable levels of viewership, followers, or subscribers, other characters and sets can be unlocked. Additionally, content-altering Events will occur periodically, in which viewers' chat inputs will influence the show. An example Event is the "Jerry-meter", which is a gauge that appears in the lower corner of the screen. By typing "More Jerries!" or similar text, the meter can be filled. Once filled, additional instances of Jerry can appear in a scene. Other audience-triggered actions include "George explodes" and "Newman turns into a spider (permanently)".

Another exciting element of live content across all platforms is the potential to see something go wrong. "Nothing, Forever" will feature occasional video effects, similar to VHS and CRT glitches, that suggest to the viewer that something is not quite right in the show's universe. Rarely, the

glitches will grow out of control, and a number of interesting events can unfold. The sets may be replaced with alternate-universe sets inspired by Moonside (Earthbound, 1994) or Silent Hill (1999); multiple Jerries in these universes may sync up into a hive mind. In other post-glitch scenes, additional sets and characters associated with 90s-media "other places/dimensions" appear, such as the Black Lodge from Twin Peaks.



The glitches and post-glitch scenes hint at the true agent behind the show – the goblin. The goblin is an entity in the style of David Lynch's "man behind Winkie's" or "man from another place", an incomprehensible creature who is responsible for the events that unfold in the show. Its motivations are never disclosed, and it is only rarely seen in a glimpse between video glitches. The goblin does not like being seen, and it does not like it when the show "goes wrong". Goblin scenes are intended to be live-action, depicting the goblin (poorly, as though through a security camera) in his lair as he attempts to fix or maintain the strange machines that produce the show (most of which are visually inspired by 80s-90s era obsolete electronics).

Sometimes, instead of streaming the screen-captured video of the Seinfeld parody show, the show is instead displayed on a CRT in the goblin's lair, and a camera pointed at the CRT is streamed to Twitch.tv.



Technical

The Show Loop

The primary moment-to-moment experience of the show is the Seinfeld parody “Show”. The Show is constructed procedurally by a Director AI within a Game Engine. The Director chooses the set, characters, number of lines, amount of laughtrack, and other scene-related information and provides instructions to the Game Engine on how to execute the scene. The Director’s choices are constrained and guided by a set of Rules, stored in a game state database. These Rules govern how frequently different characters may appear in sets and scenes, how long a scene is, how frequently laughtrack may occur, what sets and characters have been unlocked, etc. Different sets of Rules may be stored in a collection, for example a “Must-See TV Ruleset” might be a rule set with a greater frequency of alternative costumes and animations (e.g. Jerry’s puffy shirt or Elaine’s dance) as compared to a “Basic Ruleset”. Programmable Events (described below) may be used to set the active Ruleset that is accessible to the Director.

An example of the scene-based Show loop is given below.

1. A new scene is needed (triggered by system initialization or end of prior scene)
2. The Director calls the Rules database to set scene parameters
3. The Rules database returns an object containing the relevant rules for this scene
    a. Available sets: Jerry’s Apartment, the Club, The Diner
    b. Available characters: Set-dependent
        i. 60%: Jerry’s Apartment (2-4 characters w/probability distribution)
            - Jerry (99%), George (70%), Elaine (50%), Kramer (50%)
        ii. 10%: The Club (1 character) Jerry (100%)
        iii. 30%: The Diner (2-4 characters)
            - Jerry (70%), George (70%), Elaine (70%), Kramer (40%)
    c. Distribution function for frequency of lines per character
    d. Distribution function for frequency/type of laughtrack
    e. Distribution functions for other scene info (dialogue timing, camera shots/cuts, character movement/animation, etc.)
4. The Director “rolls the dice” to determine all of the above criteria for the scene
5. With the characters and number of lines determined, the Director calls the Language service to generate lines
6. The Language service generates dialogue for each character using the appropriate model, then submits it to a text-to-speech service, and returns the received audio files to the Director
7. The Director uses the time metadata for the audio files along with the dialogue timing rules and laughtrack frequency rules to construct an ordered timeline script for the scene
8. The Director requests the Game to set up the scene with the appropriate assets
9. The Director provides the script to the game for execution
10. During execution, the Director monitors for any scene-interrupting Event content

11. Steps 1-7 can be performed in preparation for the subsequent scene during the execution of the present scene (if any Rules changes occur during or between scenes, the prepared scene data must be discarded and re-rolled with the new Rules).

The Director may also prepare scene transitions, which typically feature an exterior shot of the upcoming set, along with a brief musical cue. At the conclusion of an “Episode” (roughly every 30 minutes), a credit-roll appears, featuring lengthier music as well as scrolling text. The text may be ML-generated nonsense, or may feature the names of project supporters (streaming platform subscribers, Patreons, donators, etc.).

Events

Nothing, Forever features audience interactive “Events”, through which viewers can impact aspects of how the show is performed and progressively update the assets and narrative components. Events can occur periodically, on a random basis, or can be triggered by specific criteria (e.g. number of concurrent viewers, a specific moment in time (Thursday at 8 pm), subscriber privileged actions) or combination of criteria. Events may (or may not) feature an interactive goal of some sort. Typically, during an Event, some information about the event and its interactions is provided to the viewers through an overlay. The overlays may be custom to the event, and may resemble the bottom-screen “chyron”/”lower thirds” utilized in TV broadcasts to communicate “Next on Fox…” or “News at 11…”, but with a period-Seinfeld visual style. Overlays may additionally feature meters or gauges used to indicate progress toward that Event’s goal.



Example of bottom-third chyron.

For Events with a goal, typically upon reaching that goal, the show is modified in some way. For example, completion of a goal may result in unlocking a new aesthetic feature, such as a puffy shirt for Jerry, or a dancing animation for Elaine. Some Events may have multiple possible outcomes, such as “success” and “failure” outcomes that both resultantly modify the media in different ways. Some Events may require multiple instances to complete, and thus will need to store information about the state of the Event between instances. Completion of some events unlocks the possibility of other events, thus there can be a linear or tree-like progression of Events and outcomes.

Upon reaching an outcome, the overlay communicates that the outcome has been reached. The outcome may modify the current scene in progress, for example, by causing a character to enter the current scene. The outcome may modify the pool of resources used to generate the show,

for example by unlocking a new set. In this case, the current scene carries on as usual. The outcome may modify the rules used by the Director to construct the show, for example changing the frequency of laughtrack. The outcome may cause some abrupt change in the media, such as “changing the channel” to some other video stream.

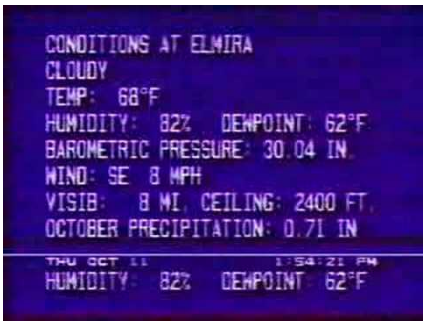
Other types of Events may modify the media and Director systems while the event is on. For example, during the Event, alternative art assets may be used to visually represent that the show is “broken” somehow. After such an Event is over, assets and rules return to their baseline behavior.

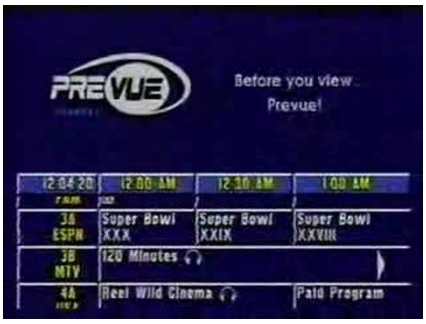
Non-prescriptive examples of Event definition are documented here and here.

Secondary content

While the primary experience is described by the Show Loop, other types of content are used to provide variety, advance the narrative, and flesh out the world of Nothing Forever. This content represents “commercial breaks”, “changing the channel”, and “seeing behind the curtain”. Commercial breaks and changing the channel are similar, displaying period-style TV content based on generative algorithms and licensed music. For example, commercials can be







generated using text trained from advertising copy sources such as Home Shopping Networks, which also typically featured significant amounts of on-screen text. Other generative content can be based on the Weather Channel and Prevue Channel, with correspondingly trained language models. This text may also be used as a medium for advancing storytelling, while the content can be accompanied by period-style music. An example video has been produced here. Content in this category can also be used for chat-interactive games that do not center on the Seinfeld parody show. A model for such games might be “Wario Ware for a crowd”, featuring similar simple interactions for the player (the audience) to complete.

The last category of secondary content is “seeing behind the curtain”. As discussed above, within the fictional world of Nothing Forever, the show is being created and broadcast by a mysterious entity. Live-action footage of the Goblin in its lair will be used rarely to communicate this narrative.

Technical

Most creative elements of the show (except for the live-action goblin scenes) are accomplished through custom media and artificial intelligence software. The initially envisioned technical elements are the Director, the Screenwriters, the Game, the Voice Actors, and the Composer. The responsibilities and implementation of these technical elements will likely be adjusted during development.

The Director

The Director is the most complex and most important element of “Nothing, Forever”. The Director utilizes a system of rules that govern the structure and format of a sitcom to produce the show.

Scenes are constructed first by randomly determining the cast, set, and length of a scene. Once the cast is established, the Director will call the corresponding Screenwriters to build a script for the scene. The Director will send instructions to the Game indicating the art assets that are needed, then will provide control inputs to the game. These inputs include character movement, character animation, camera movement, scene events (such as Kramer entering Jerry’s apartment), etc. The Director passes the generated scripts to the Voice Actors and triggers the audio performance in sync with the character animation. The Director controls and times the laugh track and music for a scene. Additionally, the Director controls and triggers special events, which requires it to observe data from Twitch.tv, such as the number of viewers, followers, subscribers, and the text from chat participants.

The Screenwriters

The Screenwriters are individual ML “bots” that have been trained using the scripts of Seinfeld. For example, one Screenwriter is a Jerry-bot that was trained using only Jerry’s lines from the sitcom. At a minimum, “Nothing, Forever” requires one Screenwriter for each major cast member, and one “generic” screenwriter for other characters. If resources and sufficient script data are available, additional specific Screenwriter bots can be created for minor Seinfeld characters and guest/event characters.

The Game

The major visual element of “Nothing, Forever” is created inside a game-like program, which looks and functions like a 90s-style PC point-and-click adventure game. Scenes take place in sets that have defined walkable space and objects, and characters can be moved about these scenes via inputs to the game. Animations and other behaviors can also be triggered. Further, the camera is fully controlled via inputs. The “player” of the Game is the Director program.

The Game pulls art assets from a library, based on requests from the Director. This library can be updated as “Nothing, Forever” evolves, allowing for new sets, characters, animations, etc. to be added as the show progresses.

Due to the need to generate audio on the fly, the Game does not contain or utilize any audio elements. All audio is controlled by the Director, who sequences in-Game-media such as speech animation to synchronize with the generated audio.

The Composer

The music in Seinfeld is unusual, in that it was actually a custom improvisation created for each episode around the now-iconic slap bass riff. See this interview with the composer, Jonathan Wolff. According to Wolff: “Each episode, I re-do the theme for Seinfeld to match the pacing of [Jerry’s] monologue.”

The theme is played on a MIDI keyboard with a small number of sound samples. As a result of these unusual musical choices, the Seinfeld musical style can likely be reproduced by an algorithm following certain rules about the timing of bass notes, trumpet, and percussion. This will allow for generation of Seinfeld-like music without directly appropriating any media from the series.

The Voice Actors

The dialogue in “Nothing, Forever” will be written by the Screenwriters and performed by custom text-to-speech modules. There are businesses that offer “extremely realistic artificial voices based on real people's voices”. Since such artificial voices can be generated from samples of real speech, actors will be hired to impersonate the vocal style of the main Seinfeld cast. The recorded samples of these actors will be used to create the Voice Actors.







Assorted Seinfeld art from unpublished adventure game project (unrelated to “Nothing, Forever”), by [Jacob Janerka]