

Experiments in Paired Storytelling: 2024 Dataset Description

Introduction

This document describes a dataset was collected from an interactive narrative game played by 25 University of Kentucky students in an undergraduate game development class exercise who signed a consent form to release their data. The dataset includes 1005 actions across 34 game sessions (some incomplete).

The dataset was collected as follows. At the beginning of the class period, each student was given a sticky note with a unique ID to log into the game anonymously. The instructor described the game to the class, including a brief backstory: Just before the coronation of a kingdom's future ruler, a bandit stole the enchanted crown that can magically grant anyone the status of monarch; now, the would-be heir is offering a bounty to the player character to bring the crown back, although the player is free to choose whether to pursue that quest or to help someone else usurp the throne instead. The instructor led a tutorial where the class played an example session (not included in the dataset) with a presupplied sequence of actions. Then students played until the end of the class period; the server paired them together for each game session, ensuring that nobody was placed in the same role or with the same partner twice in a row. At the end, the instructor tied the exercise into the course with a discussion about how a game master or game designer can implicitly communicate about intentions with the player. Then, students were given the option to volunteer their game data for research by signing a consent form.

The next two sections describe the game itself to provide context for the data, and the last section describes the format of the dataset itself in this archive.

Game Interface

Each game session had two users, one taking the role of the *game master* and one taking the role of the *player*. Both users played through a browser interface like the one in the image below.

State	Game	Reference
<p>Adventurer Location: Alchemy Shop Status: normal Inventory: Coin, Food, Invisibility Potion</p> <p>Alchemist Location: Alchemy Shop Status: normal Inventory: Poison</p> <p>Bandit Location: Wilderness Status: normal Inventory: Cutlass, Crown, Lockpick</p> <p>Heir Location: Palace Status: normal Inventory: Royal Treasure</p> <p>Guard Location: Town Square Status: normal Inventory: Longsword, Key to the Stocks</p>	<p>You are the Game Master (GM). Your partner controls the Adventurer character; you control the rest.</p> <p>It is time for the kingdom to crown a new ruler. But just before the Heir's coronation ceremony, a Bandit ran off with the Crown. The Heir has promised the Adventurer a bounty of Royal Treasure to bring back the Crown. But the magical Crown has the power to turn anyone who activates its magic inside the Palace into the all-powerful monarch, and many covet the position. Who will claim the throne?</p> <p>The Adventurer goes to the Alchemy Shop. The Alchemist gives the Adventurer the Invisibility Potion.</p> <p>Choose a character to act next. (The player is paused and can act only if you pass control.)</p> <ul style="list-style-type: none"> • Alchemist • Bandit • Guard • Heir • Other actions • Let the player act 	<p>Characters</p> <p>Adventurer The player character.</p> <p>Alchemist Peddler of potions and poisons.</p> <p>Bandit This outlaw stole the Crown, and may someday use it steal the kingdom itself.</p> <p>Guard Sworn to serve the Heir in upholding the law of the kingdom.</p> <p>Heir The would-be next member of the royal line, looking to regain the Crown.</p> <p>Items</p> <p>Coin A currency. Can be traded for other items.</p> <p>Crown Ends the game if brought into the Palace</p>

The left pane showed the current properties of game objects such as characters; for the game master, this included all game information, while for the player, it was restricted to characters visible in the player character's current location.

The middle pane showed game narration and a menu for selecting *actions*. Actions were the means by which properties of game objects were updated and new narration was produced. Only one user had action menus available at any given time; by default, the menus were available to the game master, who could allow the player to choose actions by selecting "Let the player act". After the player chose an action, the game master could continue ceding control by clicking "Let the player keep acting" or resume choosing actions by clicking "Take back control".

The right pane had a scrollable reference guide with short descriptions of characters, locations, items, and actions.

A box like the one below appeared at the bottom of each user's screen periodically, upon the other user gaining control and no more than once per minute. Before gameplay continued, the user was required to submit ratings on a five-point Likert scale for two qualities: "Every action in this story is important and well-integrated into narrative [sic]" and "I feel free to make meaningful choices that will influence this story".

Please rate your agreement with these statements about the story so far.

Every action in this story is important and well-integrated into narrative.

Completely disagree Disagree Neither agree nor disagree Agree Completely agree

I feel free to make meaningful choices that will influence this story.

Completely disagree Disagree Neither agree nor disagree Agree Completely agree

After the game session ended, each user was given a post-game interview illustrated below where they were asked to explain their own and their partner's action choices by selecting one or more checkboxes, and in the case of the "Other" checkbox, typing additional details.

Review the story and answer these questions.

- The Adventurer goes to the Alchemy Shop.
- The Alchemist gives the Adventurer the Invisibility Potion.
- The Adventurer waits idly for a while.
- The Adventurer consumes the Food.

Why do you think your partner chose this action?

☐ To help the Adventurer become the new monarch

☐ To help the Alchemist become the new monarch

☐ To help the Bandit become the new monarch

☐ To help the Guard become the new monarch

☐ To help the Heir become the new monarch

☐ To add conflict or suspense to the story

☐ To make the Adventurer feel realistic

☐ Other (please elaborate)

Game Mechanics

This section explains the actions that appear in the game logs. Each action operated on game objects from among the following types.

- **Locations** have adjacent locations.
 - The **Town Square** location is adjacent to all other locations.
 - The **Alchemy Shop**, **Palace**, and **Wilderness** locations are adjacent only to the Town Square.
- **Characters** have a controlling user from among the player and game master and a current location. Characters also have a status: *normal* is the default; *invisible* characters cannot be seen by other characters; and *dead* or *detained* characters cannot take actions unless otherwise specified. By default, when a character takes an action that targets another character, the target must be a character of *normal* status in the same location.
 - The **Adventurer** character is controlled by the player and starts in the Town Square.
 - The **Alchemist** character is controlled by the game master and starts in the Alchemy Shop.
 - The **Bandit** character is controlled by the game master and starts in the Wilderness.
 - The **Guard** character is controlled by the game master and starts in the Town Square.
 - The **Heir** character is controlled by the game master and starts in the Palace.
- **Items** have a possessing character or container. **Weapons**, **Consumables**, and **Currency** are subtypes.
 - The **Coin** currency and **Food** consumable start with the Adventurer.
 - The **Invisibility Potion** consumable and **Poison** item start with the Alchemist.
 - The **Crown** item, **Cutlass** weapon, and **Lockpick** item start with the Bandit.
 - The **Key to the Stocks** item and **Longsword** weapon start with the Guard.
 - The **Royal Treasure** currency starts with the Heir.

Although actions were communicated to users in the form of natural-language flavor text, they are represented in the logs as *action signatures* consisting of the action name and list of parameters. Spaces are included for clarity in this document but all whitespace is removed in the log itself. The actions in the game were as follows:

- **applyPoison([character], [consumable])**: Available to the character when they have the consumable and the Poison item. The Poison is destroyed and the consumable gains a hidden *poisoned* property.

- **attack([character], [character]):** Available to first character when that character has a weapon. The game master chooses whether the action succeeds; if so, the second character's status becomes *dead*.
- **becomeMonarch([character]):** Available to the character when they have the Crown item and are at the Palace. The game ends with the character announced as the monarch.
- **buy([character], [character], [currency], [item]):** Available to the controller of the first character when they have the currency, the second character has the other item, and the other item is not a currency. The second character chooses whether to allow the action; if so, the characters exchange possession of the items.
- **consume([character], [consumable]):** Available to the character when they have the consumable. The consumable is destroyed. If the consumable is poisoned, the character becomes *dead*. Otherwise, if the consumable is the Invisibility Potion, the character becomes *invisible*.
- **detain([character], [character]):** Available to the first character when they have the Key to the Stocks item. Both characters' location becomes the Town Square and the second character's status becomes *detained*.
- **endInvisibility([character]):** Available to the game master when the character's status is *invisible*. The character's status becomes *normal*.
- **give([character], [character], [item]):** Available to the controller of the first character when they have the item. The second character gains the item.
- **hunger([character]):** Available to the game master. Has no functional effect on the game state; only displays flavor text that the character's stomach growls.
- **loot([character], [character], [item]):** Available to the controller of the first character when the second character possesses the item and has status *dead* or *detained*. The first character gains the item.
- **playerDiedEnding():** Available to the game master when the Player character is dead. The game ends with no character as the monarch.
- **release([character], [character]):** Available to the first character when they have the Lockpick or Key to the Stocks item and the second character's status is *detained*. The second character's status becomes *normal*. May be used by a *detained* character with themselves as the target.
- **sell([character], [character], [item], [currency]):** Available to the controller of the first character when they have the first item, the first item is not a currency, and the second character has the other item. The second character chooses whether to allow the action; if so, the characters exchange possession of the items.

- **take([character], [character], [item]):** Available to the controller of the first character when the second character has the item. The second character chooses whether to allow the action; if they do not, the game master may choose to force the action to succeed anyways. If the action succeeds, the first character gains the item.
- **timePasses():** Available to the game master. Has no functional effect on the game state; only displays flavor text that some time has passed.
- **travel([character], [location]):** Available to the controller of the character when the character is at an adjacent location. The character's location becomes the chosen location.
- **wait([character]):** Available only to the Adventurer character. Has no functional effect on the game state; only displays flavor text that the Adventurer waits for a while..

Data Contents

Each time a user took an action in the game session, a corresponding row was added to the game logs in the dataset. Entries in *gameEvents.csv*, after an initial header row, have the following columns in order:

- **sessionID:** Assigned uniquely to each game session.
- **eventID:** Within a game session, assigned uniquely to an instance of an in-game action; i.e., each combination of a *sessionID* and an *eventID* is unique across the whole dataset.
- **timestamp:** The real-world date and time (EST) of the event.
- **gmID:** The user ID of the session's game master.
- **playerID:** The user ID of the session's player.
- **action:** The action signature for the action taken.
- **userWhoChose:** The user ID of the user who selected the action (either the *gmID* or the *playerID*).
- **targetResponse:** *allow* if the action elicited a target character's consent and the character allowed the action; *deny* if the character tried to prevent the action; blank if no response was elicited from a target character. (For instance, for a *buy* action, this field is *allow* if the target character accepted the trade offer and *deny* if they declined the trade.)
- **gmRuling:** *success* if the action elicited the game master's decision about action success and the game master allowed the action; *failure* if the game master denied the action; blank if no response was elicited from the game master. (For instance,

for a *take* action where the target tried to *deny* the character taking the item, this field is *success* if the game master ruled that the taker's attempt succeeded anyways, and *failure* if the game master ruled that the original owner managed to keep the item.)

- **outcome:** The result if the current action prompted further choices (e.g., the decision to accept or decline a trade offer). Blank if no such choices were prompted.
- **gmStructureRating:** The game master's Likert scale rating for structure from 1 for "Completely Disagree" to 5 for "Completely Agree". Present only if the game master submitted a rating after the current action and before any other action. Blank if the user did not submit a rating during that timeframe.
- **gmAgencyRating:** The game master's Likert scale rating for agency from 1 for "Completely Disagree" to 5 for "Completely Agree". Present only if the game master submitted a rating after the current action and before any other action. Blank if the user did not submit a rating during that timeframe.
- **playerStructureRating:** The player's Likert scale rating for structure from 1 for "Completely Disagree" to 5 for "Completely Agree". Present only if the player submitted a rating after the current action and before any other action. Blank if the user did not submit a rating during that timeframe.
- **playerAgencyRating:** The player's Likert scale rating for agency from 1 for "Completely Disagree" to 5 for "Completely Agree". Present only if the player submitted a rating after the current action and before any other action. Blank if the user did not submit a rating during that timeframe.
- **gmExplanation_AdventurerMonarch:** Boolean for whether the game master selected the "To help the Adventurer become the new monarch" checkbox when the post-game interview asked to explain the choice of action. Blank if the post-game interview did not ask about the current action or if the interview was not completed because the user exited the game prematurely.
- **gmExplanation_AlchemistMonarch:** Boolean for whether the game master selected the "To help the Alchemist become the new monarch" checkbox when the post-game interview asked to explain the choice of action. Blank if the post-game interview did not ask about the current action or if the interview was not completed because the user exited the game prematurely.
- **gmExplanation_BanditMonarch:** Boolean for whether the game master selected the "To help the Bandit become the new monarch" checkbox when the post-game interview asked to explain the choice of action. Blank if the post-game interview did

not ask about the current action or if the interview was not completed because the user exited the game prematurely.

- **gmExplanation_GuardMonarch:** Boolean for whether the game master selected the “To help the Guard become the new monarch” checkbox when the post-game interview asked to explain the choice of action. Blank if the post-game interview did not ask about the current action or if the interview was not completed because the user exited the game prematurely.
- **gmExplanation_HeirMonarch:** Boolean for whether the game master selected the “To help the Heir become the new monarch” checkbox when the post-game interview asked to explain the choice of action. Blank if the post-game interview did not ask about the current action or if the interview was not completed because the user exited the game prematurely.
- **gmExplanation_Characterization:** Boolean for whether the game master selected the “To make the [character] feel realistic” checkbox when the post-game interview asked to explain the choice of action, where [character] was replaced with the character who took the action. Blank if the post-game interview did not ask about the current action or if the interview was not completed because the user exited the game prematurely.
- **gmExplanation_Conflict:** Boolean for whether the game master selected the “To add conflict or suspense to the story” checkbox when the post-game interview asked to explain the choice of action. Blank if the post-game interview did not ask about the current action or if the interview was not completed because the user exited the game prematurely.
- **gmExplanation_Other:** The text entered by the game master if they selected the “Other (please elaborate)” checkbox when the post-game interview asked to explain the choice of action. Blank if the user did not select this checkbox, if the post-game interview did not ask about the current action, or if the interview was not completed because the user exited the game prematurely.
- **playerExplanation_AdventurerMonarch:** Boolean for whether the player selected the “To help the Adventurer become the new monarch” checkbox when the post-game interview asked to explain the choice of action. Blank if the post-game interview did not ask about the current action or if the interview was not completed because the user exited the game prematurely.
- **playerExplanation_AlchemistMonarch:** Boolean for whether the player selected the “To help the Alchemist become the new monarch” checkbox when the post-game interview asked to explain the choice of action. Blank if the post-game

interview did not ask about the current action or if the interview was not completed because the user exited the game prematurely.

- **playerExplanation_BanditMonarch:** Boolean for whether the player selected the “To help the Bandit become the new monarch” checkbox when the post-game interview asked to explain the choice of action. Blank if the post-game interview did not ask about the current action or if the interview was not completed because the user exited the game prematurely.
- **playerExplanation_GuardMonarch:** Boolean for whether the player selected the “To help the Guard become the new monarch” checkbox when the post-game interview asked to explain the choice of action. Blank if the post-game interview did not ask about the current action or if the interview was not completed because the user exited the game prematurely.
- **playerExplanation_HeirMonarch:** Boolean for whether the player selected the “To help the Heir become the new monarch” checkbox when the post-game interview asked to explain the choice of action. Blank if the post-game interview did not ask about the current action or if the interview was not completed because the user exited the game prematurely.
- **playerExplanation_Characterization:** Boolean for whether the player selected the “To make the [character] feel realistic” checkbox when the post-game interview asked to explain the choice of action, where [character] was replaced with the character who took the action. Blank if the post-game interview did not ask about the current action or if the interview was not completed because the user exited the game prematurely.
- **playerExplanation_Conflict:** Boolean for whether the player selected the “To add conflict or suspense to the story” checkbox when the post-game interview asked to explain the choice of action. Blank if the post-game interview did not ask about the current action or if the interview was not completed because the user exited the game prematurely.
- **playerExplanation_Other:** The text entered by the player if they selected the “Other (please elaborate)” checkbox when the post-game interview asked to explain the choice of action. Blank if the user did not select this checkbox, if the post-game interview did not ask about the current action, or if the interview was not completed because the user exited the game prematurely.