Towards Procedural Generation of Narrative Puzzles for Open World Games

Sam Davern(✉) and Mads Haahr

Trinity College Dublin, University of Dublin, Dublin, Ireland
{daverns,haahrm}@tcd.ie

Abstract. Story Puzzle Heuristics for Interactive Narrative eXperiences (SPHINX) was developed as a system for generating narrative puzzles in adventure games. While SPHINX is capable of generating narrative puzzles for narrative games with a progressive structure, it cannot be applied to genres that adopt a progressive narrative structure with emergent components like open world video games. In this paper, we present the adaptations we will make to SPHINX to allow it to generate narrative puzzles in open world games that can act as optional side content. We argue that this can be achieved by creating a large database of puzzle items, actions, rules and goals; by altering when SPHINX generates puzzles; and by creating new item properties that remedy potential design issues that may arise. Some design implications are considered and avenues for further work are put forward.

Keywords: Procedural content generation · Puzzles · Open world · Video games · Interactive narrative · Authoring tools

1 Introduction

Story Puzzle Heuristics for Interactive Narrative eXperiences (SPHINX) was developed by de Kegel and Haahr [7] as a system for generating narrative puzzles in adventure games and as an alternative to other narrative puzzle generators such as the Puzzle Dice system [9]. It was then implemented in the short 2D point-and-click adventure game *Honey, I'm Home* by Morgan and Haahr [12] where players progressed through a linear sequence of areas by completing the narrative puzzle associated with each area. While this iteration of SPHINX increases the replayability of *Honey, I'm Home*, it does not allow players to complete narrative puzzles in an emergent way, as is often the case in popular open-world video games like *Elden Ring*, *The Legend of Zelda: Tears of the Kingdom*, and *The Witcher 3: Wild Hunt* where players can choose whether to complete puzzles and choose the order in which these puzzles are completed. We aim to build upon this first iteration of SPHINX, from here referred to as 'SPHINX 1' such that not only can it be applied to open world games but that it will allow for the presentation of emergent side narrative content in narrative progression games too [11].

L. Holloway-Attaway and J. T. Murray (Eds.): ICIDS 2023, LNCS 14384, pp. 146–154, 2023.
https://doi.org/10.1007/978-3-031-47658-7_13

2 Related Work

Work on narrative puzzle generation and, more broadly, quest generation describes novel systems that fulfil a similar purpose to SPHINX. For instance, the Puzzle-Dice system [9] adopts a similar approach to SPHINX in that it requires designers to create a database of items but only has a limited set of relationships that can exist between items. Work on using genetic algorithms and automated planning for quest generation [7] takes an interesting approach of assigning 'narrative tension' values to particular puzzle actions and uses these values to map generated puzzles to a three-act structure. The three-act structure begins with a tension of 0 that gradually increase until the resolution of the quest, where it returns quickly to 0. However, quests and narrative puzzles can often be quite short, making it difficult to evaluate how effectively they increase and decrease tension.

Ammanabrolu and colleagues [2] use Markov chains and neural language models to generate quests in text based adventure games; in their first generator, the probabilities of puzzle items following each other in a puzzle sequence are mapped on a Markov chain while in the second generator uses a neural language to generate a quest title, components and instructions. Regarding their first generator, the onus is on designers to decide how to weigh the probabilities between puzzle items using only domain knowledge which can be imprecise and difficult to tune. In the case of their second generator, the neural language model does not output puzzles in a format that could be used to instantiate puzzles in a non-text based game. A prototype generator developed by Doran and Parberry [8] has similar issues around compatibility with games that use 2D or 3D graphics. More recently, work has been conducted on incorporating generated pre-trained transformer (GPT) models into quest generation with authors stating that quests could often lack coherence [1] and varied widely in quality [17]. While both of the cited examples note that future GPT models may yield better results, limits user input as developers do not have the ability to alter the language model to better fit their project.

3 SPHINX 1

The SPHINX 1 [7] framework centers on an algorithm based on an extended context-free grammar that generates narrative puzzles from three main elements: items, rules, and areas. Items are game objects with specific types and properties, rules dictate how items and actions interact to result in a particular outcome, and areas represent individual sections of the game that each have an associated goal that must be achieved to reach the next area. Items can have string and bool properties that describe the function or purpose of an item, a descriptive string, a prefab and the option to set a specific spawn point. Puzzles are generated sequentially per area, working backwards from the area's associated goal to connect a chain of rules into a puzzle 'tree'. Rules are formatted using the following structure, starting from the goal on the left (Fig. 1):

$$itemType[properties_{0\ldots n}]_{1\ldots n} ::= action\ itemType[properties_{0\ldots n}]_{1\ldots n} \quad (1)$$

4 SPHINX 2

To allow for narrative puzzles to be presented to players in a dynamic and non-linear fashion in an open world game, we propose several key changes that we will make to SPHINX 1. SPHINX 1 was created by De Kegel and Haahr [7] in Unity and our update to this system will continue to use Unity (Fig. 2).

4.1 Puzzle Database

We aim to manually create a large database of items and rules that can map the relationships between many items and actions that are common across a wide array of open world games. Morgan and Haahr [12] implemented SPHINX 1 in a the adventure game *Honey, I'm Home*. A database of 60 items, 87 rules, and 4 areas (each of which have at least one pre-assigned goal) was created specifically for this game. This database is not only small but also has limited to no utility for use in other games. Should a designer wish to use SPHINX in their game, they would need to create the entire database of rules and items themselves, significantly increasing development time and costs. It is for this reason that we intend to include a large and comprehensive database of items, actions, goals and rules based on items, actions and goals that are commonly found in open world games in a variety of genres. Doing so creates a reusable library that designers can pull from and edit as they see fit. Naturally, games will still require bespoke items and rules that can only be used in a particular game but having a large database to work from affords designers the opportunity to spend more time on these items and rules without having to create ones that appear in many games already. This approach balances a focus on user input with the time and cost saving affordances of the database. For example, items in the base SPHINX database could include Box and Cat which could then be used in the rule:

$$Box[contains : Cat] :: = PutIn\ Cat\ Box \tag{2}$$

SPHINX 1 generates puzzles on a per-area basis where a form of backwards substitution creates a puzzle tree from a/the goal associated with an area. Each tree typically has at least two potential paths to the goal that the generator randomly picks from. By creating a limited but bespoke database of puzzle elements for a specific area, the onus is on the designers to essentially create a space of possible puzzles from which the generator chooses a canonical solution randomly for the current playthrough. SPHINX 1 includes the ability to increase the space of possible puzzle solutions by adding more items and rules, but doing so on a per-game basis is both costly and time consuming for designers. Not only will having a large reusable library of items and rules reduce workload and cost, but it will also drastically increase the number and complexity of puzzles that SPHINX 2 can generate.

The purpose of SPHINX 2 in an open world game is not to increase replayability as is the case with *Honey, I'm Home*, but to continually add more optional, emergent side narrative content for the player to engage with in a persistent

world and to encourage exploration. In *Honey, I'm Home*, the world is essentially 'reset' at the beginning of each playthrough; items, characters and scenarios can be added and removed without impacting the narrative coherence of the game world. The advantage of each goal/area having a single canonical solution per playthrough in this case is that it is much easier to vary solutions for each playthrough without using up too many items in the database and ensuring that the same solution is less likely to be used in subsequent playthroughs. In contrast, using SPHINX in an open world game with a persistent world requires that many items, characters and locations remain consistent throughout a single playthrough to maintain narrative coherence. We do intend to include the provision for puzzle elements that are not constantly present in the world by allowing designers to associate a plot contrivance with an item in the form of dialogue or a description. However, doing this for every potential puzzle element, as would be required in SPHINX 1, reduces the credibility of the world and can hinder player immersion and should therefore be employed as sparingly as possible. Drastically increasing the size of the database alongside having puzzle elements present in the world be consistent throughout a playthrough allows the generator to create multiple canonical solutions to each puzzle, affording players a greater level of autonomy in how they approach solving them.

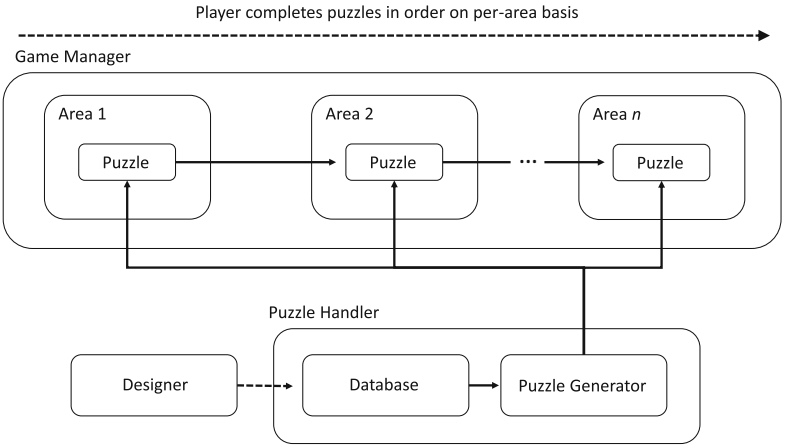

Fig. 1. Architecture of SPHINX 1 puzzle generation. The arrows point in the direction of the output from each component in the system. Solid arrows represent the direction of output within the system while dashed arrows represent input from outside the system.

4.2 Puzzle Generation

Secondly, we want to change the way in which SPHINX generates and instantiates puzzles such that it is no longer reliant on generating puzzles on a per-area

basis. In *Honey, I'm Home*, progression to the next area (and by extension, progression through the narrative) is contingent on solving the current area's puzzle. In open world games, players can often explore most or all of the world from the beginning of the game with few barriers to progression. Instead of generating puzzles per area like SPHINX 1, we propose that SPHINX 2 will generate puzzles at run-time at set initiation points placed around the map by the designers. There will be a maximum number of active puzzles at one time, and they will be instantiated outside a set radius from the player character's location. Limiting the number of active puzzles prevents players from becoming overwhelmed and prevents the generator from endlessly generating puzzles while the player is engaging in another part of the game. Similarly, instantiating puzzles outside a particular radius from the player character, ideally outside of the player/camera's line of sight, allows necessary non-player characters (NPCs) or items to be instantiated at initiation points without players noticing. This prevents ludonarrative dissonance and helps maintain players' immersion in the game. The exact number of active puzzles and distance from the player at which they will be instantiated can vary based on the game but we aim to find a guideline range through user testing.

To allow designers to control for intricacy, it will be possible to restrict SPHINX 2 to generating puzzles where there is a minimum number of degrees of separation from the goal, e.g., it finds the shortest possible chain of items or actions needed to reach a goal (based on what is in the player's inventory) and also ensures that at least one more chain of items or actions can be used to reach a goal, giving players some agency in how they solve the puzzles. In open world games, it is possible for players to explore the world at their leisure and while this grants a great deal of agency, it can also hinder satisfying puzzle solving. Players who have explored extensively and collected a wide variety of items in their inventory are likely to have most of, if not all components of a generated puzzle already in their inventory. If this is the case, solving the puzzle would just require manipulating the relevant items in the player's inventory. To combat this and to make exploration and traversal necessary for puzzle solving, we propose incorporating a system into SPHINX 2 that checks what items are in the player's inventory and only generates puzzles that require a minimum number of items that the player does not already have.

4.3 New Properties

We will add the *singleton* bool property as a special case property to denote bespoke items that can only have a single instance at any one time. This provision also makes it easier for designers to include 'key items' or unique items that are relevant to the narrative of a specific game. Items can have a 'Long Description' associated with them in SPHINX 1 that adds a level of optional detail, should players desire it. In addition to this we will also allow for this description to act as dialogue for NPCs, should it be necessary for players to obtain a particular item from an NPC. It will also be possible to associate NPC dialogue with goals and initiation points.

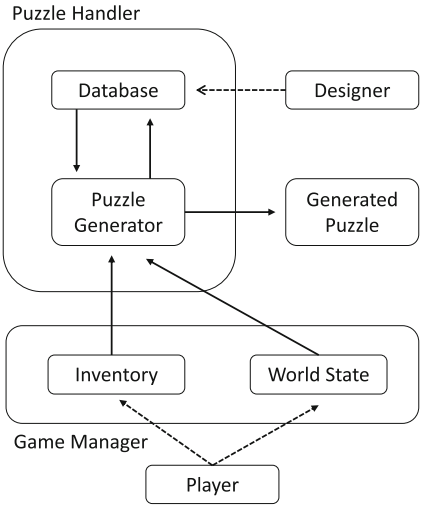

Fig. 2. Architecture of SPHINX 2 puzzle generation. The arrows point in the direction of the output from each component in the system. Solid arrows represent the direction of output within the system while dashed arrows represent input from outside the system.

4.4 Summary

In summary, the sequence of events in SPHINX 2 will be as follows:

1. The designer places puzzle initiation points throughout the game world.
2. The designer provides SPHINX with a list of rewards that can help players complete the main goal of the game.
3. The designer chooses what items and rules from the SPHINX database to include in their game and which to exclude, and adds any bespoke items or rules that can only be used in their game.
4. SPHINX, at runtime, checks what items are in the player's inventory and the current world state. The world state includes the number of active puzzles and initiation points within a pre-determined radius around the player.
5. If the conditions necessary to generate a puzzle are met, SPHINX picks a goal from its database, calculates a sequence of rules to reach a this goal and instantiates the puzzle at an initiation point.
6. The player follows the chain of rules to reach the goal and is rewarded.

5 Design Considerations

In addition to altering how SPHINX operates to allow for puzzle generation in open world games, the games themselves need to be designed with SPHINX in mind. One of the biggest hindrances to satisfying narrative puzzles in open world games is the amount of information navigation aids provide to players. For

instance, if the components of a generated puzzle are located across the world, using maps or location markers that provide too much information make solving the puzzle trivial. If players are given the exact location of the item they need on a map, then they need only travel to it rather than intuit from the design of the open world where the item could be. For this reason, careful world design is important should a developer wish to implement SPHINX in their open world game. To ensure puzzles have narrative coherence and that their solutions can be intuited by players, the design of the world should lean on players' heuristic knowledge e.g., if a player needs to a loaf of bread as part of a puzzle, they should be able to buy one in a bakery; if they need a bucket, there should be one beside a well. Designing the world in such a way allows players to make educated guesses when solving puzzles and removes the need to provide too much information through navigation aids.

Once a system to procedurally generate narrative puzzles has been created, the challenge of making players want to complete these puzzles remains. Players must have significant intrinsic (where one is motivated by the satisfaction of completing a task) or extrinsic (where one is motivated by factors outside the task) motivation [15] to engage with puzzles that are ultimately optional. Andersen and colleagues [3] describe how secondary game objectives, like optional narrative puzzles, can harm players' experience of a game when these secondary objectives are not consistently useful in pursuing the main goal of the game. They also argue that 'secondary objectives that do not support the main goal require extensive testing to avoid negative consequences', significantly increasing workload and development costs. Therefore, rewarding players for completing optional narrative puzzles with game elements that facilitate them in completing the main goal of the game is very important should SPHINX 2 be implemented in an open world game with a progressive narrative structure. Rewards such as experience points for strengthening player characters, useful items like health potions, rare weapons or money, or optional world-building narrative content could all be used to encourage players to complete generated narrative puzzles.

It is also worth noting that meta-game rewards like achievements and trophies that do not serve the main goal of a game (such as those on Valve's Steam digital storefront and on Sony's Playstation and Microsoft's Xbox account services) have been shown to not only act as extrinsic rewards but to also act as intrinsic rewards that boost self-esteem and players' social status both online and offline [6]. In terms of applying this to generated narrative puzzles, it could be possible create achievements/trophies that can be earned by completing certain numbers of narrative puzzles in a single playthrough.

6 Conclusion and Further Work

The sections above have outlined the initial steps we will take to adapt SPHINX 1 for use in open world games with a progressive narrative structure. However, several user experience challenges remain when implementing the proposed SPHINX 2 in a game; namely players' interest in completing puzzles and the

limited potential to create systems designed to aid players in completing these puzzles.

To address these user experience challenges, we intend to develop a gameplay demo of an open world game with SPHINX that incorporates the updates outlined above. Currently, we aim to create a simple world with a fantasy setting. Using this demo, we will conduct a series of user tests using player experience metrics like the *Game Experience Questionnaire* [10] or the *Ubisoft Player Experience Questionnaire* [4] to iteratively improve upon SPHINX in its new form.

Many games with human-authored puzzles have a wealth of online resources like walkthroughs and strategy guides that offer 'character information, back story, ...screen shots, ...hints and possibly "cheat codes" to help players advance quickly' [5] and solve difficult puzzles. Given the procedural nature of the puzzles SPHINX generates, creating walkthroughs of specific puzzles will not be possible. Therefore creating a 'narrative strategy guide' that compiles a game's SPHINX rules in plain language, reflecting the emergent nature of SPHINX puzzles [11], could be useful.

After the user experience challenges outlined above have been addressed, future work on SPHINX 2 could involve using player modelling data from scales such as the *Gamification User Types HEXAD Scale* [16] and *BrainHex* [13] to align narrative puzzles with player motivations. Similarly, SPHINX 2 could also be adapted to work in open worlds with procedurally generated terrain [14], similar to games like *Minecraft* and *No-Man's Sky*. Finally, we also aim to create an open-source SPHINX 2 Unity package that game developers can implement in their own games.

Acknowledgments. This work was conducted with the financial support of the Science Foundation Ireland (SFI) Centre for Research Training in Digitally-Enhanced Reality (d-real) under Grant No. 18/CRT/6224.

References

1. Al-Nassar, S., Schaap, A., Zwart, M.V.D., Preuss, M., Gómez-Maureira, M.A.: QuestVille: procedural quest generation using NLP models. In: Proceedings of the 18th International Conference on the Foundations of Digital Games, pp. 1–4 (2023)
2. Ammanabrolu, P., Broniec, W., Mueller, A., Paul, J., Riedl, M.O.: Toward automated quest generation in text-adventure games. arXiv preprint arXiv:1909.06283 (2019)
3. Andersen, E., Liu, Y.E., Snider, R., Szeto, R., Cooper, S., Popović, Z.: On the harmfulness of secondary game objectives. In: Proceedings of the 6th International Conference on Foundations of Digital Games, pp. 30–37 (2011)
4. Azadvar, A., Canossa, A.: UPEQ: ubisoft perceived experience questionnaire: a self-determination evaluation tool for video games. In: Proceedings of the 13th International Conference on the Foundations of Digital Games, pp. 1–7 (2018)
5. Consalvo, M.: Zelda 64 and video game fans: a walkthrough of games, intertextuality, and narrative. Tele. New Media **4**(3), 321–334 (2003)
6. Cruz, C., Hanus, M.D., Fox, J.: The need to achieve: players' perceptions and uses of extrinsic meta-game reward systems for video game consoles. Comput. Hum. Behav. **71**, 516–524 (2017)

7. De Kegel, B., Haahr, M.: Towards procedural generation of narrative puzzles for adventure games. In: Cardona-Rivera, R.E., Sullivan, A., Young, R.M. (eds.) ICIDS 2019. LNCS, vol. 11869, pp. 241–249. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-33894-7_25
8. Doran, J., Parberry, I.: A prototype quest generator based on a structural analysis of quests from four MMORPGS. In: Proceedings of the 2nd International Workshop on Procedural Content Generation in Games, pp. 1–8 (2011)
9. Fernández-Vara, C., Thomson, A.: Procedural generation of narrative puzzles in adventure games: the puzzle-dice system. In: Proceedings of the The Third Workshop on Procedural Content Generation in Games, pp. 1–6 (2012)
10. IJsselsteijn, W.A., De Kort, Y.A., Poels, K.: The game experience questionnaire (2013)
11. Juul, J.: Half-Real: Video Games Between Real Rules and Fictional Worlds. MIT press (2011)
12. Morgan, L., Haahr, M.: Honey, I'm Home: an adventure game with procedurally generated narrative puzzles. In: Bosser, A.-G., Millard, D.E., Hargood, C. (eds.) ICIDS 2020. LNCS, vol. 12497, pp. 335–338. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-62516-0_30
13. Nacke, L.E., Bateman, C., Mandryk, R.L.: BrainHex: a neurobiological gamer typology survey. Entertainment Comput. **5**(1), 55–62 (2014)
14. Rose, T.J., Bakaoukas, A.G.: Algorithms and approaches for procedural terrain generation-a brief review of current techniques. In: 2016 8th International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES), pp. 1–2. IEEE (2016)
15. Ryan, R.M., Deci, E.L.: Intrinsic and extrinsic motivations: classic definitions and new directions. Contemp. Educ. Psychol. **25**(1), 54–67 (2000)
16. Tondello, G.F., Wehbe, R.R., Diamond, L., Busch, M., Marczewski, A., Nacke, L.E.: The gamification user types Hexad scale. In: Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play, pp. 229–243 (2016)
17. Värtinen, S., Hämäläinen, P., Guckelsberger, C.: Generating role-playing game quests with GPT language models. IEEE Trans. Games, 1–12 (2022)