Dungeons & Maybe Dragons: a Computationally Expressive Board Game

Exploring the Effects of Computational Components in Traditional Gaming

Jean-Luc Portelli
Institute of Digital Games
University of Malta
jean-luc@portellimalta.net

ABSTRACT

This report outlines the design process of an experimental, procedural, computational controller for a board game, as well as analyzing the resultant effects and considerations of such a controller that replaces a traditional technique for board game design. The controller utilizes simple procedural content generation techniques to fill the role of map, scenario and event generation in a board game, which would otherwise be carried out through randomization with the use of dice, cards or other pre-designed instructions. With this approach towards board game design, various new possibilities emerge; of collaboration, content generation, customization, as well as data mining related to the board game's playthroughs. Some preliminary tests were also conducted into investigating the effects of such a game on the players, and whether it is perceived as beneficial or disruptive to the overall gaming experience.

General Terms

Game design, board games, tabletop games, procedural content generation, computation

Keywords

Game design, artificial intelligence, AI-based game design, design methodologies, tabletop games, expressive AI, experimental prototypes

1. INTRODUCTION

Many tabletop games, particularly those which are cooperative in nature, utilize certain techniques for content generation during gameplay. This is often done to create unpredictability and replayability in the game.

Highly complex methods for tackling this content generation can create a lot of downtime in a game, which is disruptive to the player experience. Hence, these methods commonly make use of premade cards that dictate the actions the game takes [7], dice for determining possible outcomes [1] and sometimes even having predesigned scenarios that affect the way the game is played [3]. These methods largely revolve around randomization and have functioned well enough for the media to flourish; however, they may restrict the potential for further development and innovation.

Dungeons & maybe Dragons is an experimental, cooperative board game designed for use with a dedicated software application, which generates events and outcomes within the game. The functional prototype utilizes simple methodologies of procedural content generation (PCG) that mimic the use of cards and dice. This is intended to explore the potential and benefits of such a system to board games at large, while also investigating the resultant effects on players who may utilize them. Through the use of a dedicated app, many parts of the game may become more streamlined, while also creating potential for many new features that are also discussed in this report.

2. KNOWLEDGE DOMAIN

For the purposes of this experiment, we tackled the subject of board games and the methods used for in-game content generation. This subject is particularly interesting due to how it is achieved through very traditional, straightforward means.

Usually, when this kind of mechanic is used, players are tasked with determining outcomes of events by drawing cards or rolling dice. Then, based on the rules, they interpret those outcomes and carry out its effects on the game. This can be as simple as moving a small number of tokens around, or as complex as changing the entire layout of the board. Whatever the case, these outcomes are often decided through random chance; shuffled decks, rolled dice or otherwise. The way the game acts is largely independent of players' actions and decisions. The game cannot react directly to what a player is doing, simply because it does not keep track of player's actions—unless designed specifically to do so.

Arkham Horror [7] is one such game that behaves through procedural generation. However, the order of events is almost entirely determined from the beginning—the cards are shuffled into a specific order and do not change position over the course of the game based on the players' successes or failures. It is entirely possible, and quite common, for the same game using the same components to deviate dramatically in difficulty from one play session to another, simply due to the way the cards are shuffled at the start.

It is important to note that there do exist board games that, through sophisticated mechanics, seem to react or change based on the players' behaviour. Power Grid [4] has one such mechanic, in which resources increase or decrease in value based on the demand; namely, as the cheaper resources are bought out, it only leaves behind the more expensive ones. While this serves as a good exception, it is also a very specialized one. Abstracting these computational mechanics into a form that can easily be represented by tokens for a board game is not a trivial process, and so it is rarely seen beyond simple feedback mechanics such as this.

Extracting these processes from a board game and including them in a dedicated, simplified application can provide a lot of content to players without overwhelming them. The complexity is abstracted to a simple series of button presses, as opposed to complex calculations. This methodology has been experimented with by board game manufacturers in the past, with varying degrees of success. One successful example of this hybrid game is Dark Tower [2], where the players competed with one another by interacting with a large pad of buttons that would provide them with feedback for their actions. It is this aspect of transmediality that we explored with our experiment, investigating its value to the platform.

3. DESIGN PROCESS

Our design process revolved mainly around two primary phases: game design and software design. Very early on, it was decided that the experiment would benefit if an entirely new game was designed for its purposes, as opposed to modifying an existing game. Both approaches would have been entirely relevant, but although the latter approach may have been quicker, building a whole new game would provide more flexibility for adjustments to be made based on the needs of the software, should any changes be necessary. Furthermore, starting with an entirely new design would provide a broader potential scope for future development, should it become a possibility.

3.1 Brainstorming

The design of the board game itself was greatly influenced by the concept of Dungeons & Dragons; namely, tabletop role-playing games. These games represented the pinnacle of procedurality, involving one dedicated player as the Dungeon Master to drive the game forward. Based on this, the use of quests and dungeon crawling were instrumental to the creation of the concept we would adopt: a dungeon crawling board game.

Other concepts were considered, but none provided a comparable amount of potential for design elements we could use for the procedural part of the game, while maintaining a good level of simplicity.

3.2 Board Game Design & Prototyping

Before the controller app could be designed, it was essential for the board game to be fairly complete. Most importantly was the need for establishing what parameters would be controlled by the app, and to what extent. Essentially, what we needed was to design a game with a traditional procedural generation system, and adapt it to a computational framework. In this way, our design considerations greatly revolved around mechanics, and little else, as it was what our controller would be entirely based upon.

Firstly, we considered the board generation. Many games use tile-based systems to generate maps, using a stack of modular map sections that connect together. Both Betrayal at House on the Hill [3] and Escape: the Curse of the Temple [8] use this mechanic to great effect, resulting in different map layouts from one play session to the next. After experimenting with various tile sizes and shapes, we settled on the same system that Escape uses; square rooms with different configurations of entrances and exits that force the topology to shift in specific ways. This also simplified the mechanics of movement.

Following that, we decided on a small set of actions that players can carry out: Movement, Opening Doors, Combat and Looting. This provided a good balance: a broad enough spectrum to allow for variety in gameplay, while also avoiding an overwhelming amount of rules to memorize. Each mechanic was tackled individually to ensure that they were balanced and effective, however, it was at this point that we began to also consider dynamics of play. Although they were easier to learn, certain simplified design choices were abandoned during playtests in favour of more sophisticated ones, to make the gameplay more interesting to the player.

Most of the work following completion of the system involved designing content. Each item, in terms of weapons, powers, enemies or traps, related specifically to the mechanics designed within the game.

3.3 Controller App Design & Prototyping

Once the major mechanics had been fleshed out and polished, the controller could be designed. Given the combination of the actions available and how they would interact with the game, the app had many mechanics available for use. Normally, here is where the design of the board game would cater for methods of generation. One of the major features covered by the app is that of quest and scenario management. We implemented a quest system similar to other games where players are given a scenario that describes their goals and losing conditions [3, 7]. In this case, however, the game does not provide one fixed set of criteria, but which change based on the scenario chosen. Additional rules or elements are added or changed as needed, but the underlying mechanics remain the same.

The challenge faced with this method was, similarly to the design decisions taken for the game itself, to balance complexity with functionality and gameplay. The app would need to be as transparent to the overall gameplay experience as possible, while gathering all the information necessary for it to function properly within the remit of the game. One of the first decisions, then, was that the interface was to be kept to the bare minimum of navigation requirements; every essential task could always be completed with a single button press.

Encouraging players to use the app while making it as straightforward a process as possible had two main benefits for the purpose of the design of the game. Firstly, every interaction that the players had with the controller fed data into it. This data could be used to track progress towards completion of the quests and for any other calculations that would otherwise be hidden from view. Secondly, the app could be used to feed information back to the players. This would give a stronger response to each button press, providing immediate value to the players for the app's use.

Tracking the button presses was simple, but generating content for the players to use in the game was more challenging. This was dealt with by telling the player the actual outcome of their attempt at a given action. For instance, succeeding at killing a monster wouldn't always only result in its death; items can drop, weapons can break, injuries can be sustained—the app provided an extra layer of uncertainty at every step of the game's progress, all without unnecessarily cluttering the experience in the process. The task of deciding event outcomes is completely removed from the player's hands; something which would be difficult to do with traditional board game components.

4. TECHNOLOGIES USED

Within the app's architecture, two main computational expression techniques were used: PCG and quest flags. Both were initially implemented very similarly to how they would be used in a traditional board game, but then developed further in directions that would not have otherwise been possible.

In the case of PCG, we used randomization quite broadly. Numbers were generated within specific ranges to provide different outcomes, of varying types and severity, to each action. These ranges for triggering positive or negative events could be varied to affect the difficulty of the game, which immediately led to another feature of the game; namely, that the game's difficulty could be adjusted easily to suit the players better. The numbers used were also different to what would normally be seen in a traditional board game. Dice only provide a limited range of values, while computer generated numbers can provide nearly infinitesimally small variations from one value to the next. During development, this potential for extremely granular adjustment would later give rise to the main feature of the PCG algorithm that differentiates the controller from regular board games.

Quest Flags provided a more involved development process. Initially, quest flags were targeted as a useful technology for this project because, quite simply, board games operate on this exact principle: satisfy some criteria and win the game. However, during development it turned out that they would be less useful for determining whether or not players had completed the game than they would be for advancing the game's state internally [9, pp. 47–51].

To illustrate this with an example, we can consider one of the two quests included with the prototype; the *Holy Grail* quest. Here, the game is programmed to wait until the players have explored a set number of rooms before presenting them with the treasure room they must find. This required feature was evidently a quest flag, however it was not considered to be one which the players needed to know at the start of the game or even at all, for that matter. Ironically, the quest flag that the players are concerned with; namely, carrying the grail to the exit room, is the one that the app was not designed to track.

This left us with a discrepancy in the design of the app: most of the quest flags that the players aimed for could not be automatically measured by the software. This led to a shift in focus. In its final iteration, the app does not concern itself with the outcome of the game. As is the case with traditional board games, the game does not acknowledge whether or not you have won the game. This re-prioritization led to more interesting considerations for how quests were designed. The actual winning and losing conditions were no longer bounded by the functionality of the application, but could be influenced by it along the way.

In this regard, another feature emerged; since the quests' progress was influenced by the quest flags stored within the app, the amount of actions needed could be tweaked to also affect the duration of the game. This feature, combined with the ability to adjust the difficulty of the game, allowed us to present a player with the options to tailor their experiences exactly to their needs and tastes; something which is sorely lacking in board game design nowadays.

Given the way that the players' *interactions* with the application were designed, it became very easy for us to measure how well a game was progressing. Essentially, by having players indicate when they have ended their turn, the app can measure their success rate. This provides very valuable information for measuring the difficulty of the game, but more useful to the players during gameplay is how the difficulty can be adjusted immediately. Randomization functions in board games rely on fixed ranges and probabilities, which rarely change, to determine outcomes. The app controller can adjust the probability of outcomes based on how much success the players are facing. With this variable difficulty, the app as PCG may be considered a combination of tool, material and designer for the game itself [6].

When combined, the properties of PCG and quest flags created an interesting overlap with one another. Within the algorithm, any content provided by quest flags would always have to take precedence over any generated content. While the flags represented the main progression of the scenario, the PCG worked to fill in the gaps leading to those main events. The two technologies worked well in tandem, but their uses within the overall algorithm remained mutually exclusive.

5. EXPERIMENTS AND EFFECTS

The final version of the game was tested with a small group in two separate sessions. For the first session, an approximately similar method was devised for generating events using rules and dice. In the second, the app was used as intended, through normal play.

During discussions, players stated their preference for the automated controller. Its interactions were not too invasive nor demanding, and provided benefits to the game's experience. Aside from feedback related to the game, the players were receptive towards the app's concept and explained that it allowed the game to flow a lot more smoothly by reducing the amount of downtime between turns. They appreciated not having to repeatedly stop and refer to a set of rules during play.

This topic was also dealt with in further detail with reference to other games such as Power Grid [4] and Arkham Horror [7]. In the former case, the players suggested that the task of maintaining the resource market with small, fiddly tokens could be delegated to a similar app, which would also allow players calculate the amounts they would pay for the resources they want to buy. In the latter, discussions revolved around automation of the mythos and event cards used to drive the story forward, as well as keeping track of the game's progress for players' reference, as opposed to using many separate cards that are harder to move around the table at one go. After the concept was explained to them, they were able to apply it to other games and suggest which parts could be simplified in a similar way.

Players were also asked about how many of these games' tasks they would automate. In the case of Arkham Horror, they explained that dice rolls should not be handled by the app as well, as it would take away from the game's overall experience without offering any value in return. This value was related to delegating the upkeep, in the form of repetitive tasks, that players must carry out in order for the game to progress.

They agreed that a balance would be necessary, as it would take away from the main experiences that they seek when playing tabletop games. This coincides with the overall concept of this project, and demonstrates its potential.

6. FUTURE DEVELOPMENTS

One of the major considerations throughout development was that of narrowing the scope of our development process. As soon as the designs for the basic functionality had been established, it was very easy to consider additional features to include, with the intention to further improve and extend the usefulness of the controller beyond being what may be construed as a "gimmick," or novelty.

One of the more obvious benefits and uses for this application would be of data mining. The app can easily collect information about the difficulties that are usually chosen, how many games are won and lost and how challenging the games prove to be as experienced by the players themselves. Given that board games are made up of simple, inert tokens and components, designers are entirely at the mercy of players recounting their experiences with the game without detailed heuristics for the use of analysis. Given a large enough player-base, this application would allow the game's balance to be adjusted over time based on the gathered data.

The next major benefit, with the game's ability to give back to the players, is the possibility of adding content on an ongoing basis. Essentially, it would allow designers to provide downloadable content for their board games, i.e. expansions in digital format. One of the major drawbacks of games that provide very detailed pre-written content is that it can become repetitive very quickly [3, 7]. This is also true of quiz and party games, which end up being shelved or sold off after the cards have been used enough times. Having a cost-effective, straightforward and accessible platform through which content can be delivered would benefit these sorts of games greatly, and allow a game to be monetized beyond its first purchase.

By extending this feature, digital content delivery could also act as a tool for crowd-sourced content creation. Given that the prototype is simple to modify and customize, the addition of a usable enough interface could easily allow players to design their own scenarios. This phenomenon of content creation can already be seen amongst board gaming hobbyists, who create their own expansions for existing board games. Digitizing the process would remove the need for printing or reproduction of physical materials and simplify the process immensely.

While the internal procedures of the controller app may be fairly simplistic at the surface, the value it can provide is far greater than the sum of its parts. Much of this potential, interestingly, comes as a result of its exposure to human interaction above all else; providing a creative inlet for players to express themselves in ways that they already do, but may lack the tools to do so.

7. OBSERVATIONS AND CONCLUSIONS

Overall, the process of creating Dungeons & maybe Dragons offered a different take on tabletop game design. The generative mechanics of these kinds of games can be considered incidental to the rest of the gameplay. On the other hand, they are a part of a game's system procedures that defines the fundamental differences between tabletop and digital games [5, pp. 67–68]. What this game does is bridge the gap between the two media of games and attempt to offer the best of both worlds; namely, the computational capacity of a digital game with the tactile experience of a tabletop game.

This game's focus brought a drastic change in priorities for its design and implementation. While the technologies and algorithms adopted would facilitate the game's design, its intended experience had to be built around the technologies themselves. Implementing a more complex board game would have likewise increased the complexity of the controller app's algorithms. Although the board game's design was completed first and foremost, this had to be done always while considering how these choices would eventually impact the features and feasibility of the computational component. With due diligence in its planning and development, the resultant game experience went beyond what would have been possible with traditional methods.

In conclusion, the symbiotic relationship between the physical and computational components, when handled appropriately, leads to a marked change in the game's design and player experience. As a result, players could focus exclusively on the parts of the game that concerned them, while completely disregarding the rules handled by the app. Ultimately, the relationship between the two created the potential for many more emergent features that were not planned, but became quickly evident over the course of the project. We can say that this proof of concept clearly demonstrates the potential of using computational elements in tabletop games, as well as the benefits that they provide.

8. REFERENCES

[1] A. Belmer. *Sanitarium.* Asmadi Games, 2012.

[2] R. Burten, A. Coleman, and V. A. A. J. Erato. *Dark Tower.* Milton Bradley, 1981.

[3] R. Daviau, B. Glassco, B. McQuillan, M. Selinker, and T. Woodruff. *Betrayal at House on the Hill.* Wizards of the Coast, 2004.

[4] F. Friese. *Power Grid*. Rio Grande Games, 2004.

[5] T. Fullerton. *Game design workshop: a playcentric approach to creating innovative games*. CRC Press, 2014.

[6] R. Khaled, M. J. Nelson, and P. Barr. Design metaphors for procedural content generation in games. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1509–1518. ACM, 2013.

[7] R. Launius and K. Wilson. *Arkham Horror*. Fantasy Flight Games, 2005.

[8] K. A. Østby. *Escape: The Curse of the Temple*. Queen Games, 1959.

[9] N. Wardrip-Fruin. *Expressive Processing: Digital fictions, computer games, and software studies*. MIT press, 2009.