Dungeons & (maybe) Dragons

Andrea Piano
University of Malta
andrea.piano.13@uom.edu.mt

ABSTRACT

As Game Studies become more and more structured thanks to influences by other branches of knowledge, many new experimental approaches emerge for designers. This paper will describe the AI-Based Game Design (AIGD) process behind the creation of a hybrid board-digital game entitled *Dungeons & (maybe) Dragons*, an experiment that tries to combine AI (and Procedural Generation) elements with a more "classic" board game structure. With regards to various detailed aspects of the process, this document will focus on an analysis of its three fundamental columns: Knowledge Domain, Technology Domain, and Game Design. The aim of the *Dungeon & (maybe) Dragons* project is to demonstrate that the application of computational elements to a pre-existing format can improve the design possibilities for game designers, and enhance specific game experiences for players.

General Terms

documentation, design, theory

Keywords

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

1. INTRODUCTION

Dungeons & (maybe) Dragons is a hybrid board-digital game created during the 2013-14 Computational Expression course at the Institute of Digital Games, Malta. It was created as a project based on the experimental approach of AI-Based Game Design (AIGD), intended as a method of iterative co-formation of AI (Artificial Intelligence) and game design [**1**]. The scope of this project was to demonstrate the possibility to enhance player experience in board game through the implementation of a AI dynamic controller.

The game consists, in fact, in a PCG board game played with the aid of a digital application, namely a controller. The players take the roles of a Warrior, a Mage, and a Rogue in the task of exploring a mysterious dungeon that could contain monsters, traps, treasure, and (eventually) a dangerous Dragon. The game allows various difficulty settings, related to desired duration of each session, and sensible to change due to alterations of the game internal state. *Dungeons & (maybe) Dragons* is designed to offer a different experience every time, strongly influenced by players actions, in combination with controller randomly generated content. Therefore, winning and losing conditions are chosen from a vast pool of possibilities, enhancing both replay ability and a feeling of uniqueness from game to game.

In this paper, I will address the theoretical background that led to design choices and outcomes, with particular regards to Knowledge and Technology Domains, and Game Design aspects used to create the game as it is. The aim of this analysis is to enlighten how AIGD can be approached, with its pros and cons, but also to show how the three main aspects of the process can interact with each other, contaminating mentioned choices and outcomes.

The analysis will begin from the pillars of Knowledge Domain, Technology Domain, and Game Design (in this order). It will tackle the very core of the design process, with specific reference to Computational Expression theories, and so-called "effects". The report will continue with a less analytical, and more descriptive overview of the design process; starting from scratch, through conceptualization and brainstorming, to prototyping, and evaluation methods. In the end, I will address some personal considerations about the whole process, with an eye on future application of AIGD methods and techniques.

2. KNOWLEDGE DOMAIN

The creation of *Dungeons & (maybe) Dragons*, was influenced by various Knowledge Domains: board games controllers, conventions, and procedural generation of content; dungeon crawlers; tabletop Role Playing Games (RPGs); and high-fantasy settings. It is interesting to note, especially with regards to board games controllers and conventions, how the Knowledge Domain influenced both Technology choices, and the whole Game Design process – being influenced back.

The main interest of the experiment was in fact to understand how players interact with controllers in board games, both in terms of procedural generation of content, and supervision of the game-state. RPGs implement this through the figure of the Dungeon Master (DM), which has the responsibility of controlling the progression of the game, always keeping balanced the situation between players, and gameplay elements. Many board games have specific sets of rules, in the forms of books, decks of cards, or other, that put all the players in condition to interpret the game state and modify it as play progresses. In digital games, controllers are a fundamental part of the system, and are needed to allow various elements to work properly. Hence, the idea of applying an AI to board games, creating a hybrid that could permit the coexistence of tabletop typical gameplay, implemented through a computational element.

In order to make it possible, it was necessary to apply existing elements of tabletop gaming to the project, crafting then a system capable to stand-alone without the computational element (so that the AI could actively add something different to the experience). It was taken into account a discrete number of board games, with a particular focus on *Betrayal at House on the Hill*, which implements a controller in the form of two specific rulebooks; *Escape*, which has an element of procedural generation in the board itself; and *Dungeons & Dragons* for both the setting inspiration, and controlled procedural generation of content through emergent play.

The Knowledge Domains of dungeon-crawling, fantasy setting, and TTRPGs are strongly related to aesthetical and dynamical specific design choices. We found that the specific dynamics such as those of treasure hunting in dark dungeons were perfect for our mechanics, and for the Technology Domain we chose to work with. Applying stereotypical design conventions from famous RPGs helped both in creating content for the game, and giving the players a feeling of “bridging” from classical tabletop human-controlled gameplay to a more AI-driven situation. With regards to this, it was taken into consideration an ironical appeal of the game in order to enhance the aesthetics of the game, making it viable for a vast pool of audience. The ironical and referential title *Dungeons & (maybe) Dragons* recalls this aspect, taking probably the best known TTRPG into account for a satirical revision.

3. TECHNOLOGY DOMAIN

Computational processes can be used to model, simulate, and implement everyday life behaviors [**2**], hence with regards to board games, they can substitute many functions of randomization and control. More importantly, operational logics can implement new structures into the systems, shaping them as digital-analogical media. This hybridization influences the fruition, leading to a more digital kind of interaction between the system and the players/users: it creates a loop between modification of surface and action in the processes by external actors, and data alteration by the system [**2**]. Hence, the constant feedback changes how players interact with the game, but also the game state itself.

Dungeons & (maybe) Dragons implements a few computational elements, such as Quest Flags [**2**], Procedural Content Generation (PCG), and a simple AI controller that monitors the status of the system as the game progresses. The combination of these factors creates a controller agent based on mutual trust between the players – as a given of many board games – helping them in interacting with the game through a constant updating of the system. Due to technical feasibility, its prototyped version is an executable for Windows OS / iOS, made within Game Maker which simulates an application for Android / iOS mobile devices.

The controller is then a form of simplified and digitalized Dungeon Master (DM), which reads the progression of the game by player-given input, and reacts altering the system according to it. In the beginning of the game it assigns a Quest – a set of objectives – to the players, and generates content as the users progress through the dungeon. The set of actions that it can read goes from slaying a monster to exploring a room, or looting a chest for treasure. Every time a user desires to perform those tasks, the controller alters the system, and this alteration leads to a user-driven co-created unique experience for each session.

Its pattern of interaction is strongly related to the Knowledge Domain of dungeon crawling, and also to fantasy based settings: it works exactly as a proper DM, just without the human face-to-face interaction. If it is arguable that this leads to a lack of variety, on the other hand it facilitates the process influencing the game pace. Moreover, the openness of the source code is an element that could enhance the functionality of the application, encouraging user-generated content to a virtually unlimited extent.

3.1 Quest Flags

It is undeniable that the Knowledge Domain of dungeon crawling, and – more generically – the fantasy approach of this project strongly influenced the choice of operational logics. Quest Flags [**2**] were the most natural elements to look for, right in the beginning of the process. Almost every Role Playing Game implements at least one of them, and their efficiency – given the right context – in the systems is so excellent to justify their great pervasiveness. This technology has, obviously, a huge limitation: to grant a working system with Quest Flags, designers have to take into consideration every possible state of the system itself, which requires a great amount of work, and resources too.

Rather than giving the players a false feeling of complexity within the system, in fact, Quest Flags can help achieving a good sense of *playability* in a story-driven game [**2**]. This applies, more broadly, to every game that implements multiple goals as final play outcomes, therefore board games include similar features in many cases. Quest Flags represent a method of fragmenting the outcome in little pieces, namely the “flags”, making the output from the game dependent to the system’s state. In *Dungeons & (maybe) Dragons*, Quest Flags were implemented to have, indeed, the possibility to influence the outcome not only as proper quests that give the players an objective to pursue, but also with regards to undesirable consequences, such as losing conditions.

The cooperative nature of game dynamics influenced this computational operator, adding a necessary layer of complexity to it in order to allow a decent gameplay pace and a balanced overall difficulty. Nevertheless, users will not be allowed to follow the gradual achievement of quests while the controller takes care of them; this creates a unique pattern of interaction with the board game, since usually objectives cannot be hid from the players. It allows Quest Flags to become a hybrid computational-authorial tool, rather than only a system state modifier.

3.2 Procedural Content Generation

During the last few years, both gaming industry and academia started to consider Procedural Content Generation (PCG) as a method to encourage repleyability, co-creative experiences for players, and crafting more expressive systems to interact with [**3**]. Inspired by digital and analog games such as *Escape!*, *Betrayal at House on the Hill*, *Diablo III*, *The Binding of Isaac*, and *Spelunky*, a PCG approach was taken since the earliest phases of design in order to utterly enhance the uncertainty of the outcomes, offering a valuable different experience to users every session.

In this project, PCG was implemented as a Content Selection method that could define and Construct [**3**] the experience, starting from the board, game by game. It works as a library of predefined elements that are selected through the implementation of specific algorithms during play, by players’ actions. Therefore, PCG in *Dungeons & (maybe) Dragons* could be considered a TOOL [**4**], a device designed for a specific purpose; on the other hand, and in a broader sense, it could be seen as a DESIGNER too: it participates, to some extent, in the authorial process of each play session alongside users, and original designers.

It was interesting to notice how PCG and Quest Flags can be both implemented in a design without the system to crumble. In fact, juxtaposing them could be seen as a contradiction, since procedurality does apparently not fit the necessary predefined amount of goals that Quest Flags need to successfully function. In this project, their cooperation is enhanced through alternation: when the PCG tool acts, it does not affect Quest Flags; vice-versa, the definition of Flags is not taken into account by the procedural algorithms. Thanks to this method, the two approaches appear to properly work without any incident.

4. GAME DESIGN

In the AIGD approach, Game Design is a process that defines game's mechanics, and dynamics contemporary to AI development, both being influenced by and influencing it [**1**]. It is indeed a fundamental part of the process, strongly connected to the two other main pillars of Knowledge, and Technology Domain. While the specific Game Design path followed to create *Dungeons & (maybe) Dragons* will be described in detail within a specific chapter of this paper, here are gathered some preliminary and theoretical considerations.

With regards to the discussed project, it is difficult to separate Knowledge Domain from Game Design adopted conventions. Board games rules, dynamics, and traditional elements permeate both, influencing decisions and shaping the prototype in a very specific direction. When considering the set of mechanics that structure the gameplay, then, we took inspiration from our knowledge and crafted an original system strongly rooted in many different tabletop games: *Betrayal at House on the Hill*, *Escape*, and *Dungeons & Dragons*.

Since the Game Design provides the context for the AI controller to properly function, the whole set of rules and mechanics of the game had to be polished well before the computational element could be put in the game. For this reason, *Dungeons & (maybe) Dragons* gameplay was designed in order to work even without a controller, despite the game clearly plays at its best with it. Mechanics were not *based* on the Technology, then: they were sometimes restructured to fit better with an AI controller, but could ideally stand on their own. This way, Game Design and Technology Domain influenced each other, but could be developed with sufficient independence to grant more fluidity, and flexibility, to the project.

5. DESIGN PROCESS

Dungeons & (maybe) Dragons was not created from scratch: due to time given and technical feasibility of the project, in the beginning it was a collage of mechanics, dynamics, and aesthetics from other existing board games. Then, the process slowly refined those elements, giving them a more original appeal as conceptualization led to actualization of the final prototype. In this chapter, I will explore the said process from the earliest phases to its conclusion, chronologically analyzing design choices and constrains as they emerged.

5.1 Brainstorming

The very first stage of conceptualization involved a brainstorming discussion with regards to the three Domains. Before developing any idea or concept about mechanics, dynamics, or aesthetics, the primary concern was to understand the possibilities of the tools at our disposal. Brainstorming was structured as a conversation in which every possible computational operator was enlisted, and eventually rejected in favor of more feasible or interesting technologies, subjects. This discussion enlightened the potential of many operators, with a particular interest in their connection with Knowledge Domains and Game Design conventions.

After bringing down many options, the focus shifted to Quest Flags, Dialogue and Behavior Trees, and inevitably Dungeon Crawling in procedurally generated fantasy settings. Then, at that point, it was then important to state a purpose for the game goals, set up proper rules and mechanics, narrowing every possible idea to a practical, small list of a few [**5**]. An important topic we engaged in this phase was game dynamics: our concern was to find an entertaining, but at the same time solid, set of mechanics that could enhance a cooperative and balanced multiplayer gameplay of variable duration (from 10' to 20' circa).

5.2 Prototyping

Previous experiences suggested to pay a specific attention to feasibility, considering both time given for the completion of this course-related project, and resources we could access. Due to these considerations, then, the step between brainstorming and prototyping of the project was a research of mechanics, enhanced dynamics, and pleasing aesthetics already implemented in existing games of our knowledge. *Betrayal at House on the Hill* was the main inspiration for control-driven mechanics, and it was taken into account as a game that could work properly with an analogic controller. *Escape* provided us some aesthetics in terms of tokens and tiles, but also fundamental PCG elements to include in our project. TTRPGs, especially *Dungeons & Dragons* strongly inspired the overall appeal of the game with regards to aesthetics, but also dice rolling mechanics, and DM-driven patterns of interaction between players.

As soon as the purpose of the project was stated, decided, and clarified, prototyping involved more and more technical considerations with regards to the stand-alone board game. Taking into account the experimental goals of AIGD approach, we started polishing what we took from other titles into something that could fit expectations. The application of game design frameworks such as the Mechanics, Dynamics, and Aesthetics approach [**6**] was significant for the success of the whole process: starting from the desired experience we decided to design for the player, the system was crafted towards a functioning set of rules that could enhance those defined dynamics of cooperation.

The final prototype was then implemented with a digital controller made in Game Maker, which allows various patterns of interaction with the game state by the players: it lets them choose a difficulty setting that determines both duration and complexity of the game, then generates a random Quest from an internal database, and accepts inputs such as "Explore", "Loot", "Kill", and "Update". These commands are interpret ate by the system as modifications of the internal state, while an algorithm takes in account the expected progression of the game and reacts increasing or decreasing difficulty.

Players can then decide to explore a new "room", roll the related dice, click on the "Explore" button, and receive from the system a randomly picked tile to expand the game board. The controller generates then an amount of monsters – namely, token NPCs – and traps, or treasures, in the said room. Here, according to game rules, they have the chance to engage the enemies in a combat, and eventually slain them; as this action is performed, the users can select "Kill" in the application, letting the system know that they won the fight. If the "Kill" button is being pressed too many times, there is a chance that in the next explored "room" there will be a higher amount of enemies; vice-versa, if the game pace is going too slow for the expected time given at the beginning, the controller adapts decreasing the threat levels.

5.3 Evaluation

Playtest sessions enlightened what was successful and what instead did not go as expected during the design process. A very first result was to notice that, from the player perspective, the game does not provide completely an innovative experience: almost every tester agreed that the computational component substitutes with good results the conventional, analog controls; but there was no other aspect to investigate from their perspective. The game creates, to some extent, a Tale-Spin Effect [**2**] in which the complexity of the system does not appear clearly to the players that are using it. If this could be seen as a failure of certain methods, on the other hand is a success in simplifying the game experience, obtaining a more fluid and flexible gameplay.

With regards to evaluation, though, it emerged that the experiment was more valuable for sparkling design discussions, enhancing confrontation between academics on the used methods and approaches. Among playtesters there were in fact some professional designers, and game enthusiasts, that were able to "look" into the system, understanding mechanics and appreciating better what was implemented. They observed how the controller reduces the workload for the players, enhancing a faster and fluid experience; it keeps content hidden, which eliminates the ability to spoil the game completely in the beginning; more importantly, the computational operator allows to add content to the game in a way that does not affect the system in terms of algorithm, rebalancing needs, and so on.

5.4 Possible Enhancements

Due to technical feasibility in the given time, and many other design necessities, the project was narrowed to a system that could demonstrate the desired outcome, but did not implement all the considered possibilities. One big issue was Game Maker: the tool was selected in order to allow cooperation between designers that had different backgrounds, but it casted a shadow of huge limitations on the game. Its object-oriented approach to programming did not allow to implement structures as remote difficulty adjustment, addition of content, or digital expansions that other languages could, instead, enhance. A possibility for the future is in fact to reprogram the controller with a more flexible and diffused tool such as JavaScript, which could permit better results.

The opportunities in that case could be vast, and diverse: from the implementation of Dialogue Trees [**2**] to enhance the role-playing aspect of the game, to the expansion of the participants to four or five instead of three, essentially adding more content to the system. With regards to this, also, the project is prone to a free publishing in both the board game market, so that the experiment could reach a larger pool of players; and an open-source diffusion through academia for exploring the interaction patterns of digital controllers in board games generaically.

6. CONCLUSION

AIGD revealed itself to be a really valuable approach, with a great potential. The combination of three aspects such as Knowledge Domain, Technology, and Game Design conventions or genres, represents a useful framework when considering computational elements and system-creation. The whole idea of crafting a system, rather than defining a default path for the user to traverse, and experience the game, is neat to understand better digital game design complexity, and prospective.

Definitely, the approach adopted for this specific project has room for an expansion both in terms of further academic exploration of the concept, and market opportunities. With regards to the first, though, *Dungeon & (maybe) Dragons* represents a good starting point for a better understanding of computational elements application to board and tabletop games that deserves a deeper study in future works.

7. REFERENCES

1 Eladhari, Mirjam P., Sullivan, Anne, Smith, Gillian, and McCoy, Josh. *AI-Based Game Design: Enabling New Playable Experiences*. University of California, Santa Cruz, 2011.

2 Wardrip-Fruin, Noah. *Expressive Processing. Digital Fictions, Computer Games, and Software Studies.* MIT Press, Cambridge, 2009.

3 Smith, Gillian. Understanding procedural content generation: a design-centric analysis of the role of PCG in games. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York 2014), ACM, 917-926.

4 Khaled, Rilla, Nelson, Mark J., and Barr, Pippin. Design metaphors for procedural content generation in games. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York 2013), ACM, 1509-1518.

5 Fullerton, Tracy. *Game Design Workshop. A Playcentric Approach to Creating Innovative Games*. CRC Press, Boca Raton, 2008.

6 Hunicke, Robin, LeBlanc, Marc, and Zubek, Robert. MDA: A Formal Approach to Game Design and Game Research. In *Proceedings of the Challenges in Game AI Workshop, Nineteenth National Conference on Artificial Intelligence* (2004).