CORE
Metadata, citation and similar papers at core.ac.uk
Provided by Repositório Comum

The challenge of Automatic Level Generation for platform videogames based on Stories and Quests

Fausto Mourato[1, 2], Fernando Birra[2] and Manuel Próspero dos Santos[2]

[1] Escola Superior de Tecnologia – Instituto Politécnico de Setúbal, Portugal
[2] CITI – Faculdade de Ciências e Tecnologia – Universidade Nova de Lisboa, Portugal
fausto.mourato@estsetubal.ips.pt, {fpb, ps}@fct.unl.pt

Abstract. In this article we bring the concepts of *narrativism* and *ludology* to automatic level generation for platform videogames. The initial motivation is to understand how this genre has been used as a storytelling medium. Based on a narrative theory of games, the differences among several titles have been identified. In addition, we propose a set of abstraction layers to describe the content of a quest-based story in the particular context of videogames. Regarding automatic level generation for platform videogames, we observed that the existing approaches are directed to lower abstraction concepts such as avatar movements without a particular context or meaning. This leads us to the challenge of automatically creating more contextualized levels rather than only a set of consistent and playable entertaining tasks. With that in mind, a set of higher level design patterns are presented and their potential usages are envisioned and discussed.

1 Introduction

The way how narratives and videogames are connected is still a topic that is open to debate. Nevertheless, it is undeniable that over the years several videogames have been used to tell stories. In fact, this topic started to catch attention in later years with the appearance of textual adventure videogames, some kind of *Interactive Fiction* [1]. The term *ludology* has been proposed by Frasca [2] as a movement based on *narratology* having different features with distinct merits. Therefore, the main question is not if games are effectively narratives or not but how they relate to narratives. An interesting recent approach on this question is a model presented by Aarseth [3]. Briefly, the author confirms videogames as an interactive fiction medium, whose elements can be characterized in a ludo-narrative dimension. The different elements can have a more meaningful (narrative) effect or a more challenging and amusing (ludic) importance.

In the context of automatic level generation, existing algorithms focus the lower abstraction concepts to provide adequate actions (Section 2), yet higher abstraction concepts must be considered to create more intricate scenarios. We present our studies in how to relate stories and story related concepts with automatic level generation to produce more meaningful content, which include: the definition of a set of abstraction layers in hero-centred stories (Section 3.1); a study on the *platformers'* storytelling capabilities (Section 3.2); the definition of higher level game design patterns for *platformers*

adfa, p. 1, 2011.
© Springer-Verlag Berlin Heidelberg 2011

(Section 3.3); and a final reasoning on ways to promote more contextualized automatic level generation in platform videogames applied to a prototype (Section 4).

2 Related Work

2.1 Procedural Content Generation (PCG) in platform videogames

A possible use of PCG in videogames is automatic level creation. For *platformers*, it may serve either full automatic processes that generate complete levels [4] or semi-automatic processes where some parts are humanly designed [5–7]. This idea was brought to the academic context by Compton and Mateas [8] with a conceptual model that inspired later studies, namely the framework proposed by Smith *et al.* [9] to analyse these videogames. Also, the notions of rhythm and rhythmic pattern were introduced as a "mechanism for grouping individual components into a longer sequence, while still maintaining rhythmic movement for the player", which lead to the implementation of a prototype for a rhythm based generator [10]. This approach has been implemented later in different systems, namely *Launchpad* [11], *Rathenn* [12] and *Endless Web* [13].

An alternative approach consists in the displacement of a set of humanly pre-authored chunks to create the whole level. An effective way to perform that displacement is a technique entitled Occupancy-Regulated Extension [14].

A more direct approach to generate this type of levels was proposed by Pedersen *et al.* [15], focused on *Infinite Mario Bros.*. The authors presented a generator based on an initial parameterization where several level features were defined, such as the number of gaps, their average frequency and amplitude, among several others.

Finally, we have proposed a genetic algorithm implementation [16] to generate more varied types of spaces, including rooms, corridors and mazes. It consists in the integration of a set of simple design patterns to make the algorithm evolve to a set of valid game geometries with different features. To improve content inside those levels, we have also proposed a level adaptation algorithm [17] that is able to fill the main geometry with additional gaming entities, such as enemies or traps. Lastly, an integrated system where those concepts are applied was also implemented [18].

2.2 Game/level design patterns

Game design patterns have been used to understand the rules behind the mind of game creators. This concept was introduced by Björk *et al.* [19] as commonly recurring parts of the design of a game regarding gameplay. Naturally, each genre has particular design patterns, which have been studied lately as this topic is becoming popular. For instance, First-Person Shooters have been examined by Hullet and Whitehead [20] and 'Ville Games such as *Farmville* has been studied by Lewis *et al.* [21].

In the context of platform videogames, besides the first overview that is provided in the previously referred studies by Compton and Mateas, the question of game design patterns was brought by Dahlskog and Togelius [22] in their recent work focusing *Infinite Mario Bros.*. However, the main focus goes once again to the lower abstraction layers, as the authors identify how multiple entities are combined in a sequence. As

lower level abstractions are very precise in describing a particular game, they are not likely to be generalized to other titles of the same genre.

An interesting alternative study but yet related to the topic of game design patterns was presented by Sorenson and Pasquier [23]. The authors defined a system based on a genetic algorithm implementation that creates levels by encoding design patterns into the fitness function. Some possible design patterns are given as an example for the videogame *Infinite Mario Bros.* to show how it could be applied.

An adventure *platformer* that allows exploration uses principles from an RPG, where higher abstractions patterns have been defined, and still use action based mechanics. To generate game scenarios that allow this combination, it is important to understand how those higher abstraction patterns can connect to the lower abstraction set of actions.

2.3 Stories and Quests

In order to bring higher abstraction aspects to level generation and provide meaning to the avatar actions representing somehow a story, within a certain context, it is important to understand the key concepts of plot representation. Initial requirements can be identified in Ryan's [24] proposal for the conditions of narrativity. The author refers spatial and temporal dimensions as the essential requirements for a broad concept of story. A mental dimension is defined as the next step to include the human experience in stories, where intelligence and emotions are described with agents, goals, actions and events. A final formal and pragmatic dimension expands the previous semantic concepts to significance, namely regarding the relation among events and their meaning.

Tosca [25] identified the notion of quest as "a way of structuring events in games, and (…) incarnate causality at two levels: a semantic one (how/why actions are connected); and a structural one (plan of actions, interaction of objects and events)". In fact, this concept is natural in videogames mainly because they tend to represent user incarnations on a certain character, the avatar. This hero centred approach that we find in videogames makes them suitable to be represented as quest-based games.

Sullivan *et al.* [26] created a quest browser recalling Howard's definition of quest as "a goal-oriented search for something of value" [27] and stated that quests "give meaning to the player's actions". In addition, authors gathered previous studies based on *World of Warcraft* to identify a set of quest types.

Another perspective to look at quests in videogames was proposed by Aarseth [28], based in a division of the concept into three main axis: place, time and objective.

A practical approach is presented in the playable quest-based story game *Mismanor* [29], where the authors applied their theories about playable stories. The main aspect to refer regards the proposed quest structure, which contains a certain intent and different sets of pre-conditions, starting and completion states.

Finally, a motivating study about quests was presented by Smith *et al.* [30]. Their work regards RPG's, namely the patterns used in that genre of games, which were divided into level patterns and quest patterns. The second type is particularly interesting in the scope of the present work because, as stated, adventure *platformers* can share several principles with RPG's. Authors grouped quest patterns into five main groups: actions, objectives, structure, superstructure and purpose.

In the remaining of this document we will explore how the presented different approaches to describe quests can be merged in the specific genre of platform videogames and how we envision these principles as a way to improve automatic level generation.

3 Platform Videogames

3.1 Quest-based stories (for videogames)

As previously stated, our first goal is to understand how platform videogames transmit stories. In this genre, the user controls a character or a set of characters (one at a time) in a certain environment with certain goals. If those goals have some meaning it has to be possible to consider them quests, or parts of quests. For this purpose, it is important to perceive game design patterns for quests in platform videogames. There are different approaches to this challenge, as seen previously. One first issue that might occur when establishing possible patterns and comparing proposals is that the comparison objects might be in different abstraction levels. Consequently, game design patterns must be analysed according to a certain abstraction level and, as we are focusing mainly the game design patterns that are somehow story related, it is important to emphasize that stories have different abstraction levels as well. We propose the following abstraction levels to decompose a character centred story in the context of game design patterns and automatic level generation:

- **Universe**. The higher abstraction is the vague description of the spatiotemporal context, meaning the type of world and physics, among others. It states **when** and **where** the story occurs. Examples: Nowadays, Tolkien's world and Medieval Times.
- **Main Goal**. Defines the ultimate goal and its main motivation. It explains **why** further events will happen. Example*: Find the Holy Grail*.
- **Chapters**. The main goal is naturally divided in sections, normally arranged in chapters in a book or scenes in a movie. In a long story, a grouping mechanism might be used, only for segmentation without any impact on the story itself. This will be the equivalent to dividing a same story into multiple books.
- **Quests**. They represent a non-trivial task with a certain purpose that must be part of the multiple steps to reach the main goal. By some means, this describes **how** the main character reaches the ultimate goal. Depending on how deep the story is, chapters and quests might be similar in abstraction level but they differ especially on the point of view. Example: *Retrieve the sword Excalibur*.
- **Side-quests.** A side-quest has similar principles to a regular quest but it does not have direct implications in the story, and typically it will have a shorter duration. It just endorses the character and his/her personality. Any random heroic action without any additional purpose serves as an example of side-quest.
- **Goals**. Represent the multiple steps to accomplish a quest, and is something that could be continuously expressed by the user meaning what he/she is doing at the moment. Example: Defeating an opponent or a series of opponents.

- **Actions**. As a rule of thumb, one can consider an action as something that can be performed by the avatar and that the player can assign to an input button. Actions do not mean anything in particular and can only be interpreted together with other actions in the context of a goal. Examples: walking, attacking, jumping or running.

Several approaches have been considered for automatic generation on the most concrete layer of the model. On the top layers the common efforts are focused on the narrative and are not suitable for an action game. So, the current goal for automatic level generation is to provide an appropriate bridge between the bottom and the top layers.

3.2 Platform videogames and stories

By intuition, *platformers* do not appear to contain an intense and intricate story. Probably, one of the main reasons of their success is that they are typical cases of *easy to play but hard to master* games, where instant action is presented. Normally they have a smaller set of actions and commands than other games and challenges are mainly related to immediate skill. However, content on these games must play an important role as well, as *Sonic* and *Mario* are two of the most popular videogame characters which surely succeeded in the test of time. Therefore, we want to understand how stories have been included in *platformers* and comprehend how different elements are being used regarding ludic and narrative goals. For that purpose, we collected a set of popular platform videogames. Ten titles were selected trying to include a large range of gameplay styles, hardware and themes. Franchises and multi-versioned titles should be considered a global overview over the multiple available versions where the same style is represented. Our list was defined with the following videogames (abbreviations are presented in parenthesis for further references): *Blues Brothers (BB); Braid (Br); The Cave (Ca)*; *Little Big Planet (LBP); Prince of Persia (PP); Rick Dangerous (RD); Sonic (So); Spelunky (Sp); Super Mario (SM); Trine (Tr).*

The videogames from the previous list were characterized in the already mentioned *ludo-narrative* model proposed by Aarseth [3]. The author identified world, objects, agents and events as the main storytelling elements in a videogame. Those elements can be more narrative if they are stricter to a certain description or more ludic, if their objective is mainly fun without particular relation to an initial narrative. Figure 1 presents that model, where we have positioned our previously referred example titles. The main descriptions proposed on Aarseth's model were used to establish an initial clustered classification. Afterward, similar titles were compared observing the features with more detail to obtain the represented rank. For instance, regarding world structure, both games *Trine* and *Braid* have linear paths but the first contains short bifurcations, hence their different positioning in the scale.

It is noticeable that the elements on those games have features that are more related to the narrative pole rather than the ludic. More than meaning that these games are elaborated stories, this shows that *platformers* tend to be an expression of predefined ideas. Story elements (kernels and satellites) are typically fixed, meaning that the character has an initial strict set of objectives that will lead the whole story development over the game. Hence, it is curious to wonder why *platformers* are not associated with

stories and others genres are. The first reason to explain the lack of story association to platform videogames can be found in the previous table regarding agents. *Platformers* tend to use agents merely as ludic elements. Still, some exceptions occur. *Little Big Planet* is one of those cases, mainly because the second title of the game included new characters that accompany the hero in the whole game, manifesting their own personality. *The Cave* also shows interesting dialogs and relations among the playable and the non-playable characters. In *Trine*, there is also a special situation regarding agents. The player controls three different heroes embodied in one single character that can take the physical form of one of them at any time. During the game, the hero's spirits inside the character talk to each other about the adventure that is taking place, exposing their personality. Even though they are not game agents in the common sense of the word, they work as so. Still, this leads us to another aspect that we believe that it is important to consider as a relevant element and that should be included in the previous model, which is the hero himself/herself.

	World		**Objects**		**Agents**		**Events**	
Narrative pole	*Room*		*Static*		*Deep & rich*		*Fully Plotted*	**BB** **LBP** **PP** **RD** **So** **Sp** **SM**
				BB		**Tr** **Ca**		
	Linear	**Br** **BB** **RD** **PP** **SM** **Tr** **LBP** **So**	*Usable*	**Br** **PP** **RD** **So**		**LBP**	*Satellites*	
	Labyrinth		*Modifiable*	**SM** **LBP** **Ca** **Sp**	*Flat characters*	**PP** **Br**	*Dynamic Kernels*	**Ca** **Tr** **Br**
	Hub-shaped	**Ca** **Sp**	*Creatable*	**Tr**		**So** **Sp**		
Ludic pole	*Open*		*Inventible*		*Bots*	**SM** **BB** **RD**	*No kernels*	

Figure 1 – Ludo-narrative positioning of different platform videogames

As we stated, in this quest type of stories, the hero is the centre of all the action and cannot be considered as a simple agent. Considering the hero as a fifth element on the presented model, we propose the following descriptions to define the features from the narrative to the ludic pole:

- A **character with personality** that is presented and explored throughout the game, as the presented case of the videogame *Trine*.
- An **expressive character** that shows his/her current feelings, opinions or suggestions, even though without any particular knowledge about the inner self, such as the *Sackboy* character introduced in the videogame *Little Big Planet*.
- A **living character** that suggests possible feelings but without manifesting them in any way through the game, such as the initial versions of the character *Mario*.

- An **identifiable character** in a somehow adequate level of realism to be majorly accepted as a certain object, such as a car or a spaceship.
- A **simple object** without any particular physical meaning by itself but that will mean something by user contextualization, such as the pads on the game *Pong*.

Excluding *Little Big Planet*, *Trine* and *The Cave*, which bring up the narrative side on the represented heroes, the remaining games of our sample contain a simple living character as the hero. This is another confirmation on the lack of efforts regarding storytelling in those games, as the hero does not present a particular narrative factor.

Another aspect that may justify the referred disassociation from stories to *platformers* is that they tend to present gaming elements that are disconnected to the main story. Typically, *platformers* contain a briefing mechanism when the level starts (a cutscene, an animation, a small textual description, or others) and a similar debriefing mechanism on the end of the level. However, *in-between*, where the game action effectively occurs, world, objects, agents and events do not add any information to the initial briefing. Looking at our previous quest categorization, these games tend to present a universe and a main goal which will lead to a set of well-organized small goals and actions. These games normally do not contain an intricate story because they do not contain a quest mechanism that is able to bridge the main story principles to small goals and actions. When those mechanisms exist, the perception changes.

3.3 Level design patterns for platform videogames

We have seen that some aspects may be common to RPG's or similar games and *platformers*, especially in contemporary approaches for the latter, which try to include a quest based mechanism to bridge gameplay with the story. As seen, those quests can be described in different abstraction levels. Top layers represent the game core and it is not our objective to create them automatically. Lower levels are meaningless actions by themselves and there are already effective approaches to combine them automatically. Consequently, we focused the middle abstraction levels, namely the quests and goals. With those layers in mind, we have observed level content on the previous list of videogames to identify the main level design patterns. This section presents the quest design patterns for platform videogames that we have identified and decomposed into goals as follows:

- *Checkpoint*. This is the most common type of quest in platform games. It consists on reaching a certain part of the scenario with a certain goal. Those goals might be decomposed on the following types:
  - Talk to someone (e.g. in *Little Big Planet* the character breaks one of *Zola*'s creation and is told that should talk to him);
  - Grab a certain item (e.g. in *Prince of Persia* the character has to find a sword);
  - Achieve a milestone on the path (e.g. in *Trine*, characters explicitly state land regions as milestones on the main path);
  - Deliver an item to someone or someplace (e.g. in *The Cave*, each possible character has a certain item of desire to be collected within a chapter);

  - Escort something/someone (e.g. in the *Bride Reception* level in *Little Big Planet* the character finds the groom injured in a cave and is asked to transport him back);
  - Escape a certain enemy or object (e.g. in *Rick Dangerous*, the first level starts with the player escaping a rolling stone ball);
  - Activate (or deactivate) a certain entity (e.g. in the end of the last level of *Rick Dangerous*, the character has to disable a missile);
- *Hostile Zone*. This pattern represents a stop to the main movement mechanics in a particular part of the level. It may be represented by one of the following goals:
  - Defeat *n* opponents (e.g. in *Trine 2* the player encounter hordes of Goblins);
  - Defeat a horde of enemies until a certain condition is reached (e.g. in *Trine 2* the player encounters some situations where enemies are spawned infinitely while he/she tries to accomplish another goal, such as opening a door);
  - Defeat one particular opponent, in the typical form of final *boss* (e.g. in the end of each act, *Sonic* has an encounter with *Dr. Robotnik*)*;*
- *Multiple Checkpoints*. Refers to quests that represent multiple checkpoints that should be performed in a specific order, such as the following:
  - Key/Lock situations, where the character has to take a detour to get a certain item that will allow passage (e.g. in *Braid* the character has to grab keys that open certain doors);
  - Switch/Door, which is a similar situation to Key/Lock but in this case passage is allowed because of a certain trigger. This may lead to a timed challenge, as the trigger might open passage for a short time window (e.g. in *Prince of Persia*, the character frequently presses buttons that open certain gates);
  - Composed Key/Lock, which are the situations where there is a need to gather multiple items to get passage (e.g. in some community levels of *Little Big Planet* we observed doors that open with a set of different keys);
  - Composed Switch/Door situations represent the need to activate several triggers to get passage (e.g. In *Trine*, some contraptions that are controlled with multiple levers that should be pushed in an appropriate order to open some passages);
  - Character modifier situations represent a first stop on a certain entity that will somehow change the characters' features, allowing him/her to reach another checkpoint that otherwise would not be accessible (e.g. in *Spelunky*, the player can get a rope to access additional areas);
- *Puzzle*. A non-trivial combination of game elements to overcome a small part of the level. Puzzle nature, content and difficulty are particularly subjective and we will consider those situations as a quest without a decomposition in different goals.

In addition, we have identified a set of game patterns to promote additional gameplay, which we have previously referred as side-quests, as follows:

- Grab collectible points. Represent the inclusion of scoring elements without any gameplay related reward or with a simple impact on the number of tries (e.g. in *Blues Brothers*, player earns an extra life for every 100 gathered records).
- Reach a secret zone. Represent the inclusion of areas that are not directly identifiable by the player, and that may contain the following type of reward:

- Collectibles data (game art or videos) that do not have any effect on the game and are mere trophies (e.g. in *Trine*, there are secret arcs that unlock game art);
- Collectible powers, that enhance the avatar features (e.g. in *Super Mario*, some blocks hide flowers that upgrade the character).

The previous structure of quests and goals allows a designer to sketch the content of a level that represents those concepts. In the next section we will observe how these ideas could be materialized into an automated generation tool.

4 Quest-based procedural content generation

In this section we briefly present a prototype that is currently being developed to generate platform levels based on the previously described patterns. Although the focus of this article is the previous study, this level editor unveils the potential of the referred ideas. Figure 2 shows the current UI in the creation of an example for the videogame *Prince of Persia*. It consists on a semi-automatic process, where the designer manually defines the sequence of quests with their respective goals and a set of side-quests, as observable on the left part of the figure. The automatic generation process takes those quests as input and generates the geometry using a set of algorithms that differ according to the used patterns. The output is presented on the right part of the window, where the generated segments have been highlighted for comprehension aid.

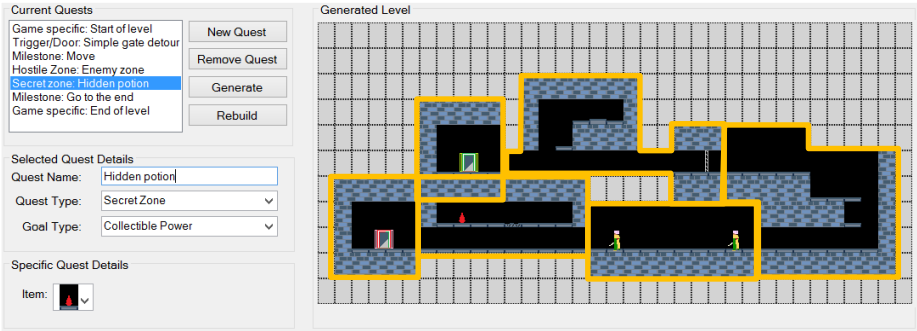

Figure 2 – Prototype application for automated level generation based on quests

It is possible to rebuild a whole level from scratch or only certain quests, allowing a designer to regenerate some parts while keeping others intact. Figure 3(a) shows a sample of multiple variants that were generated for the pattern *Checkpoint* in different full generation runs. For the generation of this specific type of quest in *Prince of Persia*, a composition algorithm was implemented in association with an automatic chunk detection process. *Checkpoint* segments were thus created by retrieving common linear segments from the original set of levels in this videogame.

When the system is rebuilding a specific quest, the algorithm receives the previously existing segment and uses the entry and exit points of that same section as restrictions for the generation process. This gives the algorithm the necessary information to create an alternative geometry that can replace the previous one.

Also, most patterns can be set according certain parameters. As observable back in Figure 2, the side quest *Secret Zone* allows the definition of an item to include in that zone. Moreover, some patterns allow the definition of difficulty based variables. An example is the *Hostile Zone* pattern, which we have implemented to map user skills to the frequency of opponents and respective strength. Figure 3(b) shows different generated alternatives for the same segment with distinct difficulty profiles, in this case using the videogame *Infinite Tux*. A simple terrain displacement generator was used followed by a random inclusion of opponents regulated by the difficulty estimation constraints (roughly, a certain difficulty profile has to match a respective frequency of opponents).

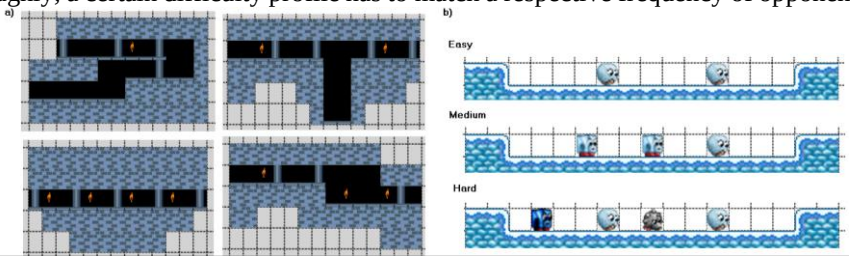

Figure 3 – (a) Alternative geometries generated for a certain patterns and (b) alternative configuration of a certain level part configured for different difficulty settings

As we are decomposing the structure in quests and goals, generation algorithms are responsible for small level parts at each time, reducing the computational complexity. For instance, evolving complete level structures with genetic algorithms as a global process is a time consuming task but generating a small portion to match a certain quest can be achieved promptly. Also, this decomposition into smaller regions potentiate simpler algorithms to be used integrated with more complex techniques. As an example, the referred *Checkpoint* pattern typically consists on a systematic sequence of jumps and/or traps, which can be generated with ease using an *ad-hoc* algorithm. Still, these parts can be merged with other segments representing quests that require more complex algorithms, such as the *Multiple Checkpoint* pattern where paths and routes must be analysed to create those situations.

5 Conclusions and Future Work

We have studied the storytelling capabilities behind platform videogames to identify possible improvements on automatic level generation techniques for this genre. The content in *platformers* tends to be structured in a way that should be considered more narrative than ludic, as most elements have a predefined structure. Yet, these games do not typically contain a rich story because storytelling elements are not explored and, as we stated, agents have not been used properly in that context. This is something that game designers should reason about and consider in the future.

Regarding level design patterns, we identified that the story represented in a videogame can be decomposed in different abstraction layers and level design patterns refer to a certain layer. Different *platformers* were analysed to identify the main containing structures where quest principles were used. A decomposition model was proposed, based on observed situations and recalling existing approaches for other genres. The

final result is a set of level design patterns grouped in two abstraction layers, quests and goals, which can lead automatic generation processes to a more contextualized output. Lower abstraction patterns presented in some other works can be integrated to generate level parts where a certain quest or goal is being represented.

As stated, it is important to bridge the higher and the lower abstraction layers of level design patterns in order to improve existing automated generation. With that in mind, a prototype for level construction using the identified patterns in multiple layers has been designed. Our first tests with our primal prototype showed us that this can be a fast way of semi-automatic level generation, having a human to define the main level content structure with the identified patterns and having lower level generation algorithms in the bottom to concretize those structures. At the moment, a few simple generation algorithms were associated with the quest creation mechanism. The main goal at the moment in this aspect is to complete the prototype by including different generation algorithms, considering that some will be more suitable for certain patterns.

Finally, multi-player gameplay is one aspect that has been evidenced in recent platform videogames. Generating a level that is suitable for multiplayer requires to take into account a few more features. For that, one of our current goals is also to map the multiplayer design patterns proposed Rocha *et al.* [31] and El-Nasr *et al.* [32] in our prototype, allowing the designers to create levels with specific multi-player situations.

6 Acknowledgments

This work was partially funded by IPS under FCT/MCTES grant SFRH/PROTEC/67497/2010 and CITI under FCT/MCTES grant PEst-OE/EEI/UI0527/2011.

7 References

1. Niesz, A., Holland, N.: Interactive fiction. Critical Inquiry. 11, 110–129 (1984).
2. Frasca, G.: Ludology meets Narratology: Similitude and differences between (video)games and narrative (1999).
3. Aarseth, E.: A narrative theory of games. Foundations of Digital Games. 129–133 (2012).
4. Shaker, N., Togelius, J., Yannakakis, G.N., Weber, B., Shimizu, T., Hashiyama, T., Sorenson, N., Pasquier, P., Mawhorter, P., Takahashi, G., Smith, G., Baumgarten, R.: The 2010 Mario AI Championship: Level Generation Track. IEEE Transactions on Computational Intelligence and AI in Games. 3, 332–347 (2011).
5. Smith, G., Whitehead, J., Mateas, M.: Tanagra: A Mixed-Initiative Level Design Tool. Foundations of Digital Games - FDG '10. pp. 209–216. ACM Press, New York, New York, USA (2010).
6. Smith, G., Whitehead, J., Mateas, M.: Tanagra: Reactive planning and constraint solving for mixed-initiative level design. IEEE Transactions on Computational Intelligence and AI in Games. 3, 201–215 (2011).
7. Smith, G., Whitehead, J., Mateas, M.: Tanagra: An Intelligent Level Design Assistant for 2D Platformers. Artificial Intelligence and Interactive Digital Entertainment. (2010).
8. Compton, K., Mateas, M.: Procedural Level Design for Platform Games. Artificial Intelligence and Interactive Digital Entertainment (AIIDE). pp. 109–111 (2006).

9. Smith, G., Cha, M., Whitehead, J.: A framework for analysis of 2D platformer levels. ACM SIGGRAPH symposium on Videogames - Sandbox '08. p. 75. ACM Press, New York, New York, USA (2008).
10. Smith, G., Treanor, M., Whitehead, J., Mateas, M.: Rhythm-based level generation for 2D platformers. Foundations of Digital Games - FDG '09. 175 (2009).
11. Smith, G., Whitehead, J.: Launchpad: A Rhythm-Based Level Generator for 2-D Platformers. Foundations of Digital Games. 3, 1–16 (2011).
12. Smith, G., Gan, E., Othenin-Girard, A., Whitehead, J.: PCG-based game design: enabling new play experiences through procedural content generation. Workshop on Procedural Content Generation in Games. 5–8 (2011).
13. Smith, G., Othenin-Girard, A.: PCG-based game design: creating Endless Web. Foundations of Digital Games 2012 (FDG '12). (2012).
14. Mawhorter, P., Mateas, M.: Procedural level generation using occupancy-regulated extension. Computational Intelligence and Games. 351–358 (2010).
15. Pedersen, C., Togelius, J., Yannakakis: Modeling Player Experience in Super Mario Bros. Computational Intelligence and Games (CIG '09). IEEE Press (2009).
16. Mourato, F., Próspero dos Santos, M., Birra, F.: Automatic level generation for platform videogames using genetic algorithms. Advances in Computer Entertainment Technology (ACE 2011). pp. 8:1–8:8. ACM, New York, NY, USA (2011).
17. Mourato, F., Birra, F., Próspero dos Santos, M.: Enhancing level difficulty and additional content in platform videogames through graph analysis. Advances in Computer Entertainment Technology (2012).
18. Mourato, F., Birra, F., Próspero dos Santos, M.: Integrated System for Automatic Platform Game Level Creation with Difficulty and Content Adaptation. Entertainment Computing-ICEC 2012. pp. 409–412. Springer Berlin Heidelberg (2012).
19. Björk, S., Lundgren, S., Holopainen, J.: Game Design Patterns. Digital Games Research (2003).
20. Hullett, K., Whitehead, J.: Design patterns in FPS levels. P Foundations of Digital Games - FDG '10. 78–85 (2010).
21. Lewis, C., Wardrip-Fruin, N., Whitehead, J.: Motivational game design patterns of 'ville games. Foundations of Digital Games. 172–179 (2012).
22. Dahlskog, S., Togelius, J.: Patterns and Procedural Content Generation: Revisiting Mario in World 1 Level 1. FDG - Workshop on Design Patterns in Games (2012).
23. Sorenson, N., Pasquier, P.: Towards a generic framework for automated video game level creation. Applications of Evolutionary Computation. (2010).
24. Ryan, M.: Avatars of story. (2006).
25. Tosca, S.: The quest problem in computer games. Technologies for Interactive Digital Storytelling and Entertainment (TIDSE). (2003).
26. Sullivan, A., Mateas, M., Wardrip-fruin, N.: QuestBrowser : Making Quests Playable with Computer- Assisted Design. (2009).
27. Howard, J.: Quests: Design, Theory, and History in Games and Narratives. A K Peters Ltd (2008).
28. Aarseth, E.: From Hunt the Wumpus to EverQuest : ICEC 2005. pp. 496–506 (2005).
29. Sullivan, A., Grow, A., Mateas, M., Wardrip-fruin, N.: The design of Mismanor: creating a playable quest-based story game. Foundations of Digital Games. 180–187 (2012).
30. Smith, G., Anderson, R., Kopleck, B., Lindblad, Z., Scott, L., Wardell, A., Whitehead, J., Mateas, M.: Situating Quests: Design Patterns for Quest and Level Design in Role-Playing Games. Interactive Storytelling. pp. 326–329. Springer Berlin / Heidelberg (2011).
31. Rocha, J., Mascarenhas, S., Prada, R.: Game mechanics for cooperative games. ZON Digital Games 2008. 72–80 (2008).
32. El-Nasr, M.S., Aghabeigi, B., Milam, D.: Understanding and evaluating cooperative games. Human Factors in Computing Systems. 253 (2010).