Generating Two-Dimensional Platformer Game Levels from Storylines

Pratama Wirya Atmaja[1], Rizky Parlika[2], Faisal Muttaqin[3]
Department of Informatics
Universitas Pembangunan Nasional "Veteran" Jawa Timur
Surabaya, Indonesia
[1]pratama_wirya.fik@upnjatim.ac.id

***Abstract*— Video game industry has been steadily growing as a lucrative entertainment business for decades. Alongside the growth, consumer's demand for bigger video games with more contents to explore and enjoy has also been on the rise. The demand naturally translates to higher development cost to developers of video games, and it is a problem that needs to be addressed. The state-of-the-art solution to the problem is a procedural content generation (PCG), which allows video game developers to generate game contents automatically as opposed to by hand, therefore keeping the development cost as low as possible. One genre where PCG is especially seen as important is a two-dimensional platformer, and the most popular types of the genre's content to be generated procedurally is gameplay level. In this research, we tried to find a way to generate a two-dimensional platformer level from a storyline that is meant to be experienced by the players. We explored the theory of narrative and the principles of platformer games to understand how to make platformer levels that tell certain stories and have dramatic qualities. The resulting method allows developers to translate a high-level storyline into a playable two-dimensional platformer level, therefore reducing development cost as they do not have to create levels separately from storylines. We tested the method by implementing it to translate two storylines into two platformer games. Based on the results, we concluded that our method is quite robust as a basis for PCG algorithms.**

Keywords*— *video game, platformer, procedural content generation, storyline, level

I. Introduction

Video game industry has grown enormously in just a few decades, attracting massive profit that surpasses every other type of entertainment [1][2]. The positive growth is accompanied by the growth of the burden for the developers to deliver increasingly more prominent and more complex products. Players demand more and more gameplay contents to enjoy, and the demand increases the production cost of video games to the point of creating contents by hand becomes inefficient [3].

Procedural content generation (PCG) is the state-of-the-art answer to the problem. It allows developers to automate the creation of game contents, which can be done at run-time when the game is played to give players seemingly endless content [4][5][6][7] or during the development process to help the developers in creating content [8][9]. Although it still has some disadvantages compared to manually creating contents [10], PCG's future is quite promising and it has been researched and implemented extensively.

A. Platformer Games

Platformer is one of the oldest and most popular video game genres ever. Not only is the genre still going strong among video game players [11], it has also been a fertile ground for research, including researches on PCG and even on AI [12].

One of the main challenges for the players of the genre is traversing some levels, each with an intricate design that involves the placements of platforms, obstacles, weapons, power-ups, enemies, and other objects. Level design is therefore seen as a crucial aspect of a platformer video game, and many successful platformer games are praised for the quality of their levels.

B. Narrative in Video Games

A narrative can be found in almost all game genres, although some genres employ it more than others. Although it is not directly related to gameplay, a narrative is useful to make the player feel more engaged and immersed in the game. As a genre well known for being addictive yet straightforward, platformer games too often "spice up" their gameplays with dramatic storylines. Titles like *Ori and the Blind Forest* [13] and *Braid* [14] showcase platformer's potential as a storytelling medium.

C. PCG for Platformer Levels

Three examples of popular platformer games with procedurally-generated levels are Spelunky [15], Rogue Legacy [16], and Dead Cells [17]. A number of approaches exist for automatically generating platformer levels, such as rhythm-based ones [6], difficulty curve-based ones [4], and occupancy regulation [7]. Closer to our goal in this paper, which is finding a way to generate levels from storylines, is what Karavolos, Bouwer, and Bidarra [18] did, which was integrate mission and space into level generation. Missions are essentially dramatic events and therefore can be seen as parts of the narrative. However, it isn't clear whether the mission and space approach is sufficient to bridge the gap between high-level dramatic storylines and platformer gameplay. A more comprehensive understanding of narrative is needed for a more robust method to translate storylines into levels.

II. Proposed Method

Our method's purpose is to translate a story into a ready-to-play platformer level. However, the method can theoretically be extended to generate a complete scenario of a platformer game, instead of individual levels, as the storytelling elements of a level can also be found in a whole-game scenario.

Fig. 1 shows the flow of our method. Our method translates a high-level storyline into a general scenario that can be used to construct playable levels of different platformer games. To achieve it, we designed our method to be as general as possible so that it can encompass platformer gameplay as diverse as possible.

A. Elements of a Platformer Storytelling

We based our understanding of the elements of storytelling in a video game on the work of Mourato [19]. The elements are:

1) The *world* where the story takes place;

2) The *objects* that the player will interact within the story;

3) The *events* that happen in the story;

4) The *agents* that the player meets and interacts with along the way.

We added the fifth element, the *hero* because it is an integral part of a storyline as described in the "Hero's Journey" theory [20].

B. World

In general, any kind of story setting can be translated into a platformer level. However, two things should be considered:

1) How to properly translate a three-dimensional location into a two-dimensional one;

2) How to employ *rhythm* in the resulting platformer level.

Usually, the two-dimensional platformer level can reflect the linear, two-dimensional path the protagonist of the story takes. However, things are more complicated when the events in the story take full advantage of the three dimensions; for example, when the protagonist is attacked from every direction on x, y, and z-axis. In this case, either the actions in the events are adjusted to be more two-dimensional, or some of them are left out completely.

Another thing to consider is the *rhythm* of a platformer level. It is related to the "musical rhythm" that influences the shape and structure of the level. On the one hand, researchers have judged rhythm as an essential factor in the quality of a platformer level [6][21]; on the other, the rhythm of a level is not directly related to the narrative in it. This is why, in our method, the rhythm is taken care of *after* the narrative in the level has been set. Fig. 2 shows our approach to it. First, we create the basic level layout that represents the setting of the story; in this case, one where the protagonist on the left side is tasked to deliver something to the girl on the right side. Second, we adjust the level's topology by applying rhythm to it.

C. Objects

The objects commonly found in platformer games can be seen in Table I. We based the object types on the work of Smith et al. [22]. We did not include *an avatar,* as they are already covered by the hero, and *platforms*, as they are very common in a platformer game and therefore do not need to be specified explicitly.

D. Events

Generally speaking, events can be considered as smaller "sub-stories" that compose a story. It is common in video games with narrative to present the "sub-stories" as of *quests,* which involve the player doing specific actions. Table 1 shows a set of possible actions for events, which is based on the work of Machado et al. [23] on role-playing game quests.

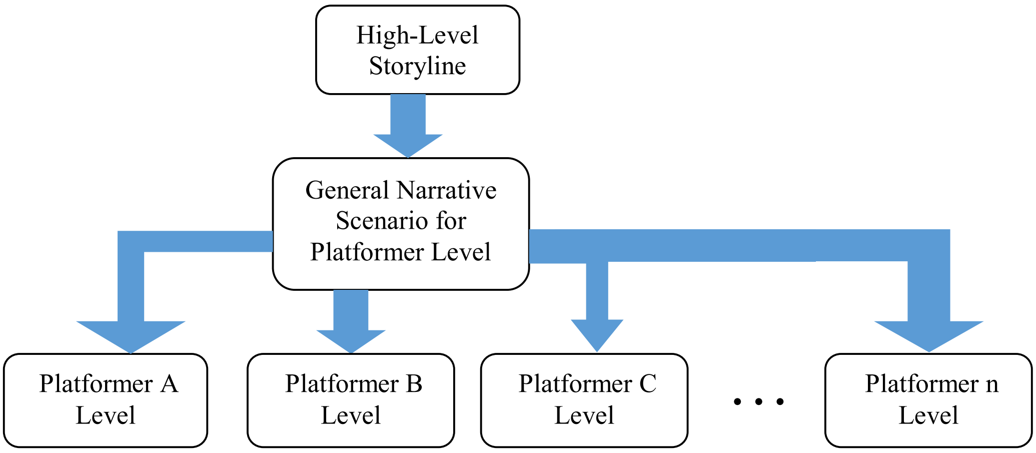

Fig. 1. Flow for Translating Storylines into Platformer Levels

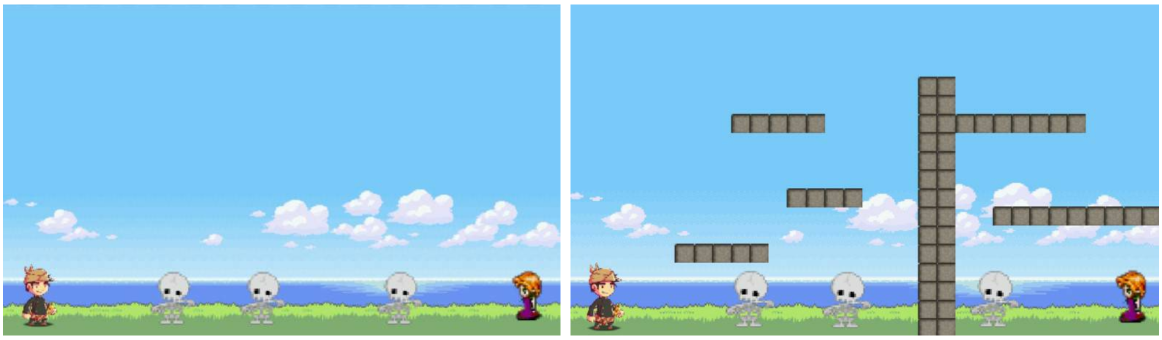

Fig. 2. Adjusting the plain layout of a level with narrative (the left) to make it more rhythmical and dynamic (the right).

TABLE I. COMMON OBJECT TYPES IN PLATFORMER GAMES

Object Type	Description
Obstacle	A static object that may hinder or harm the player character.
Movement Aid	An object that may be used to help the player traverse the level; e.g. a ladder.
Pickable Item	An object that can be moved from the level to the player's inventory.
Trigger	A static, unpickable object that can alter the level in some way when the player interacts with it.

According to the work of Mourato [19], there are three main aspects of a platformer game: movement, confrontation, and environment interaction. Based on those aspects, we split the actions in Table II into five basic types. Those that fall under *movement* and *confrontation* represent the more well-known platformer gameplay. Due to the simple nature of the genre, its players are traditionally allowed to do only two basic actions, which are moving and attacking, such as in Super Mario Bros and Castlevania. Each basic action may consist of several sub-actions; for example, Super Mario Bros players attack by jumping on top of enemies and shooting fireballs at them, whereas Castlevania players strike with the Vampire Killer whip at close range and shoot various projectiles from afar.

In some platformer games, especially those with role-playing elements or well-developed narrative, more action types are available for dealing with things that are outside of the basic moving-and-attacking gameplay. These actions are usually differentiated by where they take place: in the *field* or in the player's *inventory.* Field events happen in the level, between the player and various objects that are scattered around the level. Inventory events, on the other hand, happen in the player's inventory, which is a separate interface for dealing with objects that belong to and are carried in-game by the player. In Table 2 we differentiate between *use,* which is done to a pickable item in the player's inventory, and *activates,* which is done in the field to a pickable item or a trigger. Depending on the game, either pickable items can only be used in the player's item inventory or they can be used directly in the field. If the former rule is followed, a pickable item in the field must first be moved to the player's inventory with *gather* before it can be used.

It is common for the games to use a single dedicated button for field interaction actions; for example, players of Dead Cells [17] can press the "use" button to pick up items on the ground, talk to non-playable characters, check objects in the background, buy and sell items, and do other things. A game may employ many field interaction actions to support its narrative, therefore the "one button for multiple actions" style can be used to keep the control scheme simple. However, some field actions may overlap with each other, creating ambiguity in the controlling; for example, both "activate" and "experiment" (11th and 12th rows in Table 2) can be applied at the same time to the same trigger. To avoid the ambiguity, either the actions are controlled with different buttons, or some of them are not employed in the game.

The last action type is *idling,* which is not controlled by any buttons.

E. Agents

In video games with narrative, agents are non-playable characters that the player encounters and interacts with along the way. We based the agent types in our method on the non-playable character types on https://rpgpatterns.soe.ucsc.edu/doku.php, which appears in the work of Smith et al. [24]. The types are:

1) Enemy characters, which are hostile to the player;

2) Supporting characters, which help the player in some ways;

3) Neutral characters, which neither help nor hinder the player.

The types are originally for role-playing games, but they are also applicable to platformers. Cave Story [25] is an example of a platformer game that uses all three agent types. Table 3 shows the possible characteristics of agents based on the player's action types. The characteristics included are commonly seen in platformer games, such as "flying" and "attacking at the range", and one agent may have more than one characteristic of the same aspect (so an enemy may attack both at close and at long range, for example) or none at all. From the previous action types, we exclude *idling,* because the "wait" action does not require the player to do anything, and *inventory interaction,* because agents do not exist in the player's inventory.

TABLE II. ACTIONS IN A PLATFORMER GAME

Type	Targeted Object	Atomic Action	Purpose
Movement	(nothing)	Goto	Moving to a different location.
		Explore	Wandering in an area until something happens.
	Agent	Follow	Following an agent as he/she moves to somewhere.
		Escort	Leading the following agent to a location.
	Trigger or agent	Stealth	Silently approaching an agent or a trigger without getting noticed.
Confrontation	Agent	Kill	Hurting an agent until he/she is defeated.
	Trigger or agent	Defend	Preventing an agent or a trigger from getting attacked.
		Damage	Damaging or hurting the target.
Field interaction	Trigger	Repair	The trigger cannot be activated until the player does this to it.
		Experiment	Learning what the trigger is for (usually not with 100% certainty)
		Read	Getting an information from a trigger, which is in the form of a book or another readable object.
	Trigger or pickable item	Activate	Using a pickable item or a trigger to activate its effect.
	Trigger or agent	Examine	Learning about a trigger or an agent (may be useful for future events).
	Pickable item	Gather	Moving a pickable item from the field to the player's item inventory.
	Agent	Listen	Learning an information from an agent.
		Capture	Capturing an agent.
		Exchange	Exchange a pickable item from the player's item inventory with another one owned by an agent.
		Free	Releasing a captured agent so that the agent can move freely.
		Give	Giving a pickable item from the player's item inventory to an agent.
		Report	Giving previously learned information to an agent.
		Take	Getting a pickable item from an agent.
		Spy	Spying on an agent from a distance to learn an information.
Inventory interaction	Pickable item	Use	Using a pickable item that is in the player's inventory (instead of in the field).
Idling	(nothing)	Wait	Waiting in an area until something happens.

F. Hero

The hero is the one character the player controls, the avatar of the player. The common characteristics of a platformer player character involve several aspects, such as:

1) The movement capabilities of the character (how high they jump, whether they can fly or not, whether they can run and how fast, etc.);

2) The ways the character attack their adversaries (whether the character uses weapons, whether they use short or long-range attacks, whether they can change their weapons along the way, etc.);

3) The kinds of actions, other than moving and attacking, that the character can perform (talking to non-playable characters, picking up and moving objects around, reading books, etc.), which will be influenced by the events in the story.

III. EXPERIMENT METHODOLOGY

We tested the method by implementing it to translate two stories into platformer levels. The first story is the tearing of the Dutch flag on Yamato Hotel in Surabaya, which is a famous historical event in Indonesia around its independence in 1945. The second story is the Indonesian folklore of Sangkuriang, a young man who tried to accomplish a miraculous feat of building a dam and a large boat overnight. One Super Mario Bros-like platformer game was used to implement the scenarios and the game's characteristics are:

TABLE III. ASPECTS AND CHARACTERISTICS OF AGENTS IN PLATFORMER GAMES

Aspect	Applicable Agent Type	Characteristic	Description
Movement	All	Moving	The agent moves regularly along a path.
		Following	The agent moves toward the player.
		Flying	The agent is unaffected by gravity.
Confrontation	Enemy and supporting	Hitboxes body	Simply touching the agent hurts his/her adversaries.
		Melee	The agent attacks at close range.
		Ranged	The agent attacks from a distance.
		Invincible	The agent cannot be harmed.
Field interaction	Supporting and neutral	Conversation	The player can talk to the agent to do the following actions: listen, report, give.
	All	Sneakable	It is possible to do "stealth" on the agent to "sneakily" do actions on him/her afterward (for example, stealing items from him/her).
	Supporting	Trading	The player may exchange items with the agent.
		Combat assistance	On the player's command, the agent may heal the player, attack the enemies, or do other combat-related actions.
		Misc. actions	The agent may do other things not already mentioned.

1) Composed of levels that must be played sequentially;

2) Each level has a start and a finish point;

3) The player cannot go back to previously explored areas;

4) The gameplay screen can scroll horizontally to the right but not to the left or vertically;

5) Every enemy has "hit boxed body" characteristic, so the player must avoid touching them;

6) The player may perform a close-range attack—by stomping on the enemy's head—and long-range attacks—by shooting various projectiles.

We also added a dedicated field interaction button to help players in interacting with some objects, even if such button is uncommonly seen in Super Mario Bros-like platformers.

A. First Story

The scenario for the first story is as follows: "The hero is an Indonesian warrior with a mission to reach the roof of Yamato Hotel and tear down the Dutch flag there. The hero must enter the hotel and steadily get to the roof of the two-storied building. The hotel is full of Dutch troops with rifles and bayonets who will try to stop and kill the hero from completing their mission."

The platformer-specific scenario of the story is as follows:

1) *Hero:* A warrior equipped with a melee weapon and a firearm (as was common for an Indonesian warrior in World War II era).

2) *Events:* Damage the Dutch Flag.

3) *Agents:* Enemy characters who fight the same way as the hero (with a melee weapon and a firearm).

4) *Objects:* Dutch Flag (trigger), weapons (pickable items), stairways (movement aids).

5) *World:* A two-storied hotel with an accessible roof. Visitors can move between floors through stairways. The Dutch Flag is placed on one end of the roof. The player character starts at the beginning of the 1st floor.

B. Second Story

The scenario of the second story is as follows: "The protagonist's goal is to build a dam and a large boat in one night. The protagonist does it by meditating and conjuring djinns to lend him a hand. The djinns must be conjured one by one in the hilly area surrounding the river where the dam and the boat are meant to be built. In addition to the conjuring, the protagonist must also gather enough wood as the ingredient of the dam and the boat. If the protagonist cannot do both tasks as fast as possible, he will not be able to build the dam and the boat before the dawn of the next day."

We translated the story as follows:

1) *Hero:* A player character who can attack the enemies in any way (combat is not very important here since the focus of the scenario is the completion of the miraculous tasks).

2) *Events:* Activate all conjuring spots and gather enough wood. The events can be done in parallel, but both have to be completed before a time limit.

3) *Agents:* Enemy characters resembling ghosts or spirits (both flying and grounded), and enemy characters resembling wild animals.

4) *Objects:* Conjuring spots (triggers), wood (pickable items), weapons (pickable items), bottomless pits (obstacles).

5) *World:* A long hilly area, an outdoor area that is uneven as it rises and descends at various points, therefore must be traversed carefully. Conjuring spots and wood are scattered across the area.

IV. Results and Discussions

The platformer-specific scenario of the first story was implemented in our game and the result can be seen in Fig. 3. The enemy characters had to be split into melee and ranged types because a Super Mario Bros-like platformer does not have an enemy character type with both attack abilities. Furthermore, every enemy was made immune to stomping attack, making the attack useless and forcing the player to use long-range attacks exclusively. This was because killing enemies in a World War II-era combat using head-stomping would look strange and may even break the player's immersion. The player can gain a weapon by colliding with a weapon pickup, which will not run out of ammunition as it's commonly seen in Super Mario Bros-like platformers.

There was not much modification to the rhythm of the level because the Yamato Hotel would look odd if its layout contained hills, pits, floating platforms, and other geographical features commonly seen in traditional platformer levels. For that reason, we left the topology somewhat plain with only occasional static objects, in the form of furniture, for the player to jump over.

The implementation of the second scenario can be seen in Fig. 4. The conjuring spots, which are represented with diamond-like sprites, can be activated with the field interaction button. As the player's means to do combat was not specified, the game used all the default attacks for the player. Unlike in the first scenario, killing by stomping is viable here as it does not look out of place in a storyline based on a folklore. The topology of the level is also much more varied rhythm-wise as it's meant to resemble a natural hilly area.

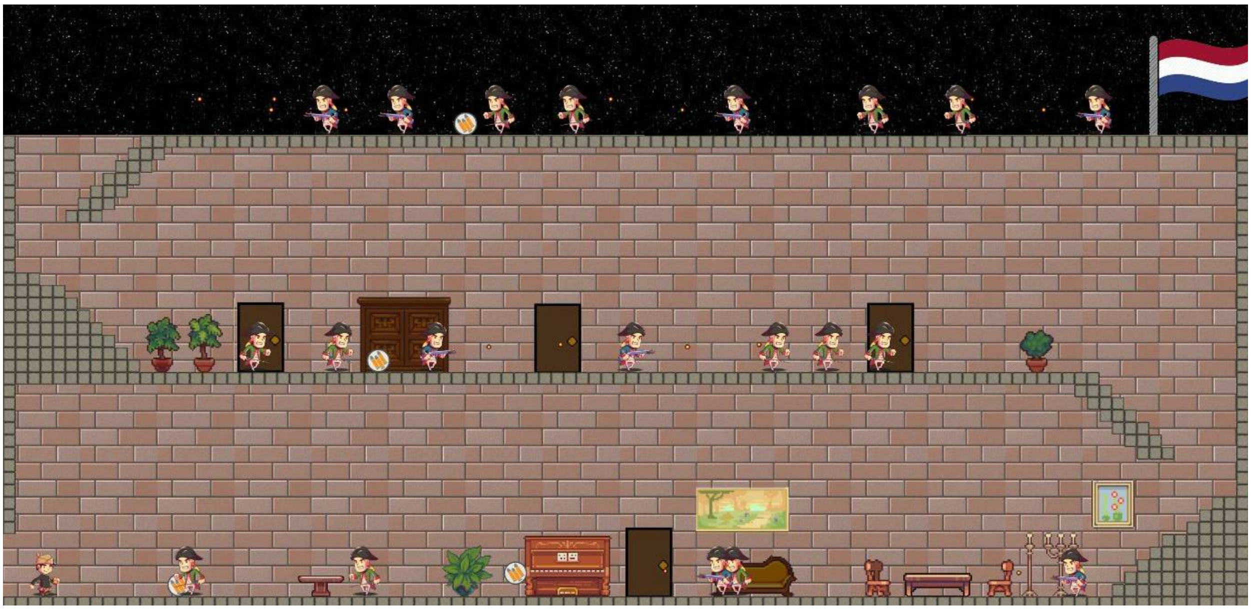

Fig. 3. Platformer Level for Yamato Hotel Scenario

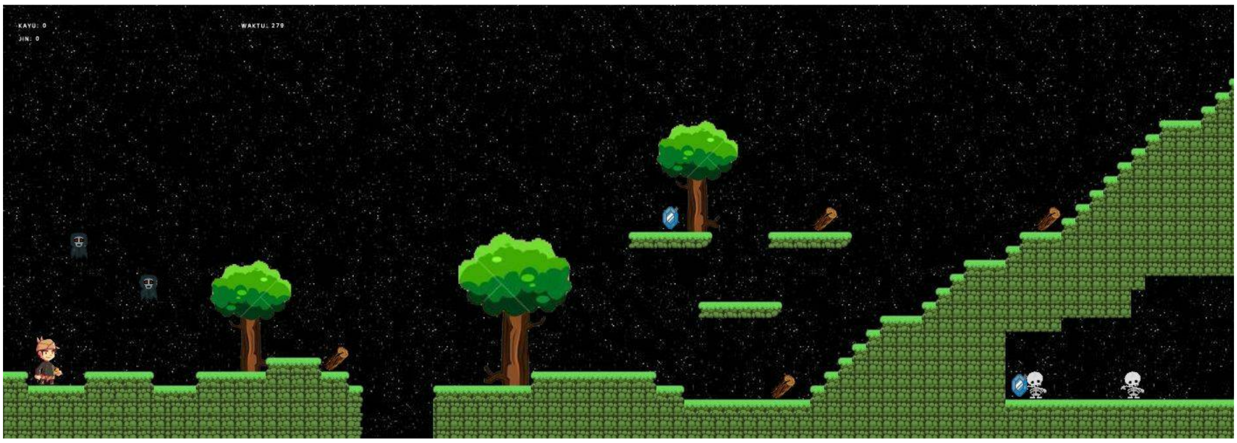

Fig. 4. Platformer Level for Sangkuriang Scenario

V. Conclusions and Future Researches

We tested the method by implementing it to translate two stories into two platformer levels. The first story was the tearing of the Dutch flag on Yamato Hotel at the end of World War II, and the resulting level resembled a hotel but lacked geographical features of traditional platformers such as hills and pits. The combat system was also slightly modified by not allowing stomping on enemies' heads so that the dramatic feels of the story could be retained. The second story was turned into a level that was more in line with traditional platformers, regarding layout and combat mechanics, as the nature of the story allowed the level to be so. Based on the results, we can conclude that our method is quite robust in translating high-level storylines into playable platformer levels.

Future researches can be done to find algorithmic approaches that will allow our translation method to be implemented as an automatic level generation procedure.

References

[1] J. Batchelor, "Games industry generated $108.4bn in revenues in 2017," Gamer Network, 31 January 2018. [Online]. Available: https://www.gamesindustry.biz/articles/2018-01-31-games-industry-generated-usd108-4bn-in-revenues-in-2017. [Accessed 2 September 2018].

[2] K. Ell, "Video game industry is booming with continued revenue," CNBC LLC., 18 July 2018. [Online]. Available: https://www.cnbc.com/2018/07/18/video-game-industry-is-booming-with-continued-revenue.html. [Accessed 2 September 2018].

[3] J. Togelius, G.N. Yannakakis and K.O. Stanley. "Search-Based Procedural Content Generation: A Taxonomy and Survey," IEEE Transactions on Computational Intelligence and AI in Games, vol. 3, no. 3, pp. 172-186, September 2011.

[4] D.-F.H. Adrian and S.-G.C.A. Luisa, "An approach to level design using procedural content generation and difficulty curves," 2013 IEEE Conference on Computational Intelligence in Games, doi: 10.1109/CIG.2013.6633640. October 2013.

[5] A. Baldwin and J. Holmberg, "Mixed-Initiative Procedural Generation of Dungeons Using Game Design Patterns," 2017 IEEE Conference on Computational Intelligence in Games, doi: 10.1109/CIG.2017.8080411. October 2017.

[6] G. Smith, M. Treanor, J. Whitehead and M. Mateas, "Rhythm-based level generation for 2D platformers," in Proceedings of the 4th International Conference on Foundations of Digital Games, pp. 175-182, April 2009.

[7] P. Mawhorter and M. Mateas, "Procedural level generation using the occupancy-regulated extension," in Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games, doi: 10.1109/ITW.2010.5593333, September 2010.

[8] G. Smith, J. Whitehead and M. Mateas, "Tanagra: Reactive planning and constraint solving for mixed-initiative level design," IEEE Transactions on Computational Intelligence and AI in Games, 3(3) pp. 201-2015, doi: 10.1109/TCIAIG.2011.2159716, September 2011.

[9] N. Shaker, M. Shaker, and J. Togelius, "Ropossum: An Authoring Tool for Designing, Optimizing and Solving Cut the Rope Levels," in Proceedings of the Ninth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, 2013.

[10] J. Togelius, A.J. Champandard, P.L. Lanzi, M. Mateas, A. Paiva, M. Preuss, and K. O. Stanley, "Procedural Content Generation: Goals, Challenges and Actionable Steps," in Artificial and Computational Intelligence in Games, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013, pp. 61-75, doi: 10.4230/DFU.Vol6.12191.61

[11] M. McWhertor, "2017 was the year of the platformer," Vox Media, 25 December 2017. [Online]. Available: https://www.polygon.com/2017-best-games/2017/12/25/16795564/the-year-of-the-platformer-2017. [Accessed 1 September 2018].

[12] S. Karakovskiy and J. Togelius, "The Mario AI Benchmark and Competitions," IEEE Transactions on Computational Intelligence and AI in Games, vol. 4, no. 1, pp. 55-67, doi: 10.1109/TCIAIG.2012.2188528, February 2012.

[13] J. Walker, "Wot I Think: Ori And The Blind Forest - Definitive Edition," Gamer Network, 3 May 2016. [Online]. Available: https://www.rockpapershotgun.com/2016/05/03/ori-and-the-blind-forest-definitive-edition-review/. [Accessed 2 September 2018].

[14] C. Purdom, "10 years later, Braid remains the definitive indie game," Onion, Inc., 4 April 2018. [Online]. Available: https://www.avclub.com/10-years-later-braid-remains-the-definitive-indie-game-1824114955. [Accessed 2 September 2018].

[15] N. Shaker, J. Togelius, A. Liapis, R. Lopes, and R. Bidarra, "Constructive generation methods for dungeons and levels," in Procedural Content Generation in Games, 1st ed., Springer International Publishing, 2016, p. 49.

[16] C. Shea, "Rogue Legacy Review," Ziff Davis, LLC., 25 July 2013. [Online]. Available: http://www.ign.com/articles/2013/07/26/rogue-legacy-review. [Accessed 2 September 2018].

[17] A. Webster, "Dead Cells is the best Castlevania game in years," Vox Media, 6 August 2018. [Online]. Available: https://www.theverge.com/2018/8/6/17655322/dead-cells-review-nintendo-switch-ps4-xbox-castlevania. [Accessed 1 September 2018].

[18] D. Karavolos, A. Bouwer, and R. Bidarra, "Mixed-Initiative Design of Game Levels: Integrating Mission and Space into Level Generation," in Proceedings of the 10th International Conference on the Foundations of Digital Games, 2015.

[19] F.J.D.S.V. Mourato, "Enhancing automatic level generation for platform videogames," 2015. Available: https://run.unl.pt/handle/10362/18497

[20] C. Dena, "Finding a Way: Reducing Design Schema Friction in Narrative Design," in Proceeding of DiGRA Australia Queensland Symposium, 2016.

[21] J. Pagnutti, "What Does Bach Have in Common with World 1-1: Automatic Platformer Gestalt Analysis," in Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference, 2016.

[22] G. Smith, M. Cha and J. Whitehead, "A framework for analysis of 2D platformer levels," in Proceedings of the 2008 ACM SIGGRAPH symposium on Video games, doi:10.1145/1401843.1401858, August 2008.

[23] A. Machado, P. Santos and J. Dias, "On the Structure of Role Playing Game Quests," Revista de Ciências da Computação, vol. 12, no. Número Especial Videojogos, pp. 81-98, 2017.

[24] G. Smith, R. Anderson, B. Kopleck, Z. Lindblad, L. Scott, A. Wardell, J. Whitehead, and M. Mateas, "Situating Quests: Design Patterns for Quest and Level Design in Role-Playing Games," in International Conference on Interactive Digital Storytelling, Vancouver, pp.326-329, 2011.

[25] M. Klappenbach, "Cave Story - Free Platform Game for the PC," Dotdash, 13 March 2017. [Online]. Available: https://www.lifewire.com/cave-story-free-game-pc-3901631. [Accessed 2 September 2018].