A.T.L.A.S.: Automatic Terrain and Labels Assembling Software

Alessandro De Francesco
Laura Anna Ripamonti
Davide Gadia
Dario Maggiorini
alessandro.defrancesco@studenti.unimi.it
ripamonti@di.unimi.it
gadia@di.unimi.it
dario@di.unimi.it
Department of Computer Science, Università degli Studi di Milano
Milan, Italy

ABSTRACT

The interactivity and the decision making processes typical of a video game have a strong influence on how the *story* of the game should be told, but also on how the *imaginary world* of the game, where the story takes place, should be structured. As a consequence, there is a growing interest in the development of tools able to couple well with the increasing demanding peculiarities of "game writing" and "world building" activities, especially when game or level designers are called to do also the work of a writer. In this paper, we present *A.T.L.A.S.* (*Automatic Terrain and Labels Assembling Software*), a tool aimed at the automatic creation of complex imaginary worlds for video games, based on Procedural Content Generation techniques, but characterized also by a story-driven approach.

CCS CONCEPTS

• **Human-centered computing**; • **Computing methodologies → Computer graphics**; • **Software and its engineering → Virtual worlds software**; **Interactive games**;

KEYWORDS

Procedural Content Generation, Imaginary Worlds, Video Games, Story Driven, Game Story, Game Writing

1 INTRODUCTION

Video games are peculiar and intrinsically multidisciplinary artifacts, requiring transversal expertise and characterized by an active involvement of their audience. In the development process, the collaboration among multidisciplinary teams is crucial, and it requires specific solutions and approaches [1]. A critical stage is often the creation and definition of the *story* of the game. The characteristics, size and intensity of the story in a video game can vary in a relevant way across different game genres: e.g., it can be very limited in a puzzle game, while it can assume an overwhelming importance in adventure games [23], in which the game progression is strongly linked to the story evolution. Except for the big companies, which can have in their staff professionals devoted to the interactive stories creation (the *game writers*), small independent studios are forced in many cases to ask game or level designers to write stories for their video games. Unfortunately, game writing requires a different set of skills than game design. Actually, a game always includes some elements of interactivity and a certain number of decision making processes: both of them have a strong influence on how the story should be told, but also on how the "game world" should be structured [23]. This is particularly true in games whose success is deeply rooted into strong and ongoing social interaction among players, such as MMOGs (Massively Multiplayer Online Games) and MOBAs (Multiplayer Online Battle Arena). In these game genres, the development process is more focused in the design of the world than on storytelling, since the "story" development is heavily affected by the emerging narrative derived from the social interaction among players [18]. Therefore, video games developers seek solutions and tools able to support their everyday work and to couple well with the increasing demanding peculiarities of "game writing" and "world building" activities (see e.g., [5, 23, 25]), especially when game or level designers are called to do also the work of a writer.

In this paper, we present *A.T.L.A.S.* (*Automatic Terrain and Labels Assembling Software*), a tool for the automatic generation of imaginary worlds for video games. A.T.L.A.S. combines state-of-the-art techniques from the Procedural Content Generation (PCG) field, with a story-driven approach:

GHItaly19: 3rd Workshop on Games-Human Interaction, September 23, 2019, Padova, Italy

indeed, the tool has been designed in order to consider specific elements, created in the game writing stage, in the automatic generation of the virtual environments.

The paper is organized as follows: in Sec. 2 we present an overview of procedural techniques for the generation of imaginary worlds, while in Sec. 3 we present the proposed tool. In Sec. 4 we present the result of an evaluation of A.T.L.A.S. and finally, in Sec. 5 we draw conclusions and discuss major future developments.

2 RELATED WORK

Procedural Content Generation (PCG) [22] is an approach with a long and established history. In the Computer Graphics (CG) field, there is a relevant literature on the use of PCG for the creation of complex models like buildings [21], cities [16, 20], and materials [3, 6]. In the development of video games, several works have shown the potentialities of PCG for the automatic creation of game levels [12, 14, 19], and for the generation of characters with different features [10, 15].

Regarding the automatic or semi-automatic creation of imaginary worlds for games, a certain number of dedicated softwares have been proposed. These softwares differ mainly in the PCG techniques used to generate the imaginary worlds, in the level of parameterization and additional editing provided to the designer, and in the nature of the generated maps (only some of them produces assets which can be imported into the most diffused game engines). Anyway, to the best of our knowledge, none of them offers any form of integration with tools for story-writing.

Fractal Terrain [9] uses fractal functions to generate fictional worlds. The procedural generation can be personalized by setting the values of a quite noticeable number of parameters (such as the climate, relative percentages of ocean and land, etc.). Moreover, the world can be modified after its generation (e.g., to adjust temperatures, mountains heights). However, it produces only a 2D map, which cannot be personalized by adding tags to identify specific locations (e.g., cities and villages names). World Creator [27] exploits the GPU to procedurally generate 3D worlds in real-time. The world can be personalized using a huge number of parameters, it can be edited and exported as an asset, ensuring its portability into the majority of 3D game engines and 3D modeling applications. Anyway, it has a steep learning curve, and it does not allow to add tags to locations. World Machine [28] has a more user-friendly interface, it is based on visual-scripting and it allows to procedurally generate 3D worlds, which can be exported as height-maps or meshes. Again, it is not possible to add tags. Terra Incognita [24] is based on a web site that offers the possibility to generate a 2D map using fractal functions. The interface is very easy, but no customization of the world is allowed, the map is only 2D and it is impossible to add any tag. Fantasy World Generator [8] is very similar to Terra Incognita, but it uses procedural generation instead of fractals. It supports tags, but they are placed automatically, with no possibility to edit them. Fantasy Maps [7] is an open-source application with a web-based interface. It procedurally generates 2D fantasy world in the Tolkienian fashion. The map can be personalized thorough a very limited number of parameters, but cities and villages are automatically added to the map, basing on a grammar consistent throughout the whole map. However, it is impossible to modify the map, once it has been generated, or to add manually other tags. In commercial video games, Civilization VI [4] presents a tool for the automatic generation of game maps. It has a limited amount of parameters, and the generated map is usable only inside the game. Cities and locations are automatically placed in the generated world, with no personalization allowed by the user.

3 A.T.L.A.S. FUNCTIONALITIES

To design and implement A.T.L.A.S., we have started from the analysis of the state of the art presented in Sec. 2. Since applications for imaginary world generation are quite diffused and their basic design principles are quite well established, the intriguing implications from the research perspective are more related to the definition of a set of features able to guide our design in developing a process and a tool able to: (*i*) offer an effective and flexible help both to game designers and to game writers, (*ii*) overcome the limits intrinsic in existing similar products, and (*iii*) smoothly integrate the "world building" activity with the output of the "game writing" stage. Hence, we focused on how we could improve the existing approaches and tools.

As a first step, we decided to focus our attention on the design of fantasy worlds, mainly for the following reasons: they are the most diffused setting in the panorama of story-based games, and they are easily generalizable to other settings; e.g., by choosing the appropriate value for certain generation parameters (like the relative quantities of water and land), they can be easily moulded into representing other settings (e.g., alien planets).

As any game designer or game writer could easily testify, to make an imaginary world "credible" and convincing for the player, it must be "consistent", in the sense that no aspect in it should be perceived as "out-of-place" [2]. Then, we have decided to use the consistency of the world as our guiding light in the design process of A.T.L.A.S.. In particular, we have rooted our generative approach into an approximated simulation of the physical phenomena at the basis of the evolution of our planet: we have selected a set of elements impacting on the configuration of the land which is sufficiently effective to generate convincing landscapes and maps for games. Our approach is top-down: it moves from the general (i.e., the generation of the Earth's crust) to the particular (roads, villages, specific buildings). Also, we have split our generation process into two subsequent phases: the first phase aims at creating the orography of the environment, basing on elements of *physical geography* (plate tectonics, rainfall and moisture maps, hydrology, etc.), while the second phase, basing on *political geography* principles, adds inhabited areas to the map (i.e., cities, villages, roads, etc.). In this second phase, elements from the story writing process can be used as input to A.T.L.A.S.. In particular, we have developed A.T.L.A.S. in order to be also used in conjunction with GHOST, a tool [11] aimed at interactively helping the game and level designers to produce a solid narrative structure for stories, plots and tales to include in games. Figure 1 summarizes the whole generation process, showing the two interconnected phases. To make A.T.L.A.S. easy to integrate in a game development pipeline, we have decided to implement it using the Unity3D game engine.

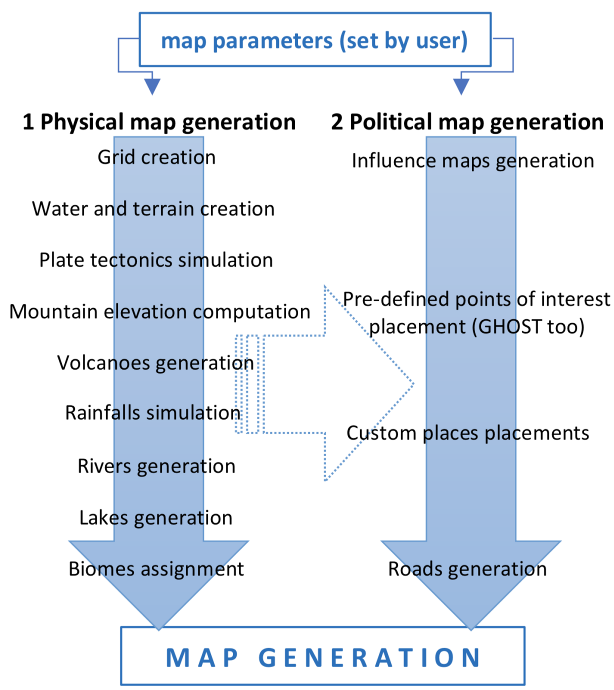


Figure 1: Scheme of A.T.L.A.S. generation process.

A.T.L.A.S. physical map

As shown in Fig. 1, the generation of the physical map has a modular structure: each step in the generation of the orography is affected by a different subset of parameters, and is based on its own generation technique. This approach simplifies the overall generation process, at the same time guaranteeing an outcome highly "credible". Moreover, it allows the maximum freedom to the designer in customizing every aspect of the world. Fig. 2(a) shows the GUI of the proposed tool, with the possible parameters for the generation of the physical map.

Grid creation. The first step is the generation of the polygonal mesh of the world. The user can set the dimension of the map, and its density (i.e., the overall number of triangles). The mesh is created by applying Voronoi tessellation on a set of random generated points on the map, and then considering Delaunay triangulation on the generated tessellation [17]. This approach produces irregular cells, thus improving the "credibility" of the terrain once the produced land is rendered. Moreover, Delaunay triangulation provides an adequate navigation graph useful to support the pathfinding algorithms [13] often used in video games.

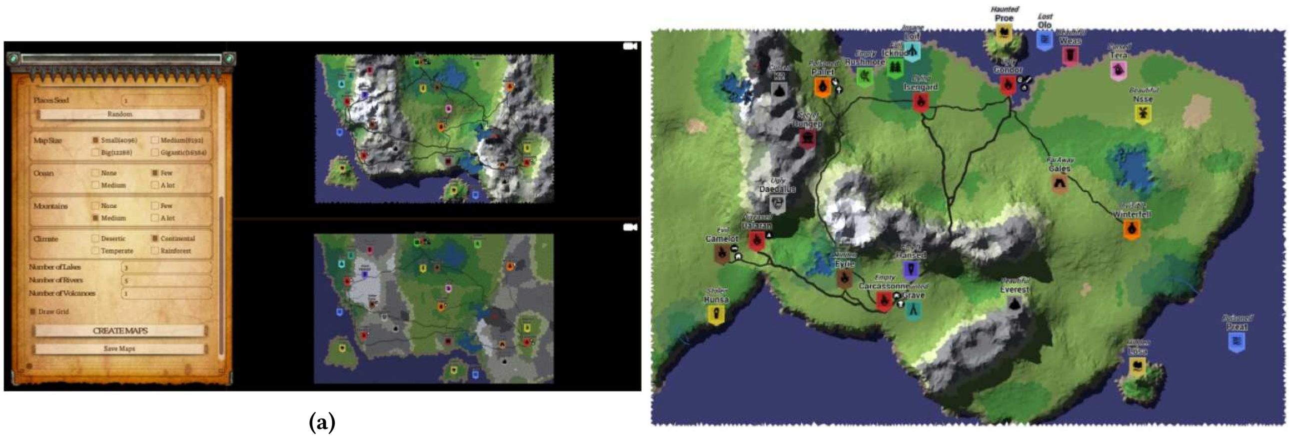
Ocean and land creation. A.T.L.A.S. then subdivides the generated world in a random number of submeshes (i.e., the *tectonic plates*). Each tectonic plate can be labelled "land" or "ocean"; the number and dimension of the oceans are determined on the basis of specific parameters set by the user.

Plate tectonics simulation. Once all the plates are in place, A.T.L.A.S. simulates tectonics effects: for each plate it is assigned a force, whose direction and magnitude affects the interaction of the plate with its neighbouring ones. The forces magnitude could be modified by the choices of the user (e.g., if she has set a high value for the number of mountains, plates will collide with a stronger impact). According to the type of interaction at the borders of two plates, we can have constructive (a depression is produced), destructive (a high ground appears) or conservative (a small rise will be generated) margins. The effect on the surface are milder for oceanic plates. The final effect of this simulation produces a map that is perceived as realistic, since it mimics the natural phenomena affecting the surface of our planet.

Mountain elevation computation. Once the high grounds and the depressions have been appropriately distributed on the grid, the elevation of each single cell is calculated by means of a distribution based on Perlin-noise [6]. The final height of each cell is obtained by adjusting it to that of its neighbours, in order to avoid unnatural effects due to excessive displacements.

Volcanoes generation. The following step is the creation of volcanoes (if any): one or more random mountains near a destructive margin are turned into volcanoes.

Rainfalls simulation, rivers and lakes generation. Basing on the climate set by the user and on the orography, a moisture map is created with a random distribution and magnitude, using again Perlin Noise. The rainfall distribution is used both to place lakes (they are randomly scattered throughout



(a)

(b)

Figure 2: A.T.L.A.S. GUI (a) and an example of generated imaginary world (b)

the map, but they appear more frequently in areas with a lot of rainfall and moisture, avoiding very high mountains) and to define the elevated areas adapt to place the springs of rivers. Both the number of rivers and lakes is affected by the parameters set by the user.

Biomes assignment. The last step assigns biomes to areas. We use a simplified version of the Whittaker diagram [26], where temperature is inversely proportional to the elevation, and moisture is directly proportional to rainfalls and proximity to water sources.

A.T.L.A.S. political map

During the second phase (Fig. 1), elements like cities, villages and roads are distributed on the map according to the parameters set by the user and the orography of the territory, generated in the first phase.

Influence maps generation. To simplify the positioning of places we have adopted an approach based on influence mapping. *Influence maps* are gray level images representing in each point the effect of a certain parameter or value on the neighborhood. They have been proposed in PCG techniques aimed at the automatic generation of large cities [16], but they are also applied in the development of real-time strategy video games [13]. This approach can be customized to label points in maps according to specific perspectives: for example, we could represent how the existence of a river influences the fertility of a region. In particular, A.T.L.A.S. produces influence maps to track the impact of: rivers, lakes, oceans, fields, volcanoes, mountains, forests, elevation, population. For example, the last variable helps in deciding where to place cities, by keeping track of the areas more favourable to the flourishing of a population. Fig.3 shows an influence map of the impact of heights (lighter areas are the highest).

Pre-defined points of interest placement. Once the influence maps have been defined, it is possible to decide the most appropriate location for points of interests like e.g., cities, castles, forests. In the first step related to the placements of points of interests, we have designed A.T.L.A.S. in order to be part of a larger ecosystem of tools aimed at the creation of contents for story-driven video games. To this aim, a certain number of locations can be imported directly from those present in a structure for a story produced by GHOST [11], a tool for the interactive production of a solid narrative structure for stories to include in games. We have defined positioning rules for each element that could be generated by GHOST: for example, a city is placed randomly in the map, but with highest probability near lakes, rivers, sea, mountains and fields; a forest is placed in an area with an adequate biome. The rules take into consideration also the influence maps: for example, the probability in the placement

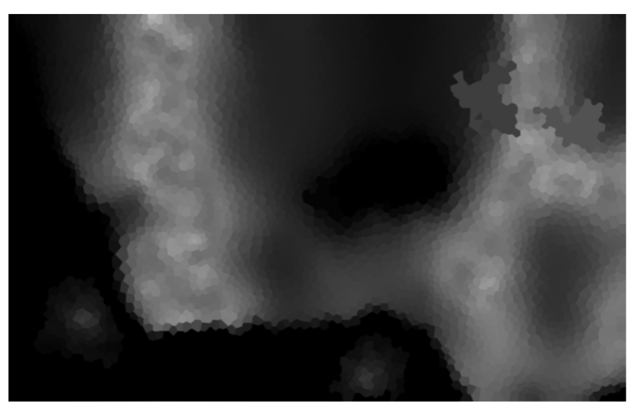


Figure 3: An influence map of the impact of heights used for the political map generation (lighter areas are positions more influenced by heights).

of a city is lowered for already densely populated areas in the influence map.

Custom places placement. A.T.L.A.S. allows also the generation of places directly defined by the user. On the basis of the data in the influence maps, and of the constraints set by the user, the tool determines the most appropriate position inside the map. For example, it is possible to force the spawning of a certain location in a position characterized either by a combination of a predefined biome, height, population density, humidity, presence of a river/road, etc.

Roads generation. To generate a convincing road network, we have emulated the principles adopted in the European Middle Age to create roads: at that time, carriageable roads were generally used only to connect large enough cities. Thus, in A.T.L.A.S., for each couple of relevant places (e.g., cities) the traveling distance is calculated: if it exceeds a predefined threshold, no road will appear. If a road can exist, the A* algorithm [13] is applied to determine the best path to move from the first to the second place. The algorithm avoids unwalkable tiles (such as: water flooded areas, high mountains, etc.). Once the best path has been found, a road is built, whose width is directly proportional to the product of the population of the two locations and inversely proportional to their distance.

In Fig. 2(b), it is shown an example of generated imaginary world by A.T.L.A.S..

4 A.T.L.A.S. EVALUATION

A.T.L.A.S., in its prototypal version, has been tested with real users, in order to evaluate its effectiveness in the support to the imaginary world creation process. We have asked to a group of testers to download the tool and its documentation, and to try to use it to create several imaginary worlds. We have then asked them to answer to a questionnaire, composed of four sections, aimed at collecting respectively: demographic data, opinions on the user experience, on the functionalities of A.T.L.A.S., and suggestions and critics.

We have enrolled 29 testers (38% females, 62% males, between 21 and 45 years old, with average age 25 years). Most of the subjects were master students in Computer Science, all of them specializing in video game design and development (37.9%). Another 24.1% of the testers were working in video game field at the time of the test. 79.3% of the subjects reported to play often to fantasy video games, and 51.7% of them agreed or strongly agreed that coherence between the map in a fantasy video game and the real word is an important factor. More than the half of the testers (51.7%) have tried in the past to create a map for a fantasy game: 35.7% of them has reached the goal in less than 3 hours, while a relevant percentage (21.4%) has spent more than 12 hours on the task.

In Fig. 4 we have resumed the outcomes of the questions on the user experience and the functionalities of A.T.L.A.S..

Regarding usability (Fig. 4(a) and 4(b)), the clear majority of testers gave us a positive feedback on A.T.L.A.S.. The generated maps seemed to be highly appreciated, in both the physical and political aspects (Fig 4(d), 4(e) and 4(f)). Regarding the parameters available in A.T.L.A.S., the feedback was again positive (Fig. 4(c)), even if, in the suggestion section of the questionnaire, some users have reported that they would have preferred a finer control on the generation of places like cities or dungeons.

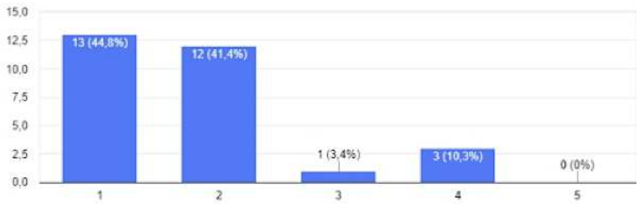
5 CONCLUSION AND FUTURE WORK

In this paper, we have presented A.T.L.A.S., a tool for the automatic generation of imaginary worlds for fantasy video games. We have based the procedural generation of the imaginary world on an approximated simulation of the physical phenomena at the basis of the evolution of our planet, starting from the placement of physical elements like mountains, oceans, etc., and then adding inhabited areas to the map. In this stage, the tool has been designed in order to consider specific places created using tools to interactively help the production of narrative structures for stories to include in games.

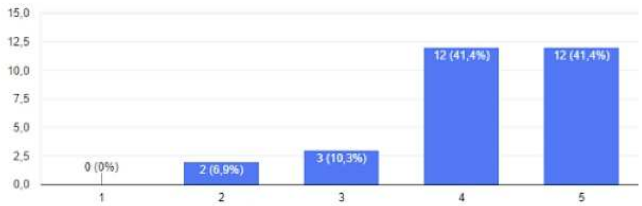
The feedbacks we have collected from the first tests with users are encouraging. Future developments will consider the introduction of a higher number of parameters aimed at a finer tuning of the generated world and of complex places like cities or dungeons, the introduction of a scripting language for a better control on the placement of pre-defined points of interest generated by GHOST (or by other tools for game writing), and the development of a final phase aimed at further checking the imaginary world consistency.

REFERENCES

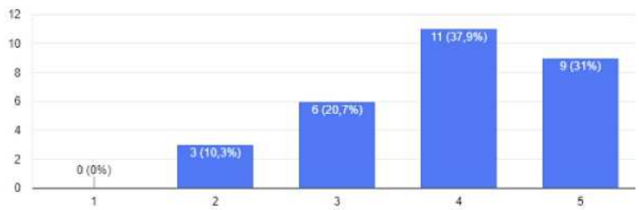
[1] Barbara R. Barricelli, Li Zhu, and Claudia Iacob. 2011. A Meta-Design Model for Creative Distributed Collaborative Design. *Int. J. Distrib. Syst. Technol.* 2, 4 (2011), 1–16.
[2] Richard Bartle. 2003. *Designing Virtual Worlds.* New Riders Games.
[3] Alessio Bernardi, Davide Gadia, Dario Maggiorini, and Laura Anna Ripamonti. 2019. Using a Genetic Algorithm for the procedural generation of layered materials for real-time rendering. In *Proceedings of GAME-ON 2019 Conference.*
[4] Civilization VI. 2019. https://civilization.com/.
[5] Mary DeMarle. 2007. *Nonlinear Game Narrative.* Charles River Media.
[6] David S. Ebert, F. Kenton Musgrave, Darwyn Peachey, Ken Perlin, and Steve Worley. 2003. *Texturing & Modeling: a procedural approach.* Morgan Kaufmann.
[7] Fantasy maps. 2019. http://mewo2.com/notes/terrain/.
[8] Fantasy world generator. 2019. http://donjon.bin.sh/fantasy/world/.
[9] Fractal terrain 3. 2019. http://secure.profantasy.com/products/ft.asp.
[10] Andrea Guarneri, Dario Maggiorini, Laura Anna Ripamonti, and Marco Trubian. 2013. GOLEM: Generator Of Life Embedded into MMOs. In *Proceedings of the Twelfth European Conference on the Synthesis and Simulation of Living Systems: Advances in Artificial Life,*



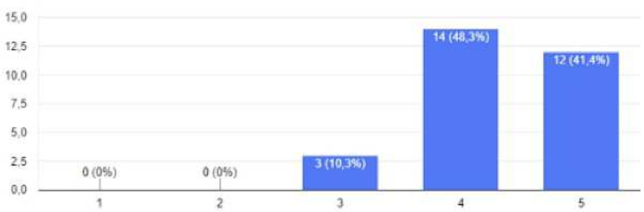
(a) *Is A.T.L.A.S. difficult to use?*



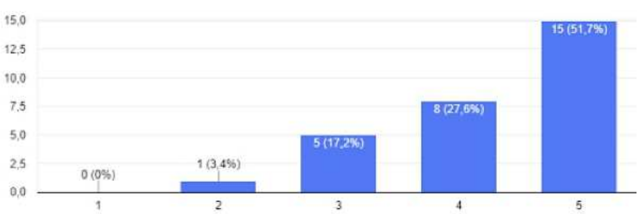
(b) *Is A.T.L.A.S. GUI easy to understand?*



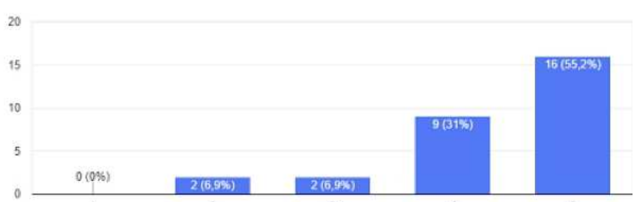
(c) *Are the parameters sufficient to tune the generated map?*



(d) *Are the generated maps adequate for a fantasy game?*



(e) *Are you satisfied by the physical geography (placement of mountains, oceans, etc.) of the generated map?*



(f) *Are you satisfied by the political geography (placement of cities, villages, etc.) of the generated map?*

Figure 4: Evaluation of the user experience and the functionalities of A.T.L.A.S.. Answers expressed using Likert scale (1 = Strongly Disagree, 5 = Strongly Agree).

ECAL. 585–592.

[11] Andrea Guarneri, Laura Anna Ripamonti, Francesco Tissoni, Marco Trubian, Dario Maggiorini, and Davide Gadia. 2017. GHOST: A GHOst STory-writer. In *Proceedings of the 12th Biannual Conference on Italian SIGCHI Chapter (CHItaly '17)*. ACM, New York, NY, USA, Article 24, 24:1–24:9 pages.

[12] Claudio Mazza, Laura Anna Ripamonti, Dario Maggiorini, and Davide Gadia. 2017. FUN PLEdGE 2.0: A FUNny Platformers LEvels GEnerator (Rhythm Based). In *Proceedings of the 12th Biannual Conference on Italian SIGCHI Chapter (CHItaly '17)*. ACM, Article 22, 22:1–22:9 pages.

[13] Ian Millington. 2019. *Artificial Intelligence for Games, Third Edition*. CRC Press.

[14] Fausto Mourato, Manuel Próspero dos Santos, and Fernando Birra. 2011. Automatic Level Generation for Platform Videogames Using Genetic Algorithms. In *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology (ACE '11)*. ACM, Article 8, 8:1–8:8 pages.

[15] Daniele Norton, Laura A. Ripamonti, Mario Ornaghi, Davide Gadia, and Dario Maggiorini. 2017. Monsters of Darwin: A Strategic Game Based on Artificial Intelligence and Genetic Algorithms. In *Proceedings of the 1st Workshop on Games-Human Interaction (GHItaly 2017) (CEUR Workshop Proceedings)*.

[16] Yoav I. H. Parish and Pascal Müller. 2001. Procedural Modeling of Cities. In *Proceedings of ACM SIGGRAPH '01*. ACM, 301–308.

[17] Franco P. Preparata and Michael I. Shamos. 1985. *Computational Geometry: An Introduction*. Springer-Verlag.

[18] Laura Anna Ripamonti, Marco Granato, Marco Trubian, Antti Knutas, Dario Maggiorini, and Davide Gadia. 2018. Multi-agent simulations for the evaluation of Looting Systems design in MMOG and MOBA games. *Simulation Modelling Practice and Theory* 83 (2018), 124 – 148.

[19] Laura Anna Ripamonti, Mattia Mannalà, Davide Gadia, and Dario Maggiorini. 2017. Procedural content generation for platformers: designing and testing FUN PLEdGE. *Multimedia Tools and Applications* 76, 4 (2017), 5001–5050.

[20] Manlio Scalabrin, Laura Anna Ripamonti, Dario Maggiorini, and Davide Gadia. 2016. Stereoscopy-based procedural generation of virtual environments. In *Proceedings of IS&T's Stereoscopic Displays and Applications XXVII (28th Symposium on Electronic Imaging : Science and Technology)*. 1–7.

[21] Michael Schwarz and Pascal Müller. 2015. Advanced Procedural Modeling of Architecture. *ACM Trans. Graph.* 34, 4, Article 107 (July 2015), 107:1–107:12 pages.

[22] Noor Shaker, Julian Togelius, and Mark J. Nelson. 2016. *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Springer.

[23] Lee Sheldon. 2013. *Character Development and Storytelling for Games, Second Edition*. Thomson Course Technology PTR.

[24] Terra Incognita. 2019. http://gmworldmap.com.

[25] Huaxin Wei, Jim Bizzocchi, and Tom Calvert. 2010. Time and Space in Digital Game Storytelling. *International Journal of Computer Games Technology* 2010, Article 8 (2010).

[26] Robert H. Whittaker. 1975. *Communities and Ecosystems*. MacMillan Publishing Company.

[27] World creator. 2019. http://www.world-creator.com/.

[28] World machine. 2019. http://www.world-machine.com.