

# Development of a User-Friendly Application for Creating Tactical Role-Playing Games

José Javier Cortés and Federico Peinado

Departamento de Ingeniería del Software e Inteligencia Artificial  
Universidad Complutense de Madrid  
c/ Profesor José García Santesmases 9, 28040 Madrid (Spain)  
josejaco@ucm.es - email@federicopeinado.com  
[www.narratech.com](http://www.narratech.com)

**Abstract.** Video Game Industry has grown exponentially in the last years. As a consequence of this evolution, development environments have become more complex toolkits focused on the many features that are common in modern video games. This phenomenon has allowed the emergence of tools that non-technical users can use, making video game development accessible to virtually everyone. Due to the high interest on this type of tools we have decided to develop *TRPG Maker*, an intuitive and self-contained tool focused on the development of tactical role-playing games for non-technical users without interest on development's technical details. Along with the development of the tool we have carried out a first experimental validation with real users to determine the degree of usefulness and comfort of the tool, as well as to detect failures and discover possible improvements. Finally, the feedback obtained has been used to create a more complete version of the product, which has been published and made available to the community for free use.

**Keywords:** Software Engineering · Video Game Tools · Video Game Maker · User Centered Design · Interactive Entertainment

## 1 Introduction

Video Game sector is today one of the most important and valued sectors within Interactive Entertainment. In recent years it has experienced a great growth, both in number of users and in business figures. This growth has meant that new development environments, such as *Unity*<sup>1</sup> or *Unreal Engine*<sup>2</sup>, have become very sophisticated tools capable of exploiting to the full the capabilities of the game platforms they are aimed at [16].

The fact that these engines have become more and more complex over time means that users must have a high degree of specialisation in order to use them. Given this situation, the engines already mentioned have added visual scripting systems [7,17], focused on users who do not have a high degree of specialized

<sup>1</sup> Unity, <https://unity.com>

<sup>2</sup> Unreal Engine, <https://www.unrealengine.com>

knowledge, making these tools more accessible. Unfortunately, these improvements are focused on profiles within video game development, in particular on artists and designers who do not possess the knowledge of a professional developer, therefore this type of software is still not accessible to everyone.

There are also other tools such as *GameMaker*[10], *Buildbox*[2] or *Core*[9], which are simpler tools than the engines already mentioned and therefore more accessible. These tools allow to create video games without too much technical knowledge, although they also allow to extend the system to those users who want to go beyond the limits, always in an optional way.

In this paper a new tool called *TRPG Maker* is presented, as a friendly app for creating video games on the Tactical Role-Playing Game (TRPGs) genre for the end user. For this purpose the article will be divided as follows: the next section summarize the related work in the field. Section 3 describes our system and how it works. Next, Section 4 explain the evaluation with real users and discuss their impressions and thoughts. Finally, we close the paper with some conclusions and future work.

## 2 Related Work

The aim of this article is not to carry out a comprehensive study of this field, so we will focus on existing tools and *frameworks* similar to the one we are proposing here, both at the level of commercial applications and at the academic level.

The first object of study on a commercial level has been *RPG Maker*[6]. This tool allows users to create role-playing games without the need for technical knowledge, placing great emphasis on the creation of maps. For this purpose, most of the possible actions within this type of games, such as moving elements of the scenario or having conversations, are supported by events, which the user uses according to his needs without the need to know how it works. Another relevant point to highlight about this tool is the integrated editor that it has, with which it is possible to create and modify any element used in the game, such as skills, characters or enemies.

Other tools reviewed such as *GDevelop*[1] or *Stencyl*[5] also allow the user to create video games without the need for programming skills. Both have a visual editor with which users only need to drag the entities to the game, and in particular *Stencyl* allows to define the same game logic that *Scratch* [13] uses.

At an academic level we have *IsoUnity* [11], a plugin for *Unity* developers that provides the necessary interfaces for the construction of TRPGs [8,12,14]. *IsoUnity* also has a map editor, as well as a sequence editor with which you can define the behaviour of the different entities in the game. This tool provides the necessary features for TRPGs, but it is necessary to have advanced knowledge of C# to be able to use it. For this reason, we believe it is convenient to create a desktop application focused on the same type of videogames, but that could be used by user without technical knowledge about game development. Our tool includes a *point and click* scenario editor and a visual conversation editor, as

well as an event editor and an event queue manager with which it is possible to develop video games without technical knowledge.

### 3 Architecture and Design

For the development of *TRPG Maker* we have taken *IsoUnity* as a reference in terms of software architecture, to implement the following functionalities at a general level: creation of maps, creation of tile-based scenarios, and game event queue management.

The creation of maps and scenarios are essential functionalities of the tool, allowing the user to define the world in which the game is set. For the creation of maps we have two basic elements:

- Nodes: Locations that we will find distributed throughout the map. These locations can be battle scenarios, cities or scenarios where players can trade or accept missions, two of the typical actions in this type of video game. Each node will contain a tile-based scenario where we can place blocks, add decorations and determine in which tiles the player’s and enemy’s units will be located.
- Edges: They will connect the nodes so that the player can move around the map through them.

The queues of game events are responsible for the correct execution of the game logic, so that each mission has its own queue and we will also have a general queue responsible for triggering events on the map. The events that can be used in each mission are the following:

- Motion event: Move a unit from one position to another on the map . This event also allows determining a sequence of points, so that the unit can go from one place to another making a constant sequence.
- Animation event: Starts an animation at a certain point on the map.
- Dialogue event: Starts a conversation between the characters on the map.
- Modifier event: Changes the status of an inventory item. Such a change may be adding a unit to that object, using it, or deleting it.

Several of the events associated with quests are available on maps, such as dialogs or modifiers. We also have the following:

- Encounter event: Spawn a battle on the map.
- Lock event: Allows you to lock and unlock edges of the map, so it is possible to disable certain areas of the map.

Our tool has been designed from scratch, implemented in C++ using Unreal Engine 4.

## 4 Experimental Setup

To test the usability of the tool and to detect errors, we carried out an experimentation session where only the scenario editor was tested when we were at an early stage of development. To do this, we created a contest in collaboration with *Electronic Arts* where the objective was to create a level with the tool making use of a character and texture package integrated into the tool.

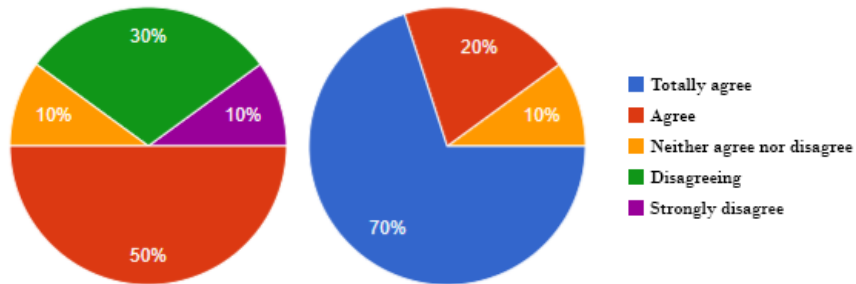
In this experiment we opted to perform remote usability tests, so that evaluators and users do not need to be in the same place. In this test we focus on measuring the effectiveness, efficiency and satisfaction when using the tool. To do this, we collected a series of data on the use of the tool, such as the level's editing time, the number of decorations/characters used on the map or the number of actions used by the editor.

This information was supplemented with a tool evaluation questionnaire. This questionnaire consists of 5 questions related to the usability of the tool, with special emphasis on ease of use and ease of learning compared to other similar tools and two more questions about the use of the tool at a commercial level and in the educational field. In addition we added several free text questions regarding the camera controls, the menu layout, as well as those functionalities that could be added in the future.

10 people participated in the experiment, all of them between 18 and 35 years old. In addition, all the users have advanced computer skills and most of them had already used some visual programming system or video game development tools before. Figure 1 shows the results of the questions related to the use of the tool as a commercial product and in the educational field. Left pie chart shows the opinions about the commercial use of the tool, where 50% consider that it could be used as a commercial product, 30% disagreeing and 10% strongly disagree about the use of the tool at a commercial level while the remaining 10% neither agree nor disagree.

Right pie chart shows the opinions about the educational use of the tool. Surprisingly 70% of the users totally agree about the use of the tool in the academic field and 20% considered it appropriate, while only 10% neither agree nor disagree. These results make us think about the possible uses of this tool beyond a commercial product.

In general, the tool had a fairly good acceptance, both among users who had already used other tools such as *Construct* or *Scratch* and those who had never tried them. Most users agreed that the controls are adequate and that the application has enough information regarding the use of it. On the other hand, most of the improvement proposals revolved around the resources within it, specifically increasing the character template, the materials to edit the levels and the presence of interactive objects on the map.



**Fig. 1.** Diagrams showing the opinions of the users about the use of the tool as a commercial product (left pie chart) and educational level (right pie chart).

## 5 Conclusions

After testing the tool at an early stage of development, we realized that there is still a lot of work to be done in order to achieve a solid product capable of competing with existing commercial tools. Even so, the results obtained both at the technical level and at the usability level are quite promising and the impressions of the users have been very positive.

On the other hand, it is interesting to use the tool at an educational level. Video games are being integrated into the classroom as a means of learning [3,4] that motivates students to explore, experiment and obtain knowledge in their own way. A clear example are the *serious games* [15], video games whose use goes beyond entertainment.

Based on the information obtained from user evaluations we will focus on increasing the content of the tool so that the level editing does not become monotonous and on adding new content to it, see new functionalities and objects for the levels or improve certain interfaces to simplify the use of the tool.

## References

1. Gdevelop <https://gdevelop-app.com>
2. Scell, I.: Buildbox (2019), <https://www.buildbox.com>
3. Barab, S.A., Scott, B., Siyahhan, S., Goldstone, R., Ingram-Goble, A., Zuiker, S.J., Warren, S.: Transformational play as a curricular scaffold: Using videogames to support science education. *Journal of Science Education and Technology* **18**(4), 305 (2009)
4. Bogost, I.: Videogames and the future of education. *On The Horizon-The Strategic Planning Resource for Education Professionals* **13**(2), 119–125 (2005)
5. Chung, J.: Stencyl (2011), <http://www.stencyl.com/>
6. Corporation, A.: Rpg maker <https://www.rpgmakerweb.com>
7. Doran, J.P.: *Unity Game Development Blueprints*. Packt Publishing Ltd (2014)
8. Druet Honrubia, J., González de la Calzada, L.A.: Desarrollo de una herramienta de creación de videojuegos de rol táctico para escenarios isométricos compuestos por bloques (trabajo fin de grado) (2017)

9. Inc., M.G.: Core (2020), <https://www.coregames.com>
10. Overmars, M.: Gamemaker: Studio (2012), <https://www.yoyogames.com/gamemaker>
11. Pérez Colado, I.J., Pérez Colado, V.M.: Un conjunto de herramientas para unity orientado al desarrollo de videojuegos de acción-aventura y estilo retro con gráficos isométricos 3d (trabajo fin de grado) (2014)
12. Pérez Durán, M., Prieto Escolar, J.J.: Desarrollo de una herramienta de autoría en unity para la creación de juegos de rol con combate basado en turnos (trabajo fin de grado) (2018)
13. Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., et al.: Scratch: programming for all. *Communications of the ACM* **52**(11), 60–67 (2009)
14. Santamaría Barcina, A., Alexiades Estarriol, A.: Reconstrucción de una video-aventura isométrica clásica utilizando isounity como herramienta de producción multiplataforma (trabajo fin de grado) (2015)
15. Susi, T., Johannesson, M., Backlund, P.: Serious games: An overview (2007)
16. Toftedahl, M., Engström, H.: A Taxonomy of Game Engines and the Tools that Drive the Industry (2019)
17. Valcasara, N.: Unreal Engine Game Development Blueprints. Packt Publishing Ltd (2015)