A Systematic Literature Review of Game Design Tools

Sebastian Gajewski, Nour El Mawas and Jean Heutte
Univ. Lille, ULR 4354 - CIREL - Centre Interuniversitaire de Recherche en Éducation de Lille, F-59000 Lille, France
{Sebastian.gajewski.etu, nour.el-mawas, jean.heutte}@ univ-lille.fr

Keywords: Game Design, Tool, Systematic Literature Review, PRISMA.

Abstract: Existing research works show the role of game-based learning to increase lifelong learners' learning and motivation. Furthermore, many game design tools are increasingly available and easy-to-use even by people without any technical knowledge, making the choice of the tool difficult. The aim of this paper is to identify and compare game design tools in order to help people to choose the most suitable game design tool for their needs. After having identified 9 criteria that characterize a game design tool, we have conducted a systematic literature review following the PRISMA methodology. This methodology allows us to identify references *via* 2 methods: databases and the interview with a pedagogical engineer. From 302 identified studies *via* 5 databases (*IEEE Xplore*, *ScienceDirect*, *Scopus*, *Springer*, and *Web of Science)*, 18 have been used for the discussion. From the 8 game design tools advised by the pedagogical engineer, 3 have been used for the discussion. These 2 methods allow us to identify 12 game design tools. This research work is dedicated to Technology Enhanced Learning community and pedagogical community, and more specifically to game designers, pedagogical engineers, teachers, and researchers who encounter difficulties in choosing the game design tool the more suitable for their needs.

1 INTRODUCTION

Nowadays, lifelong learning is becoming a central asset, beginning with the university and continuing through the professional career with different jobs. Adults have different needs for education and training throughout their lives (El Mawas et al., 2017). Therefore, lifelong learning involves a continuous learning process, acquiring and expanding skills, behaviours and knowledge throughout the life of an individual (El Mawas, and Muntean, 2018).

Game-based learning play an import role to support lifelong learning (Rasco et al., 2021) and increase the players' learning and motivation. Game-based learning includes gameplay-based learning and game design-based learning (Kafai, 2006). In the gameplay-based learning (instructionist perspectives), learners play a serious game to learn. Whereas, in the game design-based learning (constructionist perspectives), students learn by making games for learning instead of playing games for learning.

This research work is about game-based learning, more specifically about the game design tools used in those learning activities. The systematic literature review we present in this research work is intended for students who will design games to learn, and their teachers who will choose the game design tool the students will use in their classrooms (game design-based learning), and for the game designer who will develop games which will be played by students (gameplay-based learning). More generally, this research work can interest the Technology Enhanced Learning community and the pedagogical community including pedagogical engineers, teachers, and researchers.

This research work takes part of a project where we aim to experiment a game design-based learning activity with nursing students, and to assess the effects of this design activity on students' learning and social flow (Gajewski, El Mawas, and Heutte, 2020).

For this aim, we need to identify the game design tool that the nursing students will use during the learning game design activity. However, in the current context of game design-based learning, it is very difficult to make a clear view of the research landscape on works on game design tools. For instance, a simple, unrestricted Google Scholar query on the term "game design tool" returns about 2 900 000 results at the time of the writing of this paper (early 2022).

Gajewski, S., El Mawas, N. and Heutte, J.
A Systematic Literature Review of Game Design Tools.
DOI: 10.5220/0011137800003182
In *Proceedings of the 14th International Conference on Computer Supported Education (CSEDU 2022) - Volume 2*, pages 404-414
ISBN: 978-989-758-562-3; ISSN: 2184-5026

This paper is structured as follows. Section 2 of this paper oversees the theoretical works concerning criteria that characterize a game design tool. Section 3 details the methodology steps. Section 4 discusses the results of this review of literature comparing to the identified criteria. Finally, Section 5 concludes this paper and presents its perspectives.

2 RELATED WORK

In this section, we consider existing research works related to game design-based learning and we deduce important criteria to characterize a game design tool based on the authors' explanation about game design-based researches.

(Burke and Kafai, 2014) highlight that while programming a video game traditionally required extensive typing (such as Unity based on the C# programming language) in which a syntax error can offset the game play altogether, many design tools are based on a "drag-and-drop" approach to coding like Scratch. In Scratch, users manipulate blocks to build scripts which reduces syntax errors (Baytak and Land, 2011).

Moore (2016) underlines the importance of the tool language so the designer can understand all information on menus. Bertolini (2018) mentions that offering different languages and platform packages to build a video game is helpful for game designers. Thus they can select a language for the game design interface according to their preferences.

(Macklin and Sharp, 2016) considers that good resources and tutorials (manuals, videos…) are mandatory in order to help the game designers in the appropriation of a tool. Note that some game design tools provide the opportunity to users to choose the tutorial's language.

Peppler and Kafal (2010) mention the importance of providing resources, such as backgrounds for the scene, objects and a library of sprites during the game design process. If the game design tool does not provide these scenes and characters, users have to draw by themselves and this needs character designer skills.

Øygardslia and Aarsand (2018) precise that some game design tools are restricted to only one type of game (e.g., adventure, arcade, racing). That is the case of RPG Maker VX Ace where the games must be role-playing games. In contrast, Microsoft Kodu allows users to create different types of games according to (Akcaoglu and Green, 2019).

Øygardslia and Aarsand (2018) accord an importance to the target audience-designer. Are they professional or novice? For example, RPG Maker VX Ace some programming skills and game design experience to create games. Are they adults or children? In the same direction, Baytak and Land (2011) specify that Scratch is a user-friendly interface for children.

According to Akcaoglu (2016), 2 dimensions (2D) or 3 dimensions (3D) modelling is an important criterion in a game design tool. In this research work, the author confirms that the ability to create 3D games in Kodu makes it visually more appealing for young students than 2D environments.

Baytak and Land (2011) consider the prize in the choice of a game design tools: they choose the Scratch tool because it is free of charge. Note that many users have to pay a fee for a game design tool licence.

An (2016) points the export option in game design tools: some tools allow the users to upload and share their games in the game design tool's community website. For example, in Gamestar Mechanic, students share their games in Game Alley, the online community. The export option is directly related to the platforms on which the game can run like Microsoft, Mac OS, iOS, Android, Console, and HTML5.

Based on the presented research works, we identified nine criteria to be considered in the choose of a game design tool: **programming language (C1)**, **Tool language (C2)**, **tutorials (C3)**, **scenes and characters (C4)**, **game type (C5)**, **target audience-designer (C6)**, **2D or 3D modelling (C7)**, **prize (C8)**, and **Export (C9)**. In the following section, we present the methodology followed for the literature review of game design tools.

3 THE PRISMA METHODOLOGY

This literature review follows the methodology described by Kitchenham and Charters (2007), and includes the following steps: developing a review protocol, identification of the need for a review, specifying the research question(s), identification of research, selection of studies, data extraction, data synthesis, and reporting the review.

3.1 Developing a Review Protocol

"*A review protocol specifies the methods that will be used to undertake a specific systematic review*" (Kitchenham and Charters, 2007). The steps used in this literature review will be presented in the following sections.

3.2 Identification of the Need for a Review

Gajewski, El Mawas, and Heutte (2020, 2021) have developed a methodology of game co-design with 11 steps and 4 actors. In the second step of the methodology, the game designer identifies the game design tool the more suitable according to the 9 criteria identified in the previous section.

The objective of this literature review is to analyse the research works related to game design tools between 2010 and 2020 (2020-12-18).

3.3 Specifying the Research Question(S)

This research work aims to (1) identify the game design tools used in the research works, and (2) describe and compare those game design tools according to the 9 criteria.

3.4 Identification of Research, and Selection of Studies

In this section, we describe the inclusion and exclusion criteria used to constitute the body of research works for our study. We detail and justify the choice of the search engines and the search words used.

Firstly, our inclusion criteria are the research works, which are about game design tools, led between 2010-2020. We consider the 5 following search engines: *IEEE Xplore*, *ScienceDirect*, *Scopus*, *Springer*, and *Web of Science*. We have chosen those search engines to find research works that are reliable and up-to-date.

On the other hand, the exclusion criteria are the research works which are not written in English, and the literature reviews of game design tools.

Our search words are "game design tool" and its synonyms. The Boolean search in the search engines was written as follows:

"game design tool" OR "game design software" OR "game design engine" OR "game design program" OR "game design platform"

3.5 Data Extraction[1]

The selection process of the research works is illustrated in Fig.1, and it has taken place as follows.

First, we have exported the results (databases) from each of the 5 search engines in 5 different csv-files.

Then, we have regrouped the results from the 5 search engines in a single csv-file. We have sorted the results alphabetically from A to Z, so we could cluster and, later, delete the duplicates.

All in all, 302 research works have been identified, distributed as follows: *IEEE Xplore* (6), *ScienceDirect* (38), *Scopus* (112), *Springer* (120), and *Web of Science* (26).

Forty-three research works were duplicates, and have been deleted.

Then, we have looked for each research work to access its abstract, and its keywords. All the research works (titles, abstracts, and keywords) have been regrouped in a single docx-file.

One result has been deleted because the paper could not be found.

Nine results have been deleted because they did not include any abstract.

Then, we have read all the 249 abstracts of the remaining research works. To be considered relevant, the title, the abstract and/or the keywords of a research work should include the name of a game design tool or the research work subject should be about game design. Sixty-two research works mention a game design tool or a study on game design. However, nineteen research works were not available for free. One research work was limited to the abstract. One hundred and eighty-seven research works were off topic. In total, two hundred and seven research works have been rejected.

Then, we have read the forty-two remaining research works. Eighteen research works were included for the data synthesis. Twenty-four research works were rejected for different reasons: (1) five research works were about tools which are not game design tool, as Microsoft PowerPoint, (2) sixteen research works were about game design tools which are not available anymore or not sharable by their designers, (3) two research works were about modding or about game level design from existing game, reducing creativity, and (4) one research work was rejected because it was a literature review about game design tools.

In addition, we have identified references via another method[2]: the interview with a pedagogical engineer. 8 game design tools have been advised. 3 have been included in the discussion. 2 references have been excluded because they were not game design tools. 2 references have been excluded because they have not been available. 1 reference has been excluded because it was the same game deign tool identified via databases.

[1] https://guides.lib.unc.edu/prisma/step-by-step

[2] https://www.bmj.com/content/372/bmj.n160

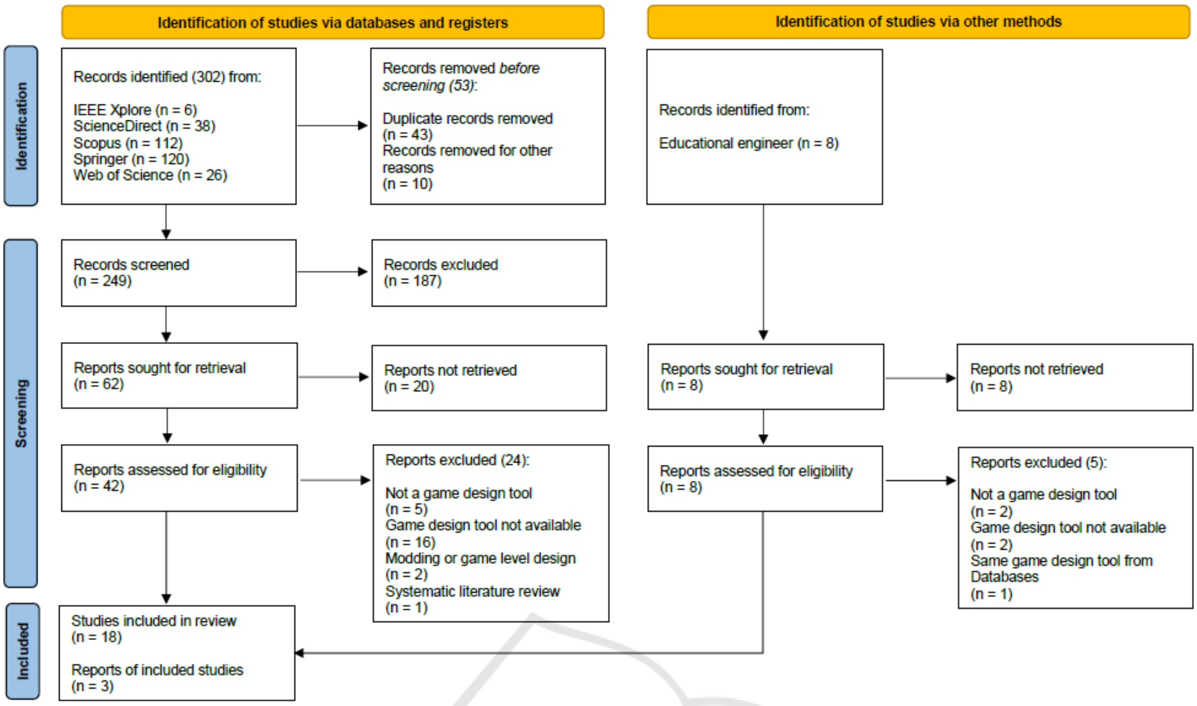
Identification of studies via databases and registers
Identification of studies via other methods
Identification
Screening
Included
Records identified (302) from:
IEEE Xplore (n = 6)
ScienceDirect (n = 38)
Scopus (n = 112)
Springer (n = 120)
Web of Science (n = 26)
Records removed before screening (53):
Duplicate records removed (n = 43)
Records removed for other reasons (n = 10)
Records identified from:
Educational engineer (n = 8)
Records screened (n = 249)
Records excluded (n = 187)
Reports sought for retrieval (n = 62)
Reports not retrieved (n = 20)
Reports sought for retrieval (n = 8)
Reports not retrieved (n = 8)
Reports assessed for eligibility (n = 42)
Reports excluded (24):
Not a game design tool (n = 5)
Game design tool not available (n = 16)
Modding or game level design (n = 2)
Systematic literature review (n = 1)
Reports assessed for eligibility (n = 8)
Reports excluded (5):
Not a game design tool (n = 2)
Game design tool not available (n = 2)
Same game design tool from Databases (n = 1)
Studies included in review (n = 18)
Reports of included studies (n = 3)

Figure 1: PRISMA flow diagram for a systematic review about game design tools from 2010-01-01 to 2020-12-18[3].

3.6 Data Synthesis, and Reporting the Review

After identifying 21 game design tools, of which 18 are from studies *via* databases (5 about *Scratch*, 4 about *Microsoft Kodu, and by the same author*, 2 *GameMaker*, 2 *Agentsheets*, and 1 for each other), and 3 are suggested by a pedagogical engineer, we describe each tool in alphabetical order according the 9 identified criteria.

3.6.1 Agentsheets[4] (T1)

Agentsheets is a visual block-based environment. It consists in dragging and dropping conditions blocks (i.e., "if…") and actions blocks ("…then…") into a script area. *Agentsheets* is available in different languages, as English, and French. On the website of *Agentsheets*, it is written: "*Programming for Kids*".[5]

Agentsheets provides "agents" that can be used as characters or building blocks of the background. Tutorials are available in different languages, as English. French is not available. *Agentsheets* costs $99,40 annually. However, a trial version of *Agentsheets* allow users to try the game design tool for free.

Agentsheets allows users to design 3D games. An inflatable icon turns the 2D characters or building blocks available in the database or drawn by the users themselves into 3D. Agentsheets allows users to publish their games on the web.

This tool was used by Basawapatna *et al.* (2011) where the students worked on their project of game design for 2 weeks. The aim of the research work led by Basawapatna *et al.* (2011) was to present to teachers how to design games, so that teachers could teach students computer science and Computational Thinking.

This tool was also used by De Souza *et al.* (2011) where the students worked on their project of game design for seven 2-hour weekly sessions. The aim of the research work led by De Souza *et al.* (2011) was to compare meaning of game descriptions in Brazilian Portuguese and Visual AgenTalk code.

3.6.2 Alice[6] (T2)

Alice is a « *block-based programming environment*

[3] www.prisma-statement.org/documents/PRISMA_2020_flow_diagram_new_SRs_v2.docx
[4] https://agentsheets.com/
[5] https://agentsheets.com/
[6] https://www.alice.org/

that makes it easy to […] *program simple games in 3D* »[7]. *Alice* provides a list of action blocks that the game designer drags and drops into a script screen to trigger different behaviours. *Alice* is available in different languages as English. French is not available. *Alice* is used by teachers at all levels from middle schools (and sometimes even younger) to universities. So, *Alice* can be used by anyone.

Alice provides different scenes and characters which lots of them are inspired from "*Alice in Wonderland*" because of its name *Alice*. Different resources, in English, are available on the website of *Alice*. *Alice* is a freeware.

Alice allows users to develop 3D games. To run and play an Alice game locally in the Alice player, the Alice game needs to be exported from the Alice Integrated Development Environment (IDE).

This tool was used by Allsop (2016) where the students worked on their project of game design. Once *Alice* has been presented to the students, the students have browsed freely the game design tool for 30 minutes to discover the game design tool's functionalities. Then, they have looked at tutorials available on the web site of *Alice* and on *YouTube*. Once their games have been designed, they have tested them. Unfortunately, the time spent by the students to design their games is not communicated. The aim of the research work led by Allsop (2016) was to study the mental activities of the students when designing a game.

3.6.3 Celestory[8] (T3)

Celestory allows users to develop games without coding using visual blocks that users have to drag and drop (D&D). The game design interface is in different languages, as English, and French.

Celestory provides over 1500 assets and designs. Tutorial are available on the *Celestory* website in different languages, as English, and French. *A* free version of *Celestory* allows users to develop only 1 project. Two different commercial versions of the game design tool, whose prizes are either $19 or $109 monthly, depending on options.

Celestory allows users to design 2D games. *Celestory* allows users to develop different types of games (Playing cards, escape game, interactive movie, quiz…). *Celestory* enables users to develop games to Windows, macOS, Linux, iOS, Android, HTML5.

3.6.4 GameMaker[9](T4)

GameMaker allows users to develop games either using its specific code programming language, *GameMaker Language* (GML), or using a friendly *Drag and Drop* (DnD™) interface without the need for writing any code. The interface of the game design tool is in different language, as English, and French. *GameMaker* proposes powerful tools for beginners and professionals alike.

GameMaker do no provide neither sprites for the characters, nor tiles for the scenes. A tutorial, the *GameMaker Studio 2 Manual*[10]*, is available on its website* in different language, as English, and French. A freeware version (with an unlimited time duration) of *GameMaker* allows users to develop their games and to export them to the GXC platform to share and monetize them. Different commercial versions of the game design tool, whose prizes range from 42,50 € to 679,99 € yearly, depending on option.

GameMaker is a "*2D Game Development Environment*"[11]. *GameMaker* allows users to develop a wide range of games: *shoot them up games*, *platform games*, *tycoon games*, *First Person Shooter game* (FPS)… *GameMaker* allows to export the games in Windows, macOS, HTML5, Android, iOS, Sony PlayStation 4 and 5, Microsoft Xbox One and Series S and X, and Nintendo Switch.

This tool was used by Baytak and Land (2010) where the students worked on their project of game design for 45-minute sessions twice a week for 8 weeks, *i.e.* 12 hours. The aim of the research work led by Baytak and Land (2010) was to study how students designed games reflect their understanding of nutrition.

This tool was also used by Buelin-Biesecker and Wiebe (2013) where the students worked on their project of game design during one daily 45-minute session, five times a week during three weeks, *i.e.* 11 hours and 15 minutes. The aim of the research work led by Biesecker and Wiebe (2013) was to compare the potential for fostering creative problem-solving behaviours between two distinctly different problem-solving environments.

3.6.5 Gamestar Mechanic[12] (T5)

Gamestar Mechanic "*enables students to create their own games without any programming knowledge*"[13].

[7] https://www.alice.org/

[8] https://www.celestory.io/

[9] https://www.yoyogames.com/en/gamemaker

[10] https://manual.yoyogames.com/#t=Content.htm

[11] https://www.yoyogames.com/en/gamemaker

[12] https://gamestarmechanic.com/

[13] https://gamestarmechanic.com/

It uses a point and click interface. *Gamestar Mechanic* is in English. *Gamestar Mechanic* is "*designed for 7- to 14-years-olds but is open to everyone*".

The workshop of *Gamestar Mechanic* provides different blocks that users drag and drop in a tileset which will be the world of the game. The workshop also provides avatars and enemies, items, like keys to open doors, and "systems", like a timer or a health meter. *Gamestar Mechanic* "*consists of three sections: The quest, workshop, and game alley*"[14]. In the quest, students learn the principles of game (rules, games mechanics…) by playing, and they earn sprites once they have successfully completed missions. "*The workshop is where students make their own games*"[15] but they can't design any game as long as they have not earned sprites. "*Game Alley is where students share their games in an online community*".[16] *Gamestar Mechanic* is a freeware.

Gamestar mechanic allows users to design 2D games. *Gamestar Mechanic* allows users to create different types of games: adventure, platform, action, and experimental. Once the game is developed with *Gamestar Mechanic*, it is uploaded in the Game Alley and shared with the online community.

This tool was used by An (2016) where the students worked on their project of game design for one 45-minute session per a day during nine months. The aim of the research work led by An (2016) was to introduce students to the basics of game design.

3.6.6 Microsoft Kodu[17](T6)

Microsoft Kodu uses a visual programming language by tiles. *Microsoft Kodu* is available in different languages as English, and French. *Microsoft Kodu* is "*intuitive, even to the degree to be used easily by children as young as nine or ten*" (Akcaoglu, 2016).

Microsoft Kodu proposes 3 different worlds that users can modify, and different objects and sprites. Tips and Resources are available on the website of *Microsoft Kodu*. They are available only in English. *Microsoft Kodu* is a freeware.

Microsoft Kodu allows users to design games in a 3D environment. *Microsoft Kodu* allows users to design "*different types of games (e.g., adventure, arcade, racing)*" (Akcaoglu and Green, 2019). The games developed with Microsoft Kodu can be hosted on the web, and shared with the Kodu community.

This tool was used by Akcaoglu and Koehler (2014) where the students worked on their project of game design for one 3-hour session once a week during five week-ends, *i.e.* 15 hours. The aim of the research work led by Akcaoglu and Koehler (2014) was to allow students to design games to foster their problem-solving and critical reasoning skills.

This tool was also used by Akcaoglu (2016) where the students worked on their project of game design either during an after-school program (seven 3-hour sessions, i.e. 21 hours), or during a summer camp (eight five-hour sessions, i.e. 40 hours). The aim of the research work led by Akcaoglu (2016) was to teach students basics of computer programming, game design, and complex problem-solving skills.

This tool was also used by Akcaoglu (2014) where the students worked on their project of game design for five hours a day during ten days, i.e. 50 hours. The aim of the research work led by Akcaoglu (2014) was to teach students problem-solving skills, specifically in system analysis and design, decision-making, and troubleshooting.

This tool was also used by Akcaoglu and Green (2019) where the students worked on their project of game design for one 1-hour sessions per week during ten weeks, i.e. 10 hours. The aim of the research work led by Akcaoglu and Green (2019) was to allow students to learn basics of digital game-design and to practice system design skills such as making flowcharts.

3.6.7 RPG Maker VX Ace[18](T7)

RPG Maker VX Ace is "*incredibly simple to learn and use, being accessed through a simple point and click interface*", and that users can create a game "*without the need for any coding knowledge*"[19]. So, Øygardslia and Aarsand (2018) have chosen *RPG Maker VX Ace* in their research work "*because it requires little programming or game design experience to create games*". The game design tool interface is in different languages, as English, and French. As written on its website, *RPG Maker VX Ace* is "*simple enough for a child. Powerful enough for a developer*"[20]. So, it can be used by anybody.

[14] https://gamestarmechanic.com/

[15] https://gamestarmechanic.com/

[16] https://gamestarmechanic.com/

[17] https://www.kodugamelab.com/

[18] https://www.rpgmakerweb.com/products/rpg-maker-vx-ace

[19] https://www.rpgmakerweb.com/products/rpg-maker-vx-ace

[20] https://www.rpgmakerweb.com/products/rpg-maker-vx-ace

A character generator allows to customize the available characters in the database of the game design tool. The Map Editor allows to create the world of the game by selecting a tileset and by painting the map with the different tiles (scenes and characters). Tutorials, in English, are available on its website. *RPG Maker VX Ace* costs 64,99 €. However, a trial version of *RPG Maker VX Ace* allow users to try the game design tool, and all its functions, for free for 30 days.

"*RPG Maker VX Ace is a game engine designed to make 2D Roleplaying Games*"[21] (RPG). When a game is developed with *RPG Maker VX Ace*, a .EXE file is created, and the game can be played on Windows.

This tool was used by Øygardslia and Aarsand (2018) where the students worked on their project of game design during two consecutive days. The aim of the research work led by Øygardslia and Aarsand (2018) was to study how artifacts, as textbooks, world maps, Google, and timelines were used when students design games in a classroom.

3.6.8 Scratch[22] (T8)

Scratch is a visual programming environment. So, Baytak, A., Land (2011) have chosen *Scratch* in their research work because "*it entails a user-friendly interface for children with visual code blocks, reducing syntax errors*". "*Users drag and snap command blocks together to build scripts*". The game design interface is in different languages, as English, and French. *Scratch* "*is designed especially for ages 8 to 16*"[23].

Scratch provides sprites for the characters, and backdrops for the scenes. Video tutorials, available on its website, are all in English. Some are dubbed, or subtitled in other languages, as French. *Scratch* is a freeware.

Scratch allows to design 2D games. *Scratch* allows users to develop different types of games: *clicker games*, *platform games*, *maze game*, *pong game*… The games developed with *Scratch* can be played locally, or online once the games have been uploaded on the Scratch website.

This tool was used by Baytak and Land (2011) where the students worked on their project of game design during twenty-one 45-minute sessions, *i.e.* 15 hours, and 45 minutes. The aim of the research work led by Baytak and Land (2011) was to allow the 5th graders to design games about environmental science to learn this topic and to teach it to 2nd graders.

This tool was also used by Puttick and Tucker-Raymond (2018) where the students worked on their project of game design for 6-hour sessions once a day during 4 days, *i.e.* 24 hours. The aim of the research work led by Puttick and Tucker-Raymond (2018) was to allow students to design games to teach others about climate change.

This tool was also used by Dishon and Kafai (2020) where the students worked on their project of game design for nine 2-hour sessions, *i.e.* 18 hours. The aim of the research work led by Dishon and Kafai (2020) was to allow students to design carnival games to examine the relevance of game design to perspective-taking.

This tool was also used by Ke and Im (2014) where the students worked on their project of game design for two 1-hour session a week during 6 weeks, *i.e.* 12 hours. The aim of the research work led by Ke and Im (2014) was to allow students to design games to learn and to teach others about mathematics.

This tool was also used by Weitze (2017) where the students worked on their project of game design for three days. The aim of the research work led by Weitze (2017) was to allow students to design games to learn and to teach their classmates about geography, chemistry, biology, and social studies.

3.6.9 Stagecast Creator[24] (T9)

Stagecast Creator is a visual programming environment using a point and click interface. *Stagecast Creator* is in different language, as English, and French. *Stagecast Creator* is a program that allows children as young as 8 to design their own games. It is "*accessible to novice programmers*" (Denner, Werner, Ortiz, 2012).

Stagecast Creator provides some backgrounds and some sprites. Tutorials, in English, are available in a *Stagecast Creator*'s directory of the computer. A tutorial, in English, available on YouTube, shows how to design a game.[25] A demo version of *Stagecast Creator* allows users to design games during 120 days after the software installation. But the game design tool "*hasn't been available for purchase since 2014*".[26]

Stagecast Creator allows users to design 2D games. *Stagecast Creator* allows user to design different types of games, as action game or adventure

[21] https://dl.degica.com/rpgmakerweb/tutorials/RPGMakerVXAceTutorial1.pdf

[22] https://scratch.mit.edu/

[23] https://scratch.mit.edu/about

[24] http://acypher.com/creator/

[25] https://www.youtube.com/watch?v=C_4A62w-dEI

[26] https://www.reddit.com/r/gamedev/comments/4aqmn9/remember_stagecast_creator_is_it_still_available/

game (Denner, Werner, Ortiz, 2012). Stagecast Creator allows users to save the games developed on a local disk or on the internet.

This tool was used by Denner, Werner, and Ortiz (2012) where the students worked on their project of game design for 14 months: twice a week during the school year, and every day for 3 weeks during the summer.

3.6.10 Unity[27] (T10)

The language that's used in *Unity* is called C#. The game design interface is in English. *Unity* is intended for professionals.

Unity provides an asset store where users can buy characters, music and sound FX, and many more. So, scenes and characters are not for free. A manual is available in different language, as English. French is not available.[28] For personal use, *Unity* is free, provided users do not earn more than $100 thousands in the last 12 months. For professional use, Unity isn't a freeware but a commercial software, whose prizes range from $399 to $2400 yearly, depending on option.

Unity allows users to design both 2D and 3D games. *Unity* allows user to design a wide variety of games. *Unity* allows to export the games in Windows, macOS, Android, iOS, Playstation 3, Playstation Vita, Playstation 4, Xbox 360, Xbox One, Xbox One X, Wii U, Nintendo 3DS, Nintendo Switch…

This tool was used by Çakır *et al.* (2017) where the girls worked on their project of game design for two half days. The aim of the research work led by Akcaoglu and Koehler (2014) was to evaluate the effects of the game design workshop on young girls' attitudes towards Computer Science.

3.6.11 Unreal Engine[29] (T11)

Unreal engine allows users to design their games either using the C++ code, or a blueprint visual scripting. The game design interface is in English. *Unreal engine* is intended for professionals.

Unreal engine provides templates including the main character with some elements of the game background. A tutorial is available in English in the help tab of the game design interface.[30] *Unreal engine* is free. If the game succeeds (i.e., the game generates over $1 million), *Unreal engine* get royalties (5%).

Unreal engine is a 3D creation tool. *Unreal engine* provides users with different game templates (First person, Flying, Puzzle, Rolling, Third person…). *Unreal engine* enables users to develop games to Windows, Playstation 5, Playstation 4, Xbox Series X, Xbox One, Nintendo Switch, macOS, iOS, Android, Linux, HTML5…

3.6.12 VTS Editor[31] (T12)

The programming language used in *VTS Editor* is based on visual command blocks. Users drag and drop command blocks which trigger different actions. The interface of the game design tool is in different languages, as English, and French.

In VTS Editor, different characters for the avatars, and different backgrounds for the scenes are available for free. Others are available for purchase. T*utorials* are available in different languages, as English, and French. *VTS Editor* is not free. Different commercial versions of the game design tool exist, whose prizes range from 499 € (only available for education) to 6228 € (all-inclusive with coaching) yearly.

VTS Editor allows users to animate 3D characters. *VTS Editor* allows users to develop simulation games (Simu games). *VTS Editor* allows to export the games in Windows, macOS, Android, iOS, directly from a web page, or from the VTS Perform platform, or from the VTS Player application, or on a Learning Management System (LMS).

4 DISCUSSION

Table 1 compares the different game design tools identified from the systematic literature review conducted with the PRISMA methodology.

If the game designer doesn't have any knowledge about programming, he will not use *Unity* which uses C#, a programming language using lines of code, and if he would like to use *GameMaker* or *Unreal Engine*, he will choose their visual programming language. Regarding the language of the game design interface, all are in English. Some propose other languages. But if *the* user speaks only French, he will not choose Alice. Indeed, language barrier could make the use of the game design tool difficult. Regarding the tutorials criteria, all game design tool provides tutorials, explaining and facilitating the use of the game design tools. All are in English. Some propose other languages. But if the user speaks only French, he will

[27] https://unity3d.com/get-unity/download

[28] https://docs.unity3d.com/Manual/index.html

[29] https://www.unrealengine.com/en-US/

[30] https://docs.unity3d.com/Manual/index.html

[31] https://seriousfactory.com/en/authoring-software-vts-editor/

Table 1: Comparison of game design tools according to the 9 criteria.

	Programming language	Tool language	Tutorials	Scenes and characters	Game type	Target audience-designer	2D or 3D modelling	Prize	Export
Tool 1	D&D	Multi En/Fr	Multi En	✓	-	Kids	3D	Free	Online
Tool 2	D&D	Multi En	En	✓	-	Anyone	3D	Free	Local
Tool 3	D&D	Multi En/Fr	Multi En/Fr	✓	Multi	-	2D	Free/ Fee	Various
Tool 4	Code/ D&D	Multi En/Fr	Multi En/Fr	✗	Multi	Novice/ Pro	2D	Free/ Fee	Various
Tool 5	D&D	En	En	✓	Multi	7 to 14 years old	2D	Free	Online
Tool 6	Visual Tiles	Multi En/Fr	En	✓	Multi	9 or 10 years old	3D	Free	Online
Tool 7	Point& Click	Multi En/Fr	En	✓	RPG	Anyone	2D	Demo/ Fee	Win
Tool 8	D&D	Multi En/Fr	En	✓	Multi	8 to 16 years old	2D	Free	Local/ Online
Tool 9	Point& Click	Multi En/Fr	En	✓	Multi	8 years old	2D	Demo	Local/ Online
Tool 10	C#	En	Multi En	✓ (Fee)	Multi	Pro	2D/3D	Demo/ Fee	Various
Tool 11	C++/ visual	En	En	✓	Multi	Pro	3D	Free	Various
Tool 12	D&D	Multi En/Fr	Multi En/Fr	✓	Simu games	-	3D	Demo/ Fee	Various

not choose *Agentsheets*, *Alice*, *Gamestar Mechanic*, *Microsoft Kodu*, *RPG Maker VX Ace*, *Stagecast*, *Unity*, or *Unreal Engine*. Here again, language barrier could make the use of these tutorials difficult.

Most of game design tools provides scenes and characters for free. If the cost aspect is not a problem, the user could use *Unity* which proposes an asset store. Indeed, in *Unity*, assets are not free. Regarding *GameMaker*, this game design tool does not provide any asset neither for free, not with charge. With *GameMaker*, the user will have to ask for a game designer character's help or draw his characters by himself, probably increasing design time, and decreasing the quality of the graphics. Some game design tools provide the possibility to develop different game types. However, *RPG Maker VX Ace* is restricted to Roleplaying games; *VTS Editor* is restricted to simulation games. These 2 game design tools could decrease creativity by imposing a single game type. The user will choose the game design tool depending on the game type he would like to develop. Some game design tools are restricted to professionals. This is the case of *Unity* and *Unreal Engine* which use programming languages using lines of code. Some game design tools are available for the general public. Others are friendly enough allowing young people and children to use them. The choice of the game design tool will depend on the target audience-designer. The choice of the game design tool will depend on the age of the game designer. So, some game design tools are suitable for children and teenagers (*Agentsheets*, *Gamestar Mechanic*, *Microsoft Kodu*, *Scratch*, *Stagecast Creator*). Others are reserved for professionals (*Unity*, *Unreal Engine*).

Some game design tools allow users to develop 3D games (*e.g. Unreal Engine*). Some are restricted to 2D games (*e.g. Scratch*). The choice of the game design tool will depend on the look the user would like that the game has. Some game design tools are freeware (*e.g. Scratch*). Others are commercial tools and users have to pay fees (*e.g. RPG Maker VX Ace*). According to the budgetary constraints of the user, the user could use some game design tools, but not others. Regarding the export of the game developed, for some game design tools, the user has to upload his game to the community website of the game (*e.g. Scratch*). For other game design tools, the export possibilities are various: Microsoft, MacOS, iOS, Android, HTML5, console (*e.g. Unity*).

5 CONCLUSION AND PERSPECTIVES

This review of literature presents a current panorama of game design tools for the last ten years based on 9 identified criteria: (1) programming language, (2) tool language, (3) tutorials, (4) scenes and characters, (5) game type, (6) target audience-designer, (7) 2D or 3D modelling, (8) prize, and (9) export. To our knowledge, there is no research work that addresses the literature review of such topic between 2010 and 2020. This study intents to fill in that gap by reviewing the most recent and high-quality academic publications on this topic.

Then, we have conducted a systematic literature review following the PRISMA methodology. Out of an academic database search result pool of 302 publications, 18 papers were reviewed. In addition to studies identified *via* the databases, 8 other references have been identified *via* another method: the interview of a pedagogical engineer. Three references out of eight have been included in the discussion. These 2 methods allow us to identify 12 game design tools.

The discussion concludes that we cannot recommend one game design tool, the choice depends on the needs of users because each user has different needs and constraints. But this systematic literature review can help game designers, pedagogical engineers, teachers, and researchers to identify the game design tool the more suitable for their needs.

This review of literature is inscribed in the context a project where we aim to experiment a game design-based learning activity with nursing students, and to assess the effects of this design activity on students' learning and social flow (Gajewski, El Mawas, and Heutte, 2020). The current challenge is to analyse our nursing students' public in order to choose the most suitable game design tools that we will use in our experimentation. Note that the experimentation will hold in IFsanté, a nursing school in Lille Catholic University in France with 135 second-year students.

REFERENCES

Akcaoglu, M. (2014). Learning problem-solving through making games at the game design and learning summer program. Educational Technology Research and Development 62, pp 583-600.

Akcaoglu, M. (2016). Design and Implementation of the Game-Design and Learning Program. TechTrends 60, pp 114-123.

Akcaoglu, M., & Green, L.S. (2019). Teaching systems thinking through game design. Education Tech Research Dev 67, 1-19. https://doi.org/10.1007/s11423-018-9596-8

Akcaoglu, M., & Koehler, M. J. (2014). Cognitive outcomes from the Game-Design and Learning (GDL) after-school program. Computers & Education 75, pp 72-81.

Allsop, Y. (2016). A reflective study into children's cognition when making computer games. British Journal of Educational Technology 47, pp 665-679.

An, Y.-J. (2016). A case study of educational computer game design by middle school students. Educational Technology, Research and Development 64, pp 555-571.

Basawapatna, A., Koh, K. H., Repenning, A., Webb, D. C., & Marshall, K. S. (2011). Recognizing Computational Thinking Patterns. SIGCSE '11.

Baytak, A., & Land, S. M. (2010). A case study of educational game design by kids and for kids. Procedia Social and Behavioral Sciences 2, pp 5242-5246.

Baytak, A., & Land, S. M. (2011). An investigation of the artifacts and process of constructing computers games about environmental science in a fifth grade classroom. Educational Technology Research and Development 59, pp 765-782.

Bertolini, L. (2018). *Hands-On Game Development without Coding: Create 2D and 3D games with Visual Scripting in Unity*. Packt Publishing Ltd.

Buelin-Biesecker, J., & Wiebe, E. N. (2013). Can Pedagogical Strategies Affect Students' Creativity? Testing a Choice-Based Approach to Design and Problem-Solving in Technology, Design, and Engineering Education. *120th American Society for Engineering Education Annual Conference & Exposition.*

Burke, Q., & Kafai, Y. (2014). Decade of Game Making for Learning: From Tools to Com-munities. Handbook of Digital Games. 689-709.10.1002/9781118796443.ch26.

Çakır, N. A., Gass, A., Foster, A., & Lee, F. J. (2017). Development of a game-design workshop to promote young girls' interest towards computing through identity exploration. Computers & Education 108 115-130.

Denner, J., Werner, L., & Ortiz, E. (2012). Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts? Computers & Education 58, pp 240-249.

De Souza C.S., Garcia A.C.B., Slaviero C., Pinto H., & Repenning A. (2011). Semiotic Traces of Computational Thinking Acquisition. In: Costabile M.F., Dittrich Y., Fischer G., Piccinno A. (eds) End-User Development. IS-EUD 2011. Lecture Notes in Computer Science, vol 6654. S*pringer, Berlin, Heidelberg.* https://doi.org/10.1007/978-3-642-21530-8_13

Dishon, G., & Kafai, Y. B. (2020). Making more of games: Cultivating perspective-taking through game design. Computers & Education 148, 103810.

El Mawas, N., Gilliot, J. M., Garlatti, S., Serrano-Alvarado, P., Skaf-Molli, H., Eneau, J., ... & Pentecouteau, H. (2017). Towards a self-regulated learning in a lifelong learning perspective. In *CSEDU 2017: 9th International Conference on Computer Supported Education-Special Session Lifelong Learning* (Vol. 1, pp. 661-670).

El Mawas, N., & Muntean, C. (2018). Supporting lifelong learning through development of 21 st century skills. In *10th International Conference on Education and New Learning Technologies.*

Gajewski, S., El Mawas, N., & Heutte, J. (2020). Towards a Methodology to Co-design a Learning Game by Nursing Students. In I., Marfisi-Schottman, F., Bellotti, L., Hamon & R., Klemke R. (eds). Games and Learning Alliance. GALA 2020. Lecture Notes in Computer Science, 12517, Springer, pp.273-282, 978-3-030-63463-6. ⟨halshs-03092035⟩

Gajewski, S., El Mawas, N., & Heutte, J. (2021). Vers une méthodologie de co-conception de learning games par les étudiants en soins infirmiers. In 10e Conférence sur les Environnements Informatiques pour l'Apprentissage Humain (pp. 306-311).

Kafai, Y. B. (2006). Playing and making games for learning: Instructionist and constructionist: Perspectives for games studies. Games and Culture Volume 1 Number 1, 36-40.

Ke, F., & Im, T. (2014). A case study on collective cognition and operation in team-based computer game design by middle-school children. International Journal of Technology and Design Education 24, pp 187-201.

Kitchenham, B., & Charters, S. (2007). Guidelines for Performing Systematic Literature Reviews in Software Engineering, Technical Report EBSE 2007-001, Keele University and Durham University Joint Report.

Macklin, C., & Sharp, J. (2016). *Games, Design and Play: A detailed approach to iterative game design*. Addison-Wesley Professional.

Moore, M. (2016). *Basics of game design*. CRC Press.

Øygardslia, K., & Aarsand, P. (2018). "Move over, I will find Jerusalem": Artifacts in game design in classrooms. Learning, Culture and Social Interaction 19, pp 61-73.

Peppler, K. A., & Kafal, Y. B. (2010). Gaming fluencies: Pathways into participatory culture in a community design studio.

Puttick, G., & Tucker-Raymond, E. (2018). Building Systems from Scratch: an Exploratory Study of Students Learning About Climate Change. Journal of Science Education and Technology 27, pp 306-321.

Rasco, A., Chan, J., Peko, G., & Sundaram, D. (2021). Evolution of Serious Games to Support Lifelong Learning and Decision Making: Design and Implementation of a Financial Literacy Game. In *Proceedings of the 54th Hawaii International Conference on System Sciences* (p. 64).

Weitze, C. L. (2017). Reflective, Creative and Computational Thinking Strategies Used When Students Learn Through Making Games. In Proceedings of the 11th European Conference on Game-Based Learning (pp. 744-753). *Academic Conferences and Publishing International.*