University of Glasgow

School of Social & Political Sciences

Report on No-/Low-Code Approaches to Computer Game Development

for the AWARE project of the University of Glasgow Games and Gaming Lab

Karolin Rippich

2673666R

I. Contents

II. List of Tables

III. List of Abbreviations

2

2D
two-dimensional 12

3

3D
three-dimensional 12

A

AI
Artificial Intelligence 6

L

LCDP
low-code development platform 5

M

ML
Machine Learning 6

N

NCDP
no-code development platform 5

R

RPG
role-playing-game 13

1 Introduction

As Anderson et al. noted in 2013, one of the many applications of computer games is digital game-based learning.[1] Corresponding approaches include game-based learning in computer science or computer-supported or -based serious gaming, as often seen in wargaming. Yet, since traditional software development approaches tend to require technical skills and expertise in the field of programming and are characterised by inflexible development processes that make digital game delivery quite resource intensive,[2] they pose a significant obstacle to non-technical game designers interested in digital games.[3] As a result, in the past a developer was often needed to facilitate the creation of computer-supported or computer-based games.

However, according to Beranic et al. (Adoption and Usability of Low-Code/No-Code Development Tools 2020, 97)**,** *“the low-code/no-code development approach is an important concept addressing [such] challenges in the software development domain”*. By offering simplified programming mechanisms that leverage the power of advanced programming languages, relevant platforms empower users without programming skills to create (digital) games of varying complexity. As such, it is no surprise that Gartner predicts that by the year 2024, 80% of all technology products will be built by non-developers using no-code and low-code tools.[4]

[1] Cf. Anderson, Eike Falk, Leigh McLoughlin, Joe Watson, Sam Homles, Peter Jones, Hayden Pallet, and Brendan Smith. 2013. “Choosing the Infrastructure for Entertainment and Serious Computer Games - a Whiteroom Benchmark for Game Engine Selection.” *5th International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES)* (Institute of Electrical and Electronics Engineers). Accessed July 16, 2023. doi:10.1109/VS-GAMES.2013.6624223.

[2] Cf. Sahinaslan, Ender, Onder Sahinaslan, and Mehmet Sabancioglu. 2021. “Low-code application platform in meeting.” *AIP Conference Proceedings 2334, 070007 (2021)* (Maltepe University, Department of Informatics) 1-4. Accessed October 28, 2022. doi:10.1063/5.0042213.

[3] Cf. Bengsch, Geraldine. 2022. “Using Custom-Built, Small-Scale Educational Solutions to Teach Qualitative Research Literacy: No Code, Code, and Complex Applications.” In *Handbook of Research on Acquiring 21st Century Literacy Skills Through Game-Based Learning*, by Carol-Ann Lane, 905-925. London: IGI Global. Accessed July 10, 2023. doi:10.4018/978-1-7998-7271-9.

[4] Cf. Rimol, Meghan. 2021. *Gartner Says the Majority of Technology Products and Services Will Be Built by Professionals Outside of IT by 2024.* 14 June. Accessed July 10, 2023. https://www.gartner.com/en/newsroom/press-releases/2021-06-10-gartner-says-the-majority-of-technology-products-and-services-will-be-built-by-professionals-outside-of-it-by-2024.

2 Game Development for Non-Programmers: The Basics of Low-/No-Code

While the terms low-code and no-code are often used interchangeably, no code development is only one of many low-code development practices.[5] Both low-code/no-code development platforms (LCDPs/NCDPs) allow for the creation of functional software using graphical user interfaces, templates, and visual abstraction instead of coding.[6] This way, they make game development more accessible. While LCDPs allow users to engage with automated code, facilitating larger and more complex game applications, NCDPs tend to be more restrictive by only offering drag-and-drop functionalities with no exposure to code.

One of the downsides of LCDPs/NCDPs is that they are generally application-specific with their own set of templates and configurations, making them less flexible than traditional development tools. Also, they do not facilitate the entirety of the software development life cycle,[7] particularly restricting the maintenance and testing capabilities of users. Consequently, both kinds of tools have their up- and downsides in terms of user friendliness and flexibility compared to more traditional software development approaches. On top of that, a recent KPMG study found that many (potential) users struggle with the lack of transparency surrounding offers of low-code tools.[8]

[5] Cf. Rokis, Karlis, and Marite Kirikova. 2022. “Challenges of Low-Code/No-Code Software Development: A Literature Review.” In *Perspectives in Business Informatics Research*, by Ērika Nazaruka, Kurt Sandkuhl and Ulf Seigerroth, 3-17. Rostock: Springer International Publishing. Accessed July 15, 2023. doi:10.1007/978-3-031-16947-2_1.

[6] Cf. Bengsch 2022, Lee, Erika, Joel Ross, and Jen Kramer. 2020. “Teaching on the Front End: Gathering all Educators Interested in Web and Mobile Design and Development.” *SIGCSE '20: Proceedings of the 51st ACM Technical Symposium on Computer Science Education* 1403. Accessed October 28, 2022. doi:10.1145/3328778.3372512, and Sahinaslan et al. 2021.

[7] Cf. Wang, Hai, and Shouhong Wang. 2022. “Improving Student Performance by Introducing a No-Code Approach: A Course Unit of Decision Support Systems.” *Journal of Information Systems Education* (Journal of Information Systems Education) 33 (2). Accessed October 28, 2022. https://go-gale-com.ezproxy.lib.gla.ac.uk/ps/retrieve.do?tabID=T002&resultListType=RESULT_LIST&searchResultsType=SingleTab&hitCount=1&searchType=AdvancedSearchForm&currentPosition=1&docId=GALE%7CA703631692&docType=Article&sort=RELEVANCE&contentSegment=ZON.

[8] Cf. Gonçalves, Rui, and Bruno Oliveira. 2023. *Shaping digital transformation with low-code platforms. Comprehensive market overview of EMA from a large-scale survey.* KPMG, 37. Accessed July 16, 2023. https://assets.kpmg.com/content/dam/kpmg/cy/pdf/KPMG_Shaping%20digital%20transformation%20with%20low-code%20platforms_BF_sec_cy.pdf.

Still, they do, however, enable '*citizen developers*'[9] to not only define a software's or game's functionalities and requirements but to fully engage in its development process.[10] More so, many LCDPs now offer AI / ML mechanisms that allow for simpler access, configuration, and (re-) training of predictive models – a feature that is often seen beneficial to new game designs.[11]

3 No-Code Game Engines for Non-Programmers

"At the centre of almost every modern game […] lies a 'game engine'" which constitutes the necessary technological base infrastructure for game development.[12] More broadly, a game engine is dedicated software for creating games.[13] While these types of software aim to simplify the game development process,[14] they very much tend to cater for programmers and other technical users, often requiring previous scripting[15] knowledge or a basic understanding of programming languages. However, thanks to the rise of no-code and low code solutions, there have been advancements in the field of game engines in recent years, leading to the development of non-programmer-friendly development platforms for game creation. Such game engines make use of various features to simplify the game development process (cf. table 1).

[9] According to Rokis & Kirikov 2022 citizen developers are the main users of LCDPs/NCDPs and who - while experts in their domain - lack programming knowledge.

[10] Cf. Rokis, Karlis, and Marite Kirikova. 2022. "Challenges of Low-Code/No-Code Software Development: A Literature Review." In Perspectives in Business Informatics Research, by Ērika Nazaruka, Kurt Sandkuhl and Ulf Seigerroth, 3-17. Rostock: Springer International Publishing. Accessed July 15, 2023. doi:10.1007/978-3-031-16947-2_1.

[11] Cf. Gonçalves & Oliveira 2023.

[12] Anderson, et al. 2013.

[13] Cf. Garcia, Josep. 2018. *Best Game Engines (2023). 14 Alternatives For Beginners.* 29 November. Accessed July 15, 2023. https://www.tooltester.com/en/blog/best-game-engine/ and Kelley, Michael. 2016. *No-Code Video Game Development Using Unity and Playmaker.* Edited by A K Peters. P. 4. New York: CRC Press. Accessed October 2022, 2022. doi:https://doi.org/10.1201/b21804.

[14] Cf. Venigalla, Akhila Sri Manasa, and Sridhar Chimalakonda. 2021. "On the comprehension of application programming interface usability in game engines." *Software: Practice and Experience* 51 (8): 1728-1744. Accessed July 12, 2023. doi:10.1002/spe.2985.

[15] Scripting languages integrate components from other languages rather than building new programs from scratch. As such, they are frequently used to enhance component features or attain higher programming levels and faster application development by combining sophisticated data structures with simple syntax to achieve outcomes without requiring considerable system architecture (Ousterhout 1998).

Table 1: Categories of Game Engines Depending on Their Difficulty for Non-Programmers[16]

Categories	***Description***	***Difficulty for non-programmers***	
		Quantitative	**Qualitive**
Full game templates	• Ready-made game. • No coding necessary. • Users make minor changes to details, such as importing artwork or moving sliders.	1	Easy Difficulty
Drag-and-drop	• No coding necessary. • Users choose events & properties from a list.	2	Moderate Difficulty
Visual scripting	• No coding necessary. • Users choose functions that replicate code. • Often used to learn basic coding without doing it yourself.	3	Moderate Difficulty
Coding	Requires experience with prominent or game engine-specific scripting language(s).	4	Hard Difficulty

As a result, certain aspects need consideration when choosing a game engine as a non-programmer, as some games may require more powerful tools and, thus, more dedication and programming skills. These include but are not limited to previous programming skills, the game genre, game performance and the platform(s) used.[17]

For examples of game engines within the aforementioned categories, see tables 2 and 3 (appendix), which divide some of the most popular game engines into those

[16] Cf. Garcia 2018.

[17] Cf. Garcia 2018 and Pavkov, Sanja, Ivona Franković, and Nataša Hoić-Božić. 2017. "Comparison of game engines for serious games." *Proceedings of the International Convention MIPRO 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (Institute of Eletrical and Electronics Engineers) 728-733. Accessed July 10, 2023. doi:10.23919/MIPRO.2017.7973518.

suitable or either 2D or 3D game development, listed in increasing difficulty as discussed above.

4 Other No-Code Approaches to Game-Based Learning

4.1 For Citizen Developers: Artificial Intelligence

More seasoned citizen developers or those interested in expanding their programming skillset might also find Chat GPT's AI capabilities useful as the tool can assist with potential struggles during scripting or coding by proposing lines of code based on natural language processing or by helping the 'developer' to make out errors and bugs. Moreover – given you have a bit more time and a greater technical skillset – the combination of NCDPs/LCDP with Chat GPT holds great potential for digital gamification as it allows for the creation of use case specific chatbots and the like.[18] Given the rapid and widespread adoption of Chat GPT it is no surprise that ample courses on how to come up with no code solutions utilising Chat GPT and the OpenAI API exist.[19] Still, this might mean a steeper learning curve and one must remain vigilant as to whether the tool fills potential knowledge gaps with incorrect information.

4.2 For Complete Beginners: Presentation and Spreadsheet Tools

Apart from game engines and the nexus of no-code and AI, there are other interactive solutions to digital game-based learning through wargaming that do not require any coding. These include *"[…] presentation tools like Microsoft PowerPoint or Apple's Keynote [that] can be used to design engaging guides, workbooks, and resource packs"* [20] in the style of narrative games and spreadsheet tools like Microsoft Excel,[21] which can automate parts of otherwise analogue games. The main benefit of such 'no-

[18] Cf. Goel, Vivek. 2023. *The Convergence of Chat GPT, No-Code and Citizen Development.* 9 June. Accessed July 10, 2023. https://media.licdn.com/dms/document/media/D4D1FAQFXD3MR4SzUWg/feedshare-document-pdf-analyzed/0/1688642783365?e=1689811200&v=beta&t=7SAxLvGZeld8f0K1Z9VuUaLk62rGvwnyynCdsuyx1dk.

[19] Cf. Udemy. 2023. *The Complete 'No-Code' ChatGPT & OpenAI API Course.* June. Accessed July 12, 2023. https://www.udemy.com/course/the-complete-no-code-chatgpt-openai-api-course/.

[20] Anwar, Zakiyah, Muhammad Syahrul Kahar, Rais Dera Pua Rawi, Nurjannah NurJannah, Hermanto Suaib, and Febrianti Rosalina. 2020. "Development of Interactive Video Based Powerpoint Media In Mathematics Learning." *Journal of Educational Science and Technology* 6 (2): 167-177. Accessed July 10, 2023. doi:10.26858/est.v6i2.13179.

[21] Cf. Bengsch 2022.

code approaches' is that they utilise technology most users are already familiar with, significantly decreasing difficulty levels and the time needed to achieve the end-goal. Nevertheless, they aren't as flexible as code-based solutions, often resulting in only partly digitalised or automated games that might at times be rigid and not as visually appealing.[22]

5 Conclusion

As discussed, game-based learning and using serious computer games are but one aspect of using computer (supported) games in educational wargaming. Yet, developing a serious (computer) game, be it fully or only partly digitalised, is a complex endeavour, particularly when the people involved don't have any prior programming or scripting knowledge. With low- and no-code technologies increasingly spreading within the game development community, such 'citizen developers' are nowadays presented with a variety of no-code game engines to simplify their development of computer-based / computer-supported (war)games.

However, when choosing a game engine or other relevant tool, citizen developers should already know what their game will look like and how much time and effort they intend to put into it. This is because different game engines or tools require varying amounts of scripting expertise and dedication depending on the type of game anticipated. The most crucial factors to consider include user-friendliness (drag-and-drop functionality or visual editors) and flexibility, as the options offered by low- and no-code tools or game engines are generally more rigid than traditional (game) development approaches. Another crucial aspect to consider within the context of game-based learning is whether the chosen tool allows for multiplatform export since citizen developers often intend to share the developed games with several institutions, students, and staff. Taking all of this into consideration often makes choosing an appropriate game engine a challenging task.

[22] Cf. Bengsch 2022.

6 Limitations and Outlook

The research for this report has shown that further improvements are needed (1) to allow non-programmers to easily compare game engines without having to understand their technological base and (2) to make game engines more accessible for non-programmers. Still, the report itself highlights that there are viable no-code approaches to computer (supported / based) game development for citizen developers. Given the context of wargaming, future research could focus on testing various no-code game engines and their usability in computer-based wargame developed by non-programmers.

7 Bibliography

Anderson, Eike Falk, Leigh McLoughlin, Joe Watson, Sam Homles, Peter Jones, Hayden Pallet, and Brendan Smith. 2013. “Choosing the Infrastructure for Entertainment and Serious Computer Games - a Whiteroom Benchmark for Game Engine Selection.” *5th International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES)* (Institute of Electrical and Electronics Engineers). Accessed July 16, 2023. doi:10.1109/VS-GAMES.2013.6624223.

Anwar, Zakiyah, Muhammad Syahrul Kahar, Rais Dera Pua Rawi, Nurjannah NurJannah, Hermanto Suaib, and Febrianti Rosalina. 2020. “Development of Interactive Video Based Powerpoint Media In Mathematics Learning.” *Journal of Educational Science and Technology* 6 (2): 167-177. Accessed July 10, 2023. doi:10.26858/est.v6i2.13179.

Bengsch, Geraldine. 2022. “Using Custom-Built, Small-Scale Educational Solutions to Teach Qualitative Research Literacy: No Code, Code, and Complex Applications.” In *Handbook of Research on Acquiring 21st Century Literacy Skills Through Game-Based Learning*, by Carol-Ann Lane, 905-925. London: IGI Global. Accessed July 10, 2023. doi:10.4018/978-1-7998-7271-9.

Beranic, Tina, Patrik Rek, and Marjan Heričko. 2020. “Adoption and Usability of Low-Code/No-Code Development Tools.” *Adoption and Usability of Low-Code/No-Code Development Tools* (Faculty of Electrical Engineering and Computer Science, University of Maribor) 97-102. Accessed October 28, 2022.

Garcia, Josep. 2018. *Best Game Engines (2023). 14 Alternatives For Beginners.* 29 November. Accessed July 15, 2023. https://www.tooltester.com/en/blog/best-game-engine/.

Goel, Vivek. 2023. *The Convergence of Chat GPT, No-Code and Citizen Development.* 9 June. Accessed July 10, 2023. https://media.licdn.com/dms/document/media/D4D1FAQFXD3MR4SzUWg/feedshare-document-pdf-analyzed/0/1688642783365?e=1689811200&v=beta&t=7SAxLvGZeld8f0K1Z9VuUaLk62rGvwnyynCdsuyx1dk.

Gonçalves, Rui, and Bruno Oliveira. 2023. *Shaping digital transformation with low-code platforms. Comprehensive market overview of EMA from a large-scale survey.* KPMG, 37. Accessed July 16, 2023. https://assets.kpmg.com/content/dam/kpmg/cy/pdf/KPMG_Shaping%20digital%20transformation%20with%20low-code%20platforms_BF_sec_cy.pdf.

Kelley, Michael. 2016. *No-Code Video Game Development Using Unity and Playmaker.* Edited by A K Peters. New York: CRC Press. Accessed October 2022, 2022. doi:https://doi.org/10.1201/b21804.

Lee, Erika, Joel Ross, and Jen Kramer. 2020. “Teaching on the Front End: Gathering all Educators Interested in Web and Mobile Design and Development.” *SIGCSE '20: Proceedings of the 51st ACM Technical Symposium on Computer Science Education* 1403. Accessed October 28, 2022. doi:10.1145/3328778.3372512.

Ousterhout, John K. 1998. “Scripting: Higher-Level Programming for the 21st Century.” *Computer* (The Institute of Electrical and Electronics Engineers (IEEE)) 31 (3): 23-30. Accessed July 15, 2023. doi:10.1109/2.660187.

Pavkov, Sanja, Ivona Franković, and Nataša Hoić-Božić. 2017. “Comparison of game engines for serious games.” *Proceedings of the International Convention MIPRO 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (Institute of Eletrical and Electronics Engineers) 728-733. Accessed July 10, 2023. doi:10.23919/MIPRO.2017.7973518.

Rimol, Meghan. 2021. *Gartner Says the Majority of Technology Products and Services Will Be Built by Professionals Outside of IT by 2024.* 14 June. Accessed July 10, 2023. https://www.gartner.com/en/newsroom/press-releases/2021-06-10-gartner-says-the-majority-of-technology-products-and-services-will-be-built-by-professionals-outside-of-it-by-2024.

Rokis, Karlis, and Marite Kirikova. 2022. “Challenges of Low-Code/No-Code Software Development: A Literature Review.” In *Perspectives in Business Informatics Research*, by Ērika Nazaruka, Kurt Sandkuhl and Ulf Seigerroth, 3-17. Rostock: Springer International Publishing. Accessed July 15, 2023. doi:10.1007/978-3-031-16947-2_1.

Sahinaslan, Ender, Onder Sahinaslan, and Mehmet Sabancioglu. 2021. “Low-code application platform in meeting.” *AIP Conference Proceedings 2334, 070007 (2021)* (Maltepe University, Department of Informatics) 1-4. Accessed October 28, 2022. doi:10.1063/5.0042213.

Udemy. 2023. *The Complete 'No-Code' ChatGPT & OpenAI API Course.* June. Accessed July 12, 2023. https://www.udemy.com/course/the-complete-no-code-chatgpt-openai-api-course/.

Venigalla, Akhila Sri Manasa, and Sridhar Chimalakonda. 2021. “On the comprehension of application programming interface usability in game engines.” *Software: Practice and Experience* 51 (8): 1728-1744. Accessed July 12, 2023. doi:10.1002/spe.2985.

Wang, Hai, and Shouhong Wang. 2022. “Improving Student Performance by Introducing a No-Code Approach: A Course Unit of Decision Support Systems.” *Journal of Information Systems Education* (Journal of Information Systems Education) 33 (2). Accessed October 28, 2022. https://go-gale-com.ezproxy.lib.gla.ac.uk/ps/retrieve.do?tabID=T002&resultListType=RESULT_LIST&searchResultsType=SingleTab&hitCount=1&searchType=AdvancedSearchForm&currentPosition=1&docId=GALE%7CA703631692&docType=Article&sort=RELEVANCE&contentSegment=ZON.

8 Appendix

Table 2: Overview of game engines suitable for 2D game development (in increasing difficulty for non-programmers)[23]

Game Engine	***Category***	***Game Type***	***Advantages***	***Disadvantages***
Buildblox	Ready-made templates	2D puzzle and casual games or clone apps	• Utilises a drag-and-drop interface. • Publishes to multiple platforms.	• Restrictive due to templates.
Game Salad	Drag-and-Drop	2D (mobile) games, including arcade, adventure, puzzle, or platformer games	• Has its own physics and ‘rules’ to apply to objects. • Is marketed as educational development tool. • Publishes to multiple platforms and offers good support.	• Limited features.
Clickteam Fusion	Visual editor	2D platforming, survival, and arcade games	• Utilises a drag-and-drop interface. • Publishes to multiple platforms. • Free versions and good support.	• Windows only. • Dated interface.
GameMaker Studio	Simplified code engine	2/3D games, like sidescrollers, point & click, roguelikes, and arcade games	• Drag-and-drop interface. • Facilitates creation of all game assets. • Good tutorials and resources. • Advanced features.	• Requires basic understanding of coding / programming.

[23] Cf. Garcia 2018 and Pavkov, et al. 2017.

Table 3: Overview of game engines suitable for 3D game development (in increasing difficulty for non-programmers)[24]

Game Engine	Main Feature	Game Type	Advantages	Disadvantages
CopperCube 5	Drag-and-Drop	3D survival or multiplayer platforming games	• Easy to use. • Visual and terrain editor. • Publishes to multiple platforms.	• Dated graphics. • Lack of tutorials.
Unity's Adventure Creator	Visual scripting	3D space simulators, RPG, survival, action, adventure, and endless running games	• Built-in no-code solution 'Adventure Creator'. • Good resources and tutorials. • Publishes to multiple platforms. • Good for learning C#.	• Favours coders and familiarity with scripting is essential. • Overwhelming for beginners.
Unreal Engine's Blueprint	Visual scripting	3D adventure, action, RPG, fighting, puzzle, and platforming games	• Built-in no-code solution 'Blueprint'. • Node-based interface. • Publishes to multiple platforms. • Good resources and graphics.	• Using features outside 'Blueprint' is challenging for non-programmers. • Demanding on performance.

[24] Cf. Garcia 2018 and Anderson, et al. 2013, 6.