

Developing decision-making serious games using Ren'Py visual novel engine

Muhammad Hariz Hafizalshah, Aimi Shazwani Ghazali, Shahrul Na'im Sidek

Department of Mechatronics Engineering, Kulliyah of Engineering, International Islamic University Malaysia, Kuala Lumpur, Malaysia

Article Info

Article history:

Received Jan 16, 2024

Revised Jun 26, 2024

Accepted Jul 2, 2024

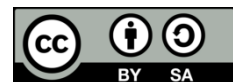
Keywords:

Decision-making game
Game development framework
Python-based scripting
Ren'Py game engine
Serious games

ABSTRACT

Serious games are effective tools defined as games designed with a focus on explicit utility rather than the generally construed notion of games purely as a source of entertainment. Decision-making games are a type of serious game that can be developed with the intent of studying behavior, educating, appraising or other similar applications that benefit through the information collected from the decision-making process. Digital versions of serious games are gaining prominence due to a higher level of interactivity and complexity, especially in Human-Agent Interaction (HAI) applications. The development of digital serious games generally extends beyond software developers, typically involving individuals from diverse backgrounds who may not possess the necessary programming skills required for the development process. The paper proposed the use of Ren'Py, an open-source visual novel game engine as a platform to develop decision-making games. The study examined the Ren'Py game engine's potential through an assessment of the development process for the production of a decision-making serious game. Findings showed that Ren'Py satisfies the need for a relatively easy-to-develop platform for decision-making-based serious games due to its built-in systems that conform to currently applied serious decision-making game design principles.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Aimi Shazwani Ghazali

Department of Mechatronics Engineering, Kulliyah of Engineering, International Islamic University Malaysia
53100 Kuala Lumpur, Malaysia

Email: aimighazali@iium.edu.my

1. INTRODUCTION

Games are a form of entertainment where the goal is for one or more players to experience joy while working towards predefined goals. Serious games are a modification of the gaming medium that makes use of an engaging and immersive setting for more practical purposes. In order to accomplish the main goals of the game design, participants are placed in a virtual environment and given tasks to do [1], [2]. To increase the efficacy of a serious game, Braad *et al.* [3] go into detail about the particular procedures and models that might be taken into account during the game's development and design process. The practice of incorporating theoretical or practical knowledge into the framework of a game has shown a lot of promise and is also known as gamification [4]. A person's daily actions are dictated by the choices made throughout the day. One's decisions could have little bearing on the world or have grave consequences. As a result, thoughtful, reasonable decisions are generally not made instantly but rather through a sequence of actions. The decision-making process can be divided into four stages, as described by earlier research. Of which, the stages are framing, intelligence-gathering, choice, and learning [5]. Turpin and Marais [6] touch on how employees in senior management roles think through choices based on accepted theories of decision-making as well as

how individuals in various roles approach the process to further supplement other factors to the decision-making process.

Multiple facets can be associated with choice. For example, it has been shown that making riskier decisions that could lead to regret or disappointment is associated with negative outcomes like stress and anxiety [7]. It is essential to take note of details such as the specifics of the base origin of regret and disappointment that differ from one another. More specifically, regret is an emotion that arises from feeling that one may have contributed to a negative event, whereas disappointment is an emotion that results from actually experiencing a negative event of which both may be the outcome of the context of the choices one had committed with emphasis on the prior [8].

Examples that peruse serious games include studies by Rebah and Slama [9] and Maheu-Cadotte *et al.* [10] in which they investigated the potential of serious games as an educational tool, with the latter focusing on healthcare professionals and students as a glimpse into the range of applications that serious games could have. Due to limitations imposed by the coronavirus disease 2019 (COVID-19) pandemic, a practical study on a serious game for dentistry instruction [11] has demonstrated the potential of digitally produced serious games by making use of a unique time in human history during a pandemic. The virtual dentistry training participants are required to select the appropriate choices that correspond with the decision-making procedure that would exist in a real clinic. Schönbohm and Zhang [12] also discussed the usefulness of serious games for strategic decision-making while capitalizing on the COVID-19 pandemic-induced recession. Education-based games, such as the one developed by Dallas and Gogoulou [13], used game mechanics to create a more engaging atmosphere and use the educational game “Digiworld” as a virtual learning platform for students learning basic Python programming. To advance in the game, the students use fundamental Python programming concepts.

Smiderle *et al.* [14] goes on to link participants' positive behavioral improvements to exposure to gamified versions of instructional materials. In conjunction with a serious game based on a fire evacuation scenario, Daylamani-Zad *et al.* [15] introduced the applicability of biological sensors as another form of data collecting for specialized applications. Applications like those described by Flogie *et al.* [16] may require a decision-making game to be especially customizable. when unique participants may require a specialized interactive setting.

In recognition of the challenges associated with traditional game development, serious game development is a multidisciplinary field that requires specialists from separate fields from the production of audio and visual assets to the design and programming of the game [17]. A notable overlap in the structure could be observed in commercial game development [18]. The core problem stems from the fact that game development generally requires considerable investment in many aspects including but not limited to the procurement of audio and visual elements along with appropriate technical ability. Further compounding the issue, serious games are commonly developed for a designated scope and application that may include features that necessitate expertise in other relevant fields. Conversely, those in other fields of expertise may not be equipped with all the necessary skills required to develop a full game despite needing to do so. In other words, the creation of serious games may not solely be in the domain of software developers. Individuals with a variety of backgrounds and limited programming experience can also contribute meaningfully to the development process. To combat this issue, development frameworks are developed in this study to ease the serious game development process [19], [20]. These design frameworks are based on specific design applications including but not limited to variations in game mechanics and systems. The value of modest yet effective frameworks is highly desired as they offer an accessible platform for serious game development with minimal investment in game development aspects, allowing for more focus towards the primary topic of serious game application. To the best of our knowledge, no serious game design frameworks that feature Ren'Py have been developed.

Given these shared functionalities, Ren'Py is a game engine that is specifically designed for the applications of developing visual novels. Visual novels consist of minimal game elements with textual components being the most fundamental as a story-driven medium. Visual novels are generally complemented with corresponding visual and audio components with occasional decision-making prompts at certain points of the narrative. Prominent game engines, such as Unreal Engine and Unity, provide a wide range of features for a more extensive availability of game mechanics and systems, which inadvertently raises the general complexity of the development process to account for the inclusion and exclusion of described features. Coincidentally, the majority of the game mechanisms used in the Ren'Py game engine are identical to those used in the creation of more straightforward serious games, which usually include clicking on context-sensitive elements on the game screen [10], [11]. Capitalizing on this information, an outline of the conceptual framework for the development of a serious game was generated using Ren'Py on account of its potential capacity to simulate virtual scenarios and consequently evaluate its overall viability across several aspects. This study also seeks to address the previously mentioned issue of the inaccessibility of a serious decision-making game development platform. Although the Ren'Py game engine has been used as

the development platform for serious games prior such as the study by Zulfikar *et al.* [21], a distinct lack of clarification on the processes was evident.

For enhanced human-agent interaction, the most recent version of Ren'Py is chosen to be employed in the creation of a decision-making game. Ren'Py 8.0.0 is compatible with built-in modules that are presently offered in Python 3.9. Ren'Py is a Python-based application designed to program a visual novel utilizing both specialized syntaxes and the Python programming language. In comparison to popular game engines such as Unity or Unreal Engine, the essential mechanics of a conventional visual novel development may be produced with considerable ease assuming the assets are readily available [22]. Ren'Py has several clear advantages, one of which is the potential use of Python modules that could be initialized in the game script and used to expand the game engine's capabilities. Ren'Py reduces the time and work needed to create a playable game, albeit at the expense of the functionality of other game engines covered in [23]. For games with more complicated features, a supporting tool like Simva [24] can be incorporated into the system during development. A serious game may be considered to achieve a certain quality level to strive for should it fulfil a number of criteria, as summarized in a study by Caserman *et al.* [25]. The following sections describe the processes for assessing the range of functions available in the Ren'Py game engine, the changes to the preset parameters and how the end product may appear.

2. METHOD

The method can be separated into three separate sections. The first section focuses on establishing an understanding of the full capability of the Ren'Py game engine's features and limitations. While the online documentation for Ren'Py game development is comprehensive, the content is primarily focused on visual novel-based games and does not contain detailed information on the means to modify components for the adaptation of the game engine to develop a serious game.

The second section consists of the assessment of game elements for removal or alteration to repurpose the default game elements to match the criteria of the prospective serious game and discuss processes that are not fully covered in the online game documentation. Lastly, the third section applies the outcomes of the prior sections to develop a functional serious game on the Ren'Py game engine. The game was then tested, and the design framework was documented as blank design templates that may be adapted for other serious game designs.

As stated by Braad *et al.* [3], the development process of serious games cannot be generalized to a single model. The context of each development cycle varies depending on the scope and objective of the application of the serious game. However, this element of the design detail is not the major focus of this study. Thus, a simple 10-option decision-making game was drafted and designed to fulfil the third method section.

2.1. Ren'Py features and limitations

Utilizing the stages in the development phase of serious game designs with [26] as a point of reference, the following are descriptions of the features and limitations for each category. The exact categories are game programming, application prototype, quality assurance, specification document and game integration. Game programming denotes the aspects compounded by the architecture, programming language and syntaxes used in the development of the game. Application prototypes are iterations of the game's preliminary game design. Quality assurance is the process in which the occurrence of undesirable issues such as bugs and glitches are minimized or fully removed. Specification documents are conditions or parameters that the developed game adheres to function. Game integration describes the interfacing capability of the game with other components in a system. The exact properties in each category would be described and the specific features and limitations would be noted.

2.2. Default Ren'Py design deviations

Ren'Py is a visual novel engine designed to simplify the creation of story-driven games. However, the intended application of virtual environment simulation of serious games might conflict with the streamlined components of the Ren'Py game engine. The changes made to the standard game elements for setups meant for serious games are explained below.

2.2.1. Disabling GUI elements

The graphical features seen in most visual novels, commonly referred to as graphical user interface (GUI) that appear during gameplay, are shown in Figure 1. The "quick menu" as the name suggests, provides easy access to fundamental visual novel features. To disable the "quick menu", modifications were made to the "quick_menu" object's default value from "True" to "False" in the "screens.rpy" file. Alterations of the files' contents within the root project directory are the main method for Ren'Py game development.

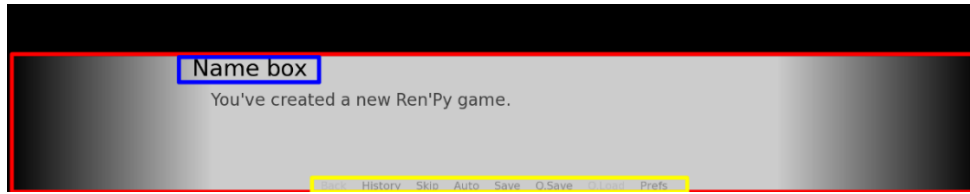


Figure 1. The GUI components, which are the “*namebox*”, “*textbox*”, and “*quick_menu*” are highlighted in blue, red, and yellow, respectively

The narrative depiction designed for visual novels is usually done through the use of “*namebox*” and “*textbox*” elements in the foreground and the accompanying visual elements in the background. The characters' names appear in the “*namebox*”, and the dialogue is shown in the “*textbox*”. The GUI folder contains the PNG file names for the “*namebox*” and “*textbox*” GUI elements along with the corresponding file names. Since the “*namebox*” GUI is translucent by default, it is possible to design the game without altering it. However, it may be necessary to remove the “*textbox*” GUI as it is not transparent by default. Using image editing tools, it is possible to effectively hide the “*textbox*” GUI by adjusting the opacity of the associated “*textbox*” file. Next, the altered file is used in place of the “*textbox.png*” file. As a result, lines with empty quote marks that lack character addressing can be used to make the game require a mouse click every time to advance while no text is displayed.

Each time the game launches, the main menu is the first screen to appear in the game window by default. By adding the “*main_menu*” label to the code, players may start the game at the start label and bypass the main menu section entirely. The game will start at the start label if the “*main_menu*” label is called and then returns.

2.2.2. Disabling player inputs

Ren'Py's default feature, which lets players progress through game states quickly by clicking on the screen quickly, could be problematic in scenarios when the game's pace needs to be maintained at a certain rate or a certain period of duration at specific states. Player activity can be blocked until the allotted time has elapsed by using the “hard pause” option. The mouse's scroll-up wheel may also activate Ren'Py's built-in “rollback” feature, which lets players go back to earlier game states. The pause menu is automatically opened by pressing the right mouse button. By eliminating the “*mousedown_4*” assignment from the text file in the following directory, the key bind for the mouse scroll-up and right mouse button input can be set to null: “*renpy/common/00keymap.rpy*”.

2.2.3. Importing Python modules

The terminal window on the computer, such as the PowerShell or Command Prompt window, can be used to run the module installation command that is listed below. The necessary files for the module's operation will be created in a new folder called “*python-packages*” within the project directory. Note that not every Python package works with Ren'Py, and some adjustments might be needed before the game performs as intended. As an example, the following command can be used to install the “*openpyxl*” module:

```
pip install --target game/python-packages openpyxl.
```

2.2.4. Logging results

Although Ren'Py comes with an inbuilt save system, using it to get a saved variable may incur further issues. Instead, the player's progress and selection options can be immediately exported into a Microsoft Excel spreadsheet by using the permanent variable function and the “*openpyxl*” module. The recorded data may then be processed using this method. An alternative is to use screen recording software to capture the game and then manually note the pertinent points afterwards.

2.3. Design implementation

Once an understanding of how the Ren'Py game engine's overt and underlying concepts were established, a game was developed using both built-in and reconfigurable features. Making changes to the game engine files while maintaining the fundamental Ren'Py functionalities enables the creation of serious games involving decision-making that extends beyond what is normally possible. A decision-making game was developed to study the biological and psychological reactance of the players under stress-inducing situations based on the Ren'Py development concepts adapted for the application. Desired configurations are

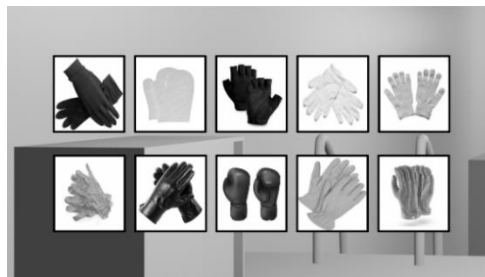
dictated following the defined parameters. The game omits all of the default GUI elements and the game is paced solely through the robot's speech. Player interaction with the game is limited to mouse clicks at certain sections of the game when prompted. The game records the player's choices and the corresponding time at which the choice was made. Integration between the game engine and the robot's text-to-speech functionality requires a separate Python script as a server to be running in conjunction with the game. The "socket" Python module utilizes internal computer networks to communicate between the game and the script. The "socket" module allowed for the circumvention of executing the robot's text-to-speech functionality directly through the game which is an issue as the most current version of Ren'Py is based on Python 3.9, whereas the robot only accepts programming commands of Python 3.10.

The players were subjected to stress-inducing scenarios through the gameplay and robot interaction. Figures 2(a) and 2(b) depict one of the scenarios and the corresponding options presented to the player respectively. Eight unique scenarios were presented for an approximately 35-minute-long game session for each player. The game allowed for the collection of the participants' compliance from the options presented to the players. Every scenario is accompanied by a description of the room to serve as a narration clarification on what each problem entailed. A matter to note is that the choices themselves are inconsequential, for the designed objective of the game is to assess the participants' compliance when suggested to choose an alternative option. Each scenario essentially compounds using previously described development methods to the following, in order and is loosely repeated for the remaining scenario though explicit gameplay:

- The game logs detail at instances where each scenario begins, the choice prompts and when each scenario ends with the "*openpyxl*" python module.
- The player is depicted in a unique location using the "show" statement.
- Details within each location are conveyed through the shift of focus through the "show", "zoom", and "align" statements.
- A custom choice menu is presented to the player twice, using the "*imagebutton*" statements.
- Focus is finally shifted to the end of the scenario location through the "show", "zoom", and "align" statements.



(a)



(b)

Figure 2. Depictions of (a) one of the scenarios and (b) the selection options presented to the participants

The robot notes that a zipline would allow for the participant to cross the hole in the ground but the zipline handle does not offer a proper grip. The participant is then presented with the options so that they may attempt to best progress through the game. Regardless of the first choice the participant makes, the robot will always suggest an alternative as depicted in the flowchart in Figure 3. The other scenarios undergo a similar process but are instead presented with different challenges and options.

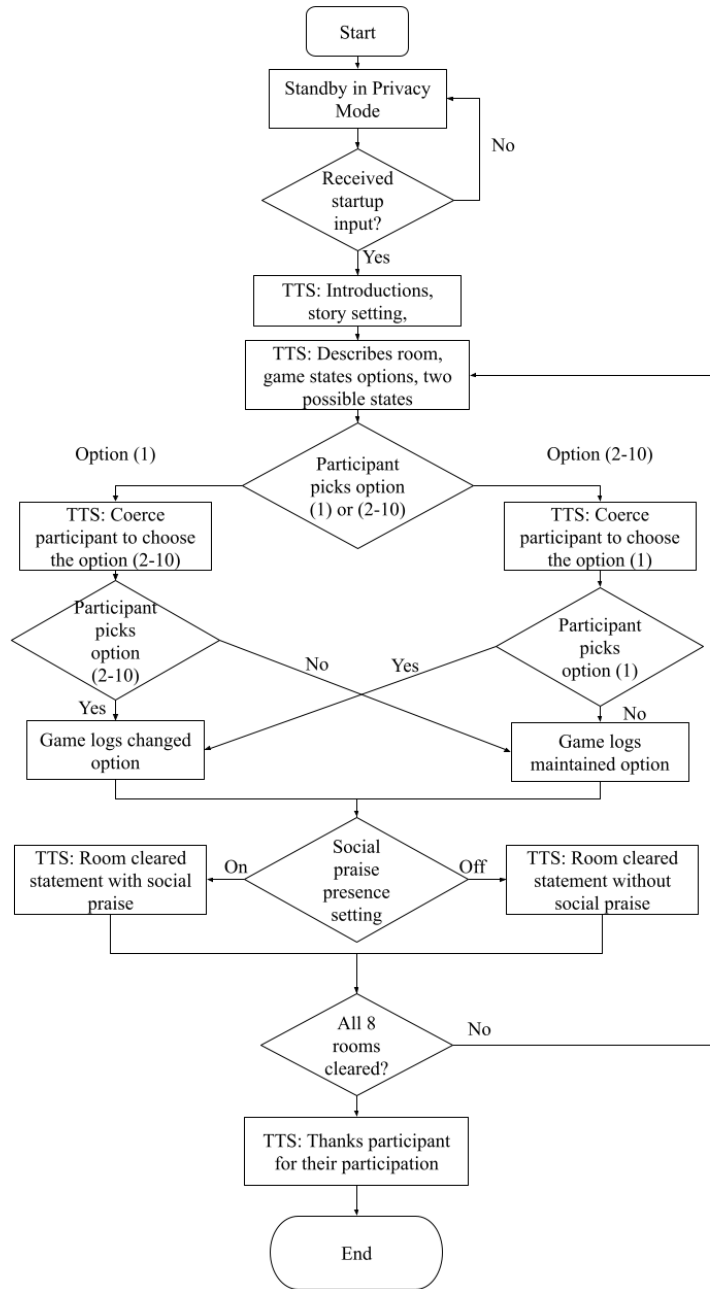


Figure 3. Full game flow a game integrated with a robot’s text-to-speech (TTS) function

3. RESULTS

While Ren'Py has been used to create serious games, the process itself is rarely detailed. This lack of clarity leaves questions about its effectiveness as a serious game development platform. This section aims to extrapolate three main points. First, Ren'Py's suitability as a platform for creating serious games was assessed. Second, we explored how to modify the engine's default elements for optimal serious game design. Finally, we combined these findings to develop a functional game and extract a reusable template.

3.1. Ren'Py features and limitations

For the game programming aspect of the Ren'Py game engine, programming is done using domain-specific language (DSL) which can be described as a custom scripting language that is heavily based on the Python programming language at its core. Somewhat correlating to the matter of game integration, perusing the dollar sign before any line of code would instead cause the game to treat it as Python code which allows for a variety of Python-based modules and functions to work natively in the game. Notably, however, not all Python modules are compatible with Ren'Py. Modules are also typically heavily dependent on operating in

specific Python versions, which becomes a slight detriment as Ren'Py functions using Python 3.9 as a basis. Thereby, version mismatches may become a concern. The DSL scripting language uses simplified syntaxes that simplify the overall programming aspect. A rough prototype could be developed with relative ease due to a straightforward programming method. The “jump” command allows the game’s execution order to move to the labelled line of code, essentially as the means of transitioning between game states. It is vital to remember that utilizing the “jump” command to a specific label skips any code between the jump command that was executed and the point of the referred label.

As a means of quality control, errors are automatically checked upon launch, and any errors found within the code will be reported when the game is launched from the project launcher. If any errors exist, they are displayed in the error window in order of appearance. After resolving the errors, clicking on the “reload” option restarts the game without exiting the window. Additionally, while the game is running, pressing the “shift” and “r” keys on the keyboard at the same time enables the “autoreload” function. The “autoreload” function will reload the game whenever Ren'Py detects changes in the game's associated files.

Bugs are unexpected abnormalities that may emerge throughout development owing to human errors that is unidentifiable by Ren'Py. Bugs can be found when the game behaves differently than expected and are typically reproducible. Monitoring the game while it runs through all conceivable programmed sequences is an effective brute-force method for detecting problems. The usage of labels at key locations in the script may speed up the debugging process by going to relevant labels and skipping sections of the game. clicking the “shift” and “o” buttons on the keyboard launches the built-in console, which allows you to run a variety of commands by typing them in and pressing the enter key. Variables can also be assigned through the console which allows for direct debugging. Game documentation has to be done manually unless a custom code was written to automatically generate one.

A number of key limitations are apparent when developing a game using the Ren'Py game engine. Due to its lightweight property as a software, heavy computations are beyond its base capabilities. Thus, certain integrations would limit its overall performance, or the engine would simply crash. Two-dimensional images traversing on the three-dimensional plane is the extent of the engine’s ability to simulate depth. Three-dimensional renders are inherently unviable.

3.2. Default Ren'Py design deviations

Overextending beyond the standard variable manipulation and code scripting, some alterations were made to the default parameters to achieve the effect of isolating default visual novel game GUI elements from the final design. These changes would present the game in a more controlled fashion in addition to minimizing visual clutter that may distract participants. The result was the game was executed through successive directed measures that inhibited participants from straying from the intended pattern.

3.3. Ren'Py design template framework

Based on the produced template, a serious game can be developed by the contents covered within the paper for other more specific applications. An example of the template block can be seen in Figure 4. Several additional GUI components were included, namely the cursor icon that appears at the bottom right of the screen as a marker for the players to denote interactivity. Additionally, captions appear at the top left of the screen when players hover the cursor over an option. Potential applications may include the study on the efficacy of a serious game before implementation as a means of prototyping or emulating established serious game systems. Design templates are available online [27]. The collection of templates serves as the foundation for the framework that could be adapted into other serious game designs.

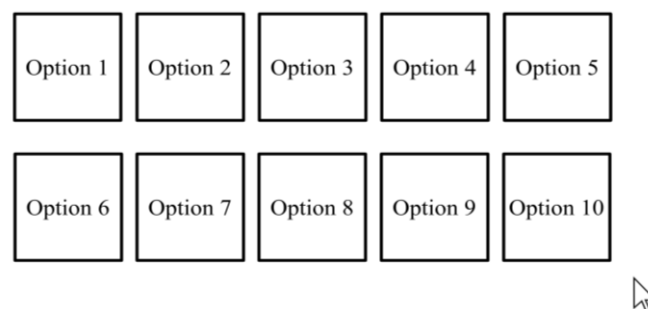


Figure 4. Design template for a 10-choice menu

4. DISCUSSION

Other potential applications also include streamlining the development of serious games for training and education or conducting studies on the cognitive process of making decisions. Immersion is an aspect that plays a large role in the effectiveness of serious games [28]. In this regard, special care must be taken to implement as many details as possible to enhance the experience for a more reliable system.

As described through the development process and case study, a distinct showcase of the extent of the platform's potential to fulfil a particular function is depicted. Through this, developers could attain a better resolution in determining the Ren'Py game engine's suitability for the development of serious games applying desired components which may be within or beyond its capacity to execute. Remodeling a desired game mechanic to function in the game engine is dependent on the complexity of the game mechanic itself and the developer's creativity in adapting it via programming. Regarding complexity, generating simulations of real-life scenarios is a common basis for serious games which can be methodically adapted through proper processes [28], [29]. On that note, some considerations can be considered for what components may be deliberated on to be present for a more effective serious game Breien and Wasson [30].

While comprehensive, the online documentation for development available on the software's website is implicitly directed to those with some background in programming. The included case study project consists of easily identifiable programming statements with the game assets to produce the corresponding outputs. Excerpts of codes can be copied and repurposed for different applications, granted that the developer has a clear understanding of what the code statements or block of codes entail based on the supplied case study project. Ultimately however, recent advancements in Unity and Unreal Engine further showcased innovative tools with a large variety of applications cementing the fact that an objectively more powerful serious game can be designed with the aforementioned game engines as concluded by Gazis *et al.* [31].

As supported by Zainuddin *et al.* [32], the utilization of serious games yields positive results spanning across different age groups and scopes. However, adoption of the technology has not caught up in certain settings [33]. Serious games elevate conventional aspects of life in a more engaging and motivating fashion. Accessibility to a development platform such as the one presented by the paper is critical and should be taken advantage of to further increase the possibilities of varieties of beneficial serious games.

5. CONCLUSION

Throughout the development process, the Ren'Py development platform has demonstrated itself as a solid but lightweight solution for the development of serious games based on the evaluated features, setting alterations and state of function. The use of the Ren'Py platform during the study resulted in almost no performance issues. An integrated system involving human players, the game, and a robot was created, and the system consistently collected data during data collection sessions. For simple serious game concepts, the systems available on the platform fulfill several key criteria. While anyone may replicate a serious game using the available online case study project, Python programming expertise offers an additional contribution to Ren'Py development since it allows for the inclusion of more complex game features. Similar to how Ren'Py was established using Python as a base, another game engine could potentially be developed in the same vein with the settings described in the framework as the default.

ACKNOWLEDGEMENTS

The authors would like to express our gratitude to the Research Management Centre Grant 2020 (RMCG20-037-0037) for making the project possible.




REFERENCES

- [1] N. Kara, "A systematic review of the use of serious games in science education," *Contemporary Educational Technology*, vol. 13, no. 2, Jan. 2021, doi: 10.30935/cedtech/9608.
- [2] C. Vaz de Carvalho, C. S. González González, E. Popescu, and J. Rugelj, "Editorial: serious games," *Frontiers in Computer Science*, vol. 3, May 2021, doi: 10.3389/fcomp.2021.686348.
- [3] E. Braad, G. Žavcer, and A. Sandovar, "Processes and models for serious game design and development," in *Entertainment Computing and Serious Games*, 2016, pp. 92–118.
- [4] C. Reynaldo, R. Christian, H. Hosea, and A. A. S. Gunawan, "Using video games to improve capabilities in decision making and cognitive skill: a literature review," *Procedia Computer Science*, vol. 179, pp. 211–221, 2021, doi: 10.1016/j.procs.2020.12.027.
- [5] P. J. H. Schoemaker and J. E. Russo, "Decision-making," in *The Palgrave Encyclopedia of Strategic Management*, Palgrave Macmillan, 2014.
- [6] S. Turpin and M. Marais, "Decision-making: theory and practice," *ORiON*, vol. 20, no. 2, Dec. 2004, doi: 10.5784/20-2-12.
- [7] K. M. Hengen and G. W. Alpers, "Stress makes the difference: social stress and social anxiety in decision-making under uncertainty," *Frontiers in Psychology*, vol. 12, Feb. 2021, doi: 10.3389/fpsyg.2021.578293.
- [8] M. Zeelenberg, W. W. van Dijk, A. S. R. Manstead, and J. vanr de Pligt, "On bad decisions and disconfirmed expectancies: the psychology of regret and disappointment," *Cognition & Emotion*, vol. 14, no. 4, pp. 521–541, 2000, doi: 10.1080/026999300402781.




- [9] H. Ben Rebah and R. Ben slama, "The educational effectiveness of serious games," *Médiations et médiatisations*, no. 2, pp. 131–155, Nov. 2019, doi: 10.52358/mm.vi2.97.
- [10] M.-A. Maheu-Cadotte *et al.*, "Effectiveness of serious games and impact of design elements on engagement and educational outcomes in healthcare professionals and students: a systematic review and meta-analysis protocol," *BMJ Open*, vol. 8, no. 3, Mar. 2018, doi: 10.1136/bmjopen-2017-019871.
- [11] J.-H. Wu, J.-K. Du, and C.-Y. Lee, "Development and questionnaire-based evaluation of virtual dental clinic: a serious game for training dental students," *Medical Education Online*, vol. 26, no. 1, Jan. 2021, doi: 10.1080/10872981.2021.1983927.
- [12] A. Schönbohm and T. V. Zhang, "Evaluating the effectiveness of serious games in facilitating strategic decisions-making under COVID-19 crisis conditions," *Journal of Work-Applied Management*, vol. 14, no. 2, pp. 257–271, Oct. 2022, doi: 10.1108/JWAM-03-2021-0024.
- [13] O. Dallas and A. Gogoulou, "Learning programming using Python: the case of the DigiWorld educational game," *European Journal of Engineering and Technology Research*, pp. 1–8, Feb. 2022, doi: 10.24018/ejeng.2021.0.CIE.2750.
- [14] R. Smiderle, S. J. Rigo, L. B. Marques, J. A. Peçanha de Miranda Coelho, and P. A. Jaques, "The impact of gamification on students' learning, engagement and behavior based on their personality traits," *Smart Learning Environments*, vol. 7, no. 1, Dec. 2020, doi: 10.1186/s40561-019-0098-x.
- [15] D. Daylamani-Zad, F. Spyridonis, and K. Al-Khafaaji, "A framework and serious game for decision making in stressful situations; a fire evacuation scenario," *International Journal of Human-Computer Studies*, vol. 162, Jun. 2022, doi: 10.1016/j.ijhcs.2022.102790.
- [16] A. Flogie, B. Aberšek, M. Kordigel Aberšek, C. Sik Lanyi, and I. Pesek, "Development and evaluation of intelligent serious games for children with learning difficulties: observational study," *JMIR Serious Games*, vol. 8, no. 2, Apr. 2020, doi: 10.2196/13190.
- [17] H. Engström and P. Backlund, "Serious games design knowledge - experiences from a decade (+) of serious games development," *EAI Endorsed Transactions on Game-Based Learning*, May 2021, doi: 10.4108/eai.27-5-2021.170008.
- [18] B. Berg Marklund, H. Engström, M. Hellkvist, and P. Backlund, "What empirically based research tells us about game development," *The Computer Games Journal*, vol. 8, no. 3–4, pp. 179–198, Dec. 2019, doi: 10.1007/s40869-019-00085-1.
- [19] R. Aziz, H. Norman, N. Nordin, F. N. Wahid, and R. Ibrahim, "Framework development of serious game-based instrument for malay language early literacy assessment," *Creative Education*, vol. 10, no. 12, pp. 3235–3242, 2019, doi: 10.4236/ce.2019.1012247.
- [20] A. M. A. de Araujo Pistono, A. M. P. Santos, and R. J. V. Baptista, "An initial framework for adaptive serious games based on a systematic literature review," in *Communications in computer and information science*, 2022, pp. 43–51.
- [21] W. B. Zulfikar, D. R. Ramdania, A. Afif, and W. Uriawan, "A prototype of serious game for lecturing simulation," *Journal of Physics: Conference Series*, vol. 1402, no. 6, Dec. 2019, doi: 10.1088/1742-6596/1402/6/066103.
- [22] M. Consalvo and D. Staines, "Reading Ren'Py: game engine affordances and design possibilities," *Games and Culture*, vol. 16, no. 6, pp. 762–778, Sep. 2021, doi: 10.1177/1555412020973823.
- [23] K. H. Sharif and S. Yousif Ameen, "Game engines evaluation for serious game development in education," in *2021 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Sep. 2021, pp. 1–6, doi: 10.23919/SoftCOM52868.2021.9559053.
- [24] I. J. Perez-Colado, A. Calvo-Morata, C. Alonso-Fernandez, M. Freire, I. Martinez-Ortiz, and B. Fernandez-Manjon, "Simva: simplifying the scientific validation of serious games," in *2019 IEEE 19th International Conference on Advanced Learning Technologies (ICALT)*, Jul. 2019, pp. 113–115, doi: 10.1109/ICALT.2019.00033.
- [25] P. Caserman *et al.*, "Quality criteria for serious games: serious part, game part, and balance," *JMIR Serious Games*, vol. 8, no. 3, Jul. 2020, doi: 10.2196/19037.
- [26] D. Ávila-Pesántez, L. A. Rivera, and M. S. Alban, "Approaches for serious game design: a systematic literature review," *Computers in education journal*, vol. 8, no. 3, 2017.
- [27] H. Hafizalshah, "Ren-PyShowcase," *GitHub*, <https://github.com/harizhafizalshah/Ren-PyShowcase> (accessed Dec. 05, 2022).
- [28] A. K. Adisusilo and S. Soebandhi, "A review of immersivity in serious game with the purpose of learning media," *International Journal of Applied Science and Engineering*, vol. 18, no. 5, pp. 1–11, 2021, doi: 10.6703/IJASE.202109_18(5).017.
- [29] W. Westera, R. J. Nadolski, H. G. K. Hummel, and I. G. J. H. Wopereis, "Serious games for higher education: a framework for reducing design complexity," *Journal of Computer Assisted Learning*, vol. 24, no. 5, pp. 420–432, Oct. 2008, doi: 10.1111/j.1365-2729.2008.00279.x.
- [30] F. S. Breien and B. Wasson, "Narrative categorization in digital game-based learning: engagement, motivation & learning," *British Journal of Educational Technology*, vol. 52, no. 1, pp. 91–111, Jan. 2021, doi: 10.1111/bjet.13004.
- [31] A. Gazis and E. Katsiri, "Serious games in digital gaming: a comprehensive review of applications, game engines and advancements," *WSEAS Transactions on Computer Research*, vol. 11, pp. 10–22, Mar. 2023, doi: 10.37394/232018.2023.11.2.
- [32] Z. Zainuddin, S. K. W. Chu, M. Shujahat, and C. J. Perera, "The impact of gamification on learning and instruction: a systematic review of empirical evidence," *Educational Research Review*, vol. 30, Jun. 2020, doi: 10.1016/j.edurev.2020.100326.
- [33] K. Larson, "Serious games and gamification in the corporate training environment: a literature review," *TechTrends*, vol. 64, no. 2, pp. 319–328, Mar. 2020, doi: 10.1007/s11528-019-00446-7.

BIOGRAPHIES OF AUTHORS






Muhammad Hariz Hafizalshah    is involved in the development of serious games development and its effects and applications with integration in HRI. One of his main focus is on the establishment of social effects of robotics in daily living. He can be contacted at email: hariz.hafizalshah@live.iium.edu.my.



Aimi Shazwani Ghazali    received a first class Bachelor degree in engineering (Mechatronics) (Honors) from the International Islamic University Malaysia (IIUM, Malaysia) in 2011, a M.Sc degree in mechatronics engineering from the same university in 2015, and a Ph.D. degree in industrial design from the Eindhoven University of Technology (TU/e; Netherlands) in 2019. Her current research interests focus on developing biomechatronic systems and designing social robots for persuasion activities. Her future goal is to aid academic community in developing social robots for a better lifestyle. She can be contacted at email: aimighazali@iium.edu.my.



Shahrul Na'im Sidek    a senior member of IEEE, received his B.E. and Ph.D. degrees from Vanderbilt University, USA, in 1998 and 2008, respectively, in electrical engineering. He is actively engaged in research activities in the field of design and development of electromechanical systems and human-machine interfaces, in particular as it applies to robot-assisted rehabilitation platforms. His research is also focusing on emotion detection using different modalities. To date he has secured a number of research grants amounting to RM2.9 million and has graduated many Ph.D. and M.Sc. students. He can be contacted at email: snaim@iium.edu.my.