Evaluating the feasibility of fast game development using open source tools and AI algorithms

Ioannis Kavouras[1][0000−0003−0448−9459], Ioannis Rallis[1][0000−0003−3876−0024], Anastasios Doulamis[1][0000−0002−0612−5889], and Nikolaos Doulamis[1][0000−0002−4064−8990]

National Technical University of Athens,
Heroon Polytechniou str (15780), Zografou, Athens, Greece
ikavouras@mail.ntua.gr
irallis@central.ntua.gr
adoulam@mail.ntua.gr
ndoulam@mail.ntua.gr

Abstract. Video games designer production is boosted, in recent years, by the evolution in hardware and software domains. Today is easier than ever to use multiple free and open-source softwares, libraries and frameworks for the creation of a game. In this work we describe a simple pipeline for the creation of a hand game emulator, using exclusively free and open-source tools. This is achieved by combining different visualization and rendering tools, such as Blender and Gimp to create simple graphics, as well as python libraries, such as PyQt5, OpenGL and Tensorflow to create the game's interface and AI decision mechanisms. In less than 48 hours, a full featured Rock-Paper-Scissors game has been created. The game supports multiple ways for input, ranging from point and click in the UI, to hand-gestured recognition from RGB cameras.

Keywords: game · development · interface · artificial intelligence · machine learning · graphics · Rock-Paper-Scissors

1 Introduction

Video game production has been extremely versatile in recent years, supported by the evolution of hardware and software resources. Depending on the situation, developed applications vary from indie games, with 8-bit graphical environments, to highly detail extreme in depth flagship titles. For the former case individuals or smaller development teams without the financial and technical support of a large game publisher. For the later case, i.e. "AAA" (triple-A) games, we have plentiful resources dedicated. Regardless of the content type, or the budget invested, most games include some short of AI tools in their core.

The evolution of machine learning and deep learning techniques created multiple opportunities that have a tremendous impact in the game experience. Characteristic examples involve: (a) NPCs who are programmed to mimic human

player behaviour to increase realism [16]; (b) prediction of likelihood of psychological disorders in MOBA games [1]; (c) cheat detection using machine learning techniques [15]; and (d) player emotion detection [17].

In this work we describe how a simple game can be created using open source tools for both the game environment and the AI tools supporting it. Our implementation is a Rock-Paper-Scissors (RPS) game, using PyQt5 library for building the interface, and a pre-trained classification model, serving as the detection mechanism for the player's input. In the following paragraphs, we will explain the motivation behind PyQt5 library, why we choose not to include existing Game Engines, and how Tensorflow based AI tools could be coupled to produce a fast, reliable, low resources game. Thus we focus more on the how-to part, rather than the impact of the outcome.

The rest of this paper is organised as follows: (a) Section 2 provides details related to AI and computer games; (b) Section 3 describes the adopted steps for the deployment of our platform; (c) Section 4 presents how our implemented game can be played; (d) Section 5 concludes this work.

2 Related Work

The game development process has a multidisciplinary nature, which combines (a) sound [13, 10]; (b) art [20]; (c) control systems [6, 4]; (d) artificial intelligence[5, 8], and (e) human factors [2]. Software game is a kind of application that is used not only for entertainment, but also for serious purposes that can be applicable to different domains such as education [9], business [7], and health care[11].

Regardless of the application scenario video games require an appropriate user interface to motivate players. The interface must be intuitive, easy understandable and simple to use. These three factors, can inspire the player to play the game for many hours. Ng et al [19] propose a development of a suitable affecting user-centered design (AUCD) guideline to determine if the expressed emotion, semantics, and mental concept of a tangible and intangible video gaming interface are well received by its intended users.

Machine learning techniques have been widely used in gaming, at the recent years. There are several cases, where AIs are developed to play video games. Justesen et all [12] explained in their review, how deep learning techniques have been applied to play different types of video games, such us first-person shooter [14], arcade games [18] and real-time strategy games [3].

3 Proposed Methodology

The main goal of this work is to demonstrate how easy would be to replicate in a video game environment a hand game, like Rock-Paper-Scissors (RPS), using open source tools and pre-trained AI models. RPS is a simultaneous, zero-sum game, with three possible outcomes: (a) draw, (b) win, or (c) loss. The intriguing

part is that RPS allows the experienced players to recognise and exploit non-random behaviour in opponents. In our case, the user can provide his/her choice, by either using a webcam, uploading an image from the disk or change a value in a combo-box.

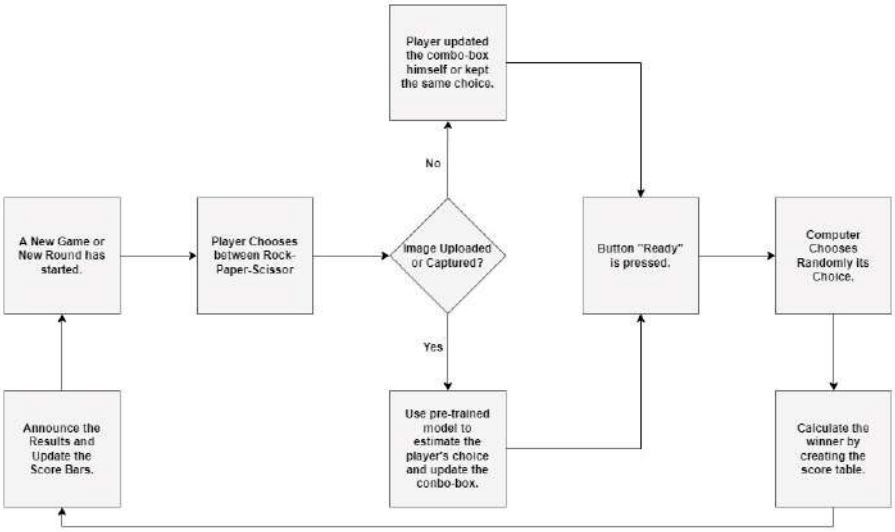

Fig. 1. The proposed pipeline of the RPS application flowchart.

Figure 1 depicts in flowchart the proposed pipeline for the RPS game. The player makes his choice and then press ready. The computer makes a random choice and calculates the winner. The winner is announced and the score-bar updates accordingly. Finally, the next round is started and the player can make a new choice and click ready to play the next round. Further analysis is presented in Section 4.

3.1 User Interface Alternatives

In our work we decided to use PyQt5 to create the User Interface (UI) environment. PyQt5 is a python library developed by Riverbank Computing Ltd, released on April 2016, under licences like Gnu Public Licence (GPL) and commercial use. PySide2 is a different implementation of PyQt5, developed by Qt, released on July 2018, supported only with GPL licence. Other alternative python libraries for user interface are Tkinter or Kivy.

Tkinter is a well-known GUI python library. This library is embedded with Python, thus is installed along python. However, Tkinter framework is a low-level library, which makes it difficult to be used along with other interfaces, like Open Graphical Library (OpenGL). Kivy is an OpenGL framework that comes

in the form of a third-party library. This library contains various features to help developers build robust applications for their system.

A different approach for user interface creation would be to use an existing Game Engine software (e.g. Unreal Engine 4, Unity). However, using a software like these, would made the pipeline complex, because these softwares are usually developed in C, C++ or C# languages. In our case the application is developed completely in python.

3.2 AI Frameworks

Our RPS game implementation uses deep learning models for gesture identification and a rule-based engine for score calculation. First of all a pre-trained model is used to identify player's hand's gesture. The model is based on a Convolutional Neural Network with dense layers (Figure 2), to recognise if user wants rock, paper or scissor. Player's starts a new round by pressing "Ready" button. At that time, the computer choose randomly one of the three options, and compares it with player's selection.

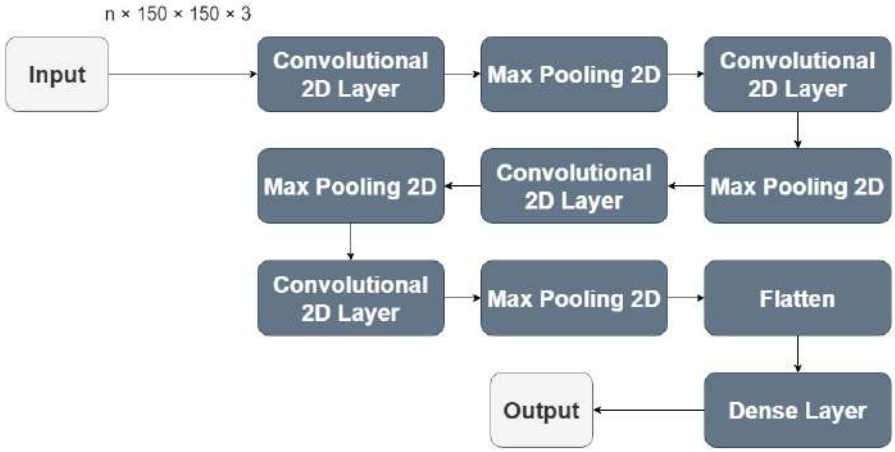

Fig. 2. The pre-trained model's architecture.

The RPS logic, developed as a stand alone class object, i.e. a stand alone library capable to be used in other projects. A significant function of this class is the decision of the winner, if more in case of more than two players. The solution of this problem is the creation of a matrix with zeros in diagonal and either a value of -1, 0 or 1 in cells. A -1 means the player A loses from player B, a 0 means draw and a 1 means the player A wins player B. The final winner(s) decided by the total sum of the scores for each player. The highest score wins. Table 1 indicates an example of a game with three players, were A choose Rock,

B choose Paper and C choose Rock. The winner is B with final score 2 and players A and C draw each other with final score -1.

Table 1. An example of a game with three players, were A choose Rock, B choose Paper and C choose Rock. The winner is B with final score 2 and players A and C draw each other with final score -1.

	A (Rock)	**B (Paper)**	**C (Rock)**	**Total Score**
A (Rock)	0	-1	0	-1
B (Paper)	1	0	1	2
C (Rock)	0	-1	0	-1

3.3 Graphic Design

To visualize the computer's selection we used the following graphical softwares: (a) Blender; (b) GIMP; and (c) Microsoft's Paint. The commonalities of these three softwares can be sum up as following: (a) they are free to use programs, and (b) Blender and GIMP are cross platformed softwares, while Paint is installed along with Windows (on other OSes it can be installed via some kind of Virtual Machine).

Blender is a free and open-source 3D computer graphics software tool-set [21]. Common alternative to Blender includes: (a) Unity (free and commercial); (b) Unreal Engine 4 (free); (c) Cinema 4D (commercial), (d) 3ds Max Design (commercial); (e) Modo (commercial); (f) Maya (commercial); (g) Adobe Premiere Pro (commercial); (h) Adobe After Effects (commercial); and (i) Revit (commercial).

GIMP and Paint are image manipulation programs, which can be used from creating new images/graphics from scratch to editing already existed images/-graphics. Free and non-free alternatives for these softwares includes: (a) Adobe Photoshop (commercial); (b) Canva for Enterprise (commercial); (c) Krita (free); (d) Adobe Photoshop Lightroom (commercial); (e) Photoshop Elements (commercial); (f) Pixlr (free); (g) CorelDRAW (commercial); (h) PicMonkey (commercial); and (i) Fotor Photo Editor (commercial).

3.4 Hardware and Software Requirements

The design of this game was made on the notion that no special hardware requirements should be needed. Thus, the game can work on any system, which supports python 3.7 and above. This makes the application cross-platformed and can run in Windows, Linux and Mac-OS computers. Webcam is required if the player wants to use it. Players without webcam can either upload an image with their choice from a media storage or make a choice from a combo box inside the game.

4 Playing the Game

Fig. 3. Our RPS interface implementation.

Figure 3 depicts the interface of our RPS implementation. In our interface there are two OpenGL widgets for showing images, one for the computer's choice and one to show the captured player's choice. In the bottom left corner exists a combo-box, which corresponds at the choice of the player, which he wants to play for the next round. The player can: (1) either change his option by clicking and selecting one of the three options in the combo box menu, (2) either by enabling the webcam (click on "the Use Camera" checkbox) and then click on "Capture" or press "C" in the keyboard or (3) either by uploading an image with their option ("Upload Image" button). When they are sure for their option, they will press enter and the result will show in the screen. If for any reason they want to change their captured option, they can click on "Reset Capture" button or press the shortcut key "R" in the keyboard.

Figure 4 depicts a draw situation where both computer and player have chosen Scissor. Figure 5 depicts a winning case for the player by choosing Paper against a computer's Rock. Figure 6 depicts a losing case for the player by choosing Rock against a computer's Paper. For all the above cases, the player enabled the webcam and captured an image of his selection. Then he pressed "Ready" and instantly the computer "decided" randomly his selection, calculated the winner by comparing it's choice with the player, as described in Section 3.2, and updated the score bars accordingly. The next round started and the player can either change his choice or he can keep his previous selection.

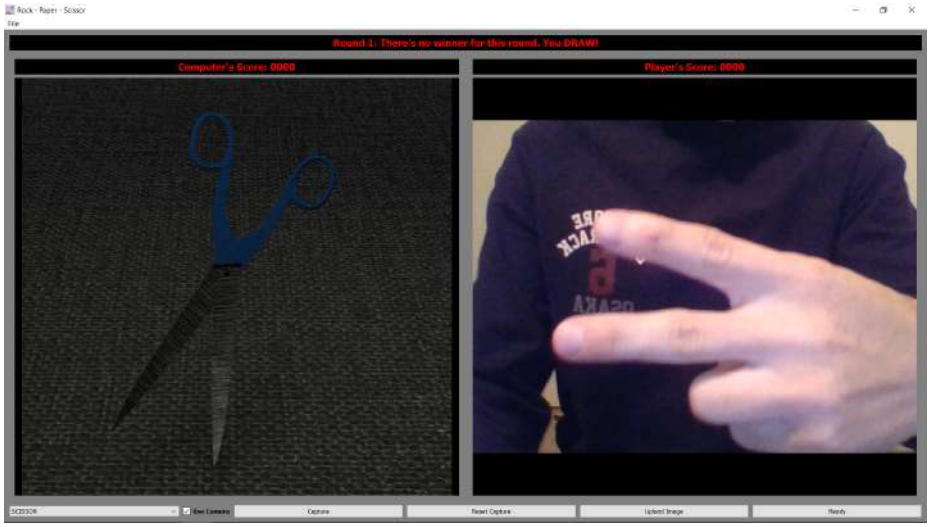

Fig. 4. A draw situation where both computer and player have chosen Scissor.

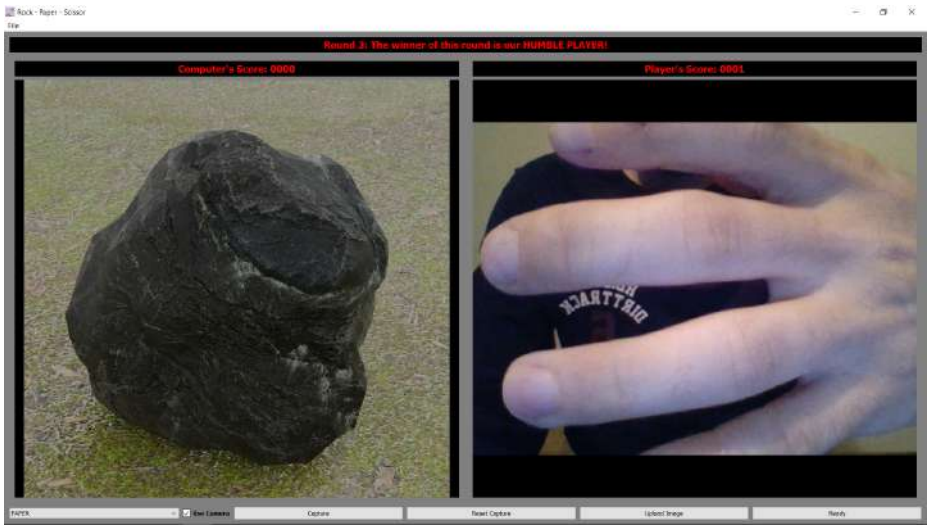

Fig. 5. A win for the player by choosing Paper against a computer's Rock.

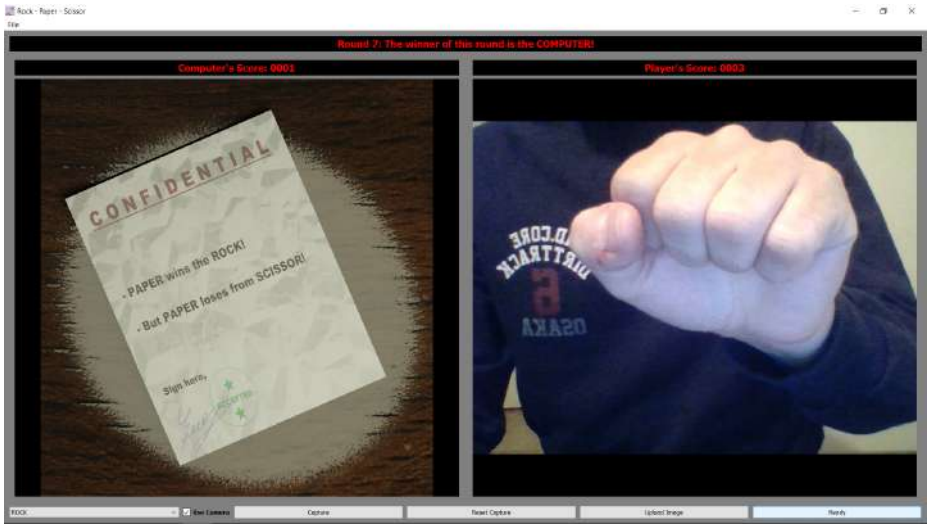

Fig. 6. A loss for the player by choosing Rock against a computer's Paper.

We asked five students to play with the developed game, ten rounds each, using the webcam / deep learning model for player's input. At a 90% rate the input was correctly identified. For the remaining 10% most errors corresponded to misinterpreting paper. On a different test experiment, we asked players to choose their preferred input type. In this case, most of the players at fist preferred to use the webcam as a newly interesting interactive way, however, later on they moved on using the check-box.

5 Conclusions

In this paper we investigated the feasibility of creating from scratch a Rock-Paper-Scissors video game, using free and open-source toolboxes, softwares, features and libraries. The entire concept delivered a working application, capable of recognising hand gestures, without any specific high-end hardware requirements. In terms of setting up and delivering the final program, the time consumed did not exceeded 48 hours in total.

6 Acknowledgement

This paper is supported by the H2020 Marie Curie Programme (MSCA) YADES "Improved Resilience and Sustainable Reconstruction of Cultural Heritage Areas to cope with Climate Change and Other Hazards based on Innovative Algorithms and Modelling Tools" with grant agreement No. 872931.

References

1. Aggarwal, S., Saluja, S., Gambhir, V., Gupta, S., Satia, S.P.S.: Predicting likelihood of psychological disorders in playerunknown's battlegrounds (pubg) players from asian countries using supervised machine learning. Addictive Behaviors **101**, 106132 (2020). https://doi.org/https://doi.org/10.1016/j.addbeh.2019.106132, https://www.sciencedirect.com/science/article/pii/S0306460319303910
2. Aleem, S., Capretz, L.F., Ahmed, F.: Game development software engineering process life cycle: a systematic review. Journal of Software Engineering Research and Development **4**(1), 6 (11 2016). https://doi.org/10.1186/s40411-016-0032-7, https://doi.org/10.1186/s40411-016-0032-7
3. Andersen, P.A., Goodwin, M., Granmo, O.C.: Deep rts: A game environment for deep reinforcement learning in real-time strategy games. In: 2018 IEEE Conference on Computational Intelligence and Games (CIG). pp. 1–8. IEEE, Maastricht, Netherlands (2018). https://doi.org/10.1109/CIG.2018.8490409
4. Baldauf, M., Fröhlich, P., Adegeye, F., Suette, S.: Investigating on-screen gamepad designs for smartphone-controlled video games. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) **12**(1s), 1–21 (2015)
5. Barriga, N.A.: A short introduction to procedural content generation algorithms for videogames. International Journal on Artificial Intelligence Tools **28**(02), 1930001 (2019)
6. Blomberg, J.: The semiotics of the game controller. Game Studies **18**(2), 3 (2018)
7. Buil, I., Catalán, S., Martínez, E.: Encouraging intrinsic motivation in management training: The use of business simulation games. The International Journal of Management Education **17**(2), 162–171 (2019)
8. Casañ-Pitarch, R.: An approach to digital game-based learning: Video-games principles and applications in foreign language learning. Journal of Language Teaching and Research (Online) **9**(6), 1147–1159 (2018)
9. Dankov, Y., Bontchev, B.: Towards a Taxonomy of Instruments for Facilitated Design and Evaluation of Video Games for Education, p. 285–292. Association for Computing Machinery, New York, NY, USA (2020), https://doi.org/10.1145/3407982.3408010
10. Guillen, G., Jylhä, H., Hassan, L.: The Role Sound Plays in Games: A Thematic Literature Study on Immersion, Inclusivity and Accessibility in Game Sound Research, p. 12–20. Association for Computing Machinery, New York, NY, USA (2021), https://doi.org/10.1145/3464327.3464365
11. Halbrook, Y.J., O'Donnell, A.T., Msetfi, R.M.: When and how video games can be good: A review of the positive effects of video games on well-being. Perspectives on Psychological Science **14**(6), 1096–1104 (2019)
12. Justesen, N., Bontrager, P., Togelius, J., Risi, S.: Deep learning for video game playing. IEEE Transactions on Games **12**(1), 1–20 (2020). https://doi.org/10.1109/TG.2019.2896986
13. Kenwright, B.: There's more to sound than meets the ear: Sound in interactive environments. IEEE Computer Graphics and Applications **40**(4), 62–70 (2020). https://doi.org/10.1109/MCG.2020.2996371
14. Khan, A., Naeem, M., Asghar, M.Z., Din, A.U., Khan, A.: Playing first-person shooter games with machine learning techniques and methods using the vizdoom game-ai research platform. Entertainment Computing **34**, 100357 (2020)
15. Kock, E., Sarwari, Y., Russo, N., Johnsson, M.: Identifying cheating behaviour with machine learning. In: 2021 Swedish Artificial Intelligence Society Workshop (SAIS). pp. 1–4 (2021). https://doi.org/10.1109/SAIS53221.2021.9484044

16. Kopel, M., Hajas, T.: Implementing ai for non-player characters in 3d video games. In: Nguyen, N.T., Hoang, D.H., Hong, T.P., Pham, H., Trawiński, B. (eds.) Intelligent Information and Database Systems. pp. 610–619. Springer International Publishing, Cham (2018)
17. Makantasis, K., Liapis, A., Yannakakis, G.N.: From pixels to affect: A study on games and player experience. In: 2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII). pp. 1–7 (2019). https://doi.org/10.1109/ACII.2019.8925493
18. Melnik, A., Fleer, S., Schilling, M., Ritter, H.: Modularization of end-to-end learning: Case study in arcade games (2019)
19. Ng, Y., Khong, C.W., Nathan, R.J.: Evaluating affective user-centered design of video games using qualitative methods. International Journal of Computer Games Technology **2018** (2018)
20. Reed, E.: Exhibition strategies for videogames in art institutions: Blank arcade 2016. Transactions of the Digital Games Research Association **4** (12 2018). https://doi.org/10.26503/todigra.v4i2.91
21. team, B.: Blender's official site (2022), https://www.blender.org/