



An Analysis of Hybrid Game Design: Combining Pac-Man and Xonix

Samyaka Bhavsagar

Department of Computer Science and Engineering

Medi-Caps University

Indore, India

bsamyaka@gmail.com

Prof. Ajeet Singh Rajput

Department of Computer Science and Engineering

Medi-Caps University

Indore, India

ajeetsingh.rajput@medicaps.ac.in

Abstract – this research paper explores the design and development of a hybrid game. It combines elements from the classic arcade games Xonix and Pac-Man. This innovative game retains the simplicity and charm of its inspiration while introducing new dynamics that creates a fresh and engaging experience among players. Unlike Xonix, which typically features only two entities, our game incorporates the diverse and iconic entities from Pac-Man, adding layers of complexity and strategy to the game play. This paper provides a comprehensive analysis of the hybrid game's design, mechanics, algorithms, user experience, and educational applications. Through analysis with the original games, we highlight the unique features and innovations introduced in the hybrid game. The findings demonstrate the potential of hybrid game design to create compelling and dynamic gaming experiences, contributing to the field of game development and interactive entertainment.

Keywords—xonix, pacman, entity, enemy, hybrid game, game development, arcade, maze

INTRODUCTION

Classic arcade games like *Pac-Man* and *Xonix* have left an undeniable imprint on the gaming world, captivating players through their simple yet addictive gameplay. Released in 1980, *Pac-Man* revolutionized the arcade scene with its innovative maze-navigation mechanics. Players are supposed to control the playing entity, Pac-Man, as it navigates through a maze. The Pac-Man entity will consume pellets while avoiding the dangerous group of four colorful enemies. The game becomes accessible to players of all skill levels due to its straightforward controls and fast-paced actions. A different area-capturing mechanism was also introduced by Xonix, which is rather a less well-known gem from the same era. It works by establishing boundaries on an open grid, and the players must cover as much area as they can in this game while dodging the enemy obstacles that might hinder their progress.

Because of the classic appeal of these straightforward

but captivating gameplay, and user-friendly controls, both games *Pac-Man* and *Xonix* have managed to survive over the time. In order to expand upon these timeless foundations important gameplay elements need to be carefully thought out and worked upon. The goals of the player should be well-defined, and elements of Xonix-style territory capture should blend in seamlessly with Pac-Man-style maze navigation. Power-ups might offer momentary benefits like faster speed or invincibility, which could enhance player strategy, while responsive, smooth controls will guarantee a positive user experience. In order to create a dynamic progression that strikes a balance between challenges and enjoyment, the level design ought to incorporate both open territory sections and maze-like structures.

.Enemy entity patterns are thoughtfully designed, to ensure that the game remains engaging, entertaining and challenging.

The appeal from the game will be increased by a consistent visual theme and smooth animations. If the game is further enhanced the players might get a chance to get engrossed by a catchy soundtrack and sound effects. Co-op or competitive multiplayer modes could add depth, and a leaderboard-based scoring system will encourage healthy competition. Multiplayer gaming opportunities could also benefit user interest in the future. Last but not least, creating an emotionally engaging plot and appealing characters will enhance the immersive and enduring experience.

LITERATURE REVIEW

By integrating traditional gameplay elements from games like *Pac-Man* and *Xonix*, the field of hybrid game development offers a great deal of opportunity for innovation. Even though these games have influenced arcade gaming since they first appeared, there is still a lack

of research on how to create a unified hybrid design that skilfully combines their distinctive features. By combining the strategic area-capturing mechanics of Xonix with the fast-paced maze navigation of Pac-Man, our research attempts to narrow this gap and provide players with a balanced and entertaining experience [1][2].

In the 1980s Namco released Pac-Man, which featured a character navigating the maze-like structure while dodging enemies and acquiring items (completely changed maze chase games) [6][7]. On the other hand, the 1980s game Xonix created territorial gameplay [11], in which players have to draw lines while avoiding the enemies to claim territory. Careful consideration of the control schemes, enemy behaviours, and gameplay balance are necessary when combining these distinct mechanics. According to research, hybrid games can effectively blend aspects of several genres to create fresh experiences. To guarantee player interest, it's necessary to have smooth controls, increasing complexity, and a unified visual and aural style at each significant level.

By creating a immersive hybrid gaming experience, this research aims to build on the legacy of these retro-style games [3]. Dynamic gameplay mechanics blend the strategic territory capture with maze navigation and are important components. Players will experience balanced progression via the intricate weaving of challenges from both genres into level designs. A key component of the project is the introduction of sophisticated enemy AI, which allows entities to dynamically adjust to hybrid gameplay mechanics and pose altering challenges. Player engagement will be further increased by imaginative scoring systems, immersive visual styles, and captivating sound design.

This effort also includes contemporary gaming sensibilities, such 3D graphics to enhance player involvement. The game fits a variety of skill levels by customising difficulty levels for each stage, therefore balancing the experience between accessibility and challenge. This creative idea pushes the envelope of retro-inspired gaming while maintaining the nostalgic attraction of vintage arcade games.

Though Pac-Man and Xonix are still well-liked, studies on hybrid designs combining their distinct mechanics is lacking. While Xonix tests players with tactical area-capturing gameplay [4][8], Pac-Man enchants them with its dynamic pursuit-evasion mechanics. Both games provide different experiences. The possibility to combine these components into a coherent hybrid game offers an interesting chance to change player involvement and propel the development of arcade-style games. Furthermore, current research mostly focusses on single-gameplay mechanics; few investigate the artistic blending of several components. This disparity emphasises the significance of creative ideas for hybrid game creation.

This research seeks to close the gap in hybrid game design by investigating the combination of fast-paced maze navigation and territorial strategy [5]. The hybrid game promises to attract players looking for new challenges as well as fans of vintage games. Moreover, it has advantages in education by encouraging strategic thinking, problem

solving and reflex based abilities.

Including AI algorithms controls enemy behaviour and their interaction with player actions, this research paper will offer a thorough examination of the game's design. It will investigate game workflow, characteristics of entities, game levels and their enhancement along with the visuals of the game. The paper will highlight unique qualities and innovations in the hybrid game by means of comparative analysis with Pac-Man and Xonix. This initiative seeks to add to the knowledge of hybrid game development and redefine arcade gameplay for modern audiences by combining the finest qualities of two legendary classics.

METHODS

The different techniques and procedure taken into account while the making of this game is described below in brief.

A. Techniques Acquired

1) Computer System

- a) Processor: Intel Core i3 or a similar processor. Justification: The Intel Core i3 processor was selected due to its affordability, which strikes a balance between cost and performance. It is perfect for development and testing without having to spend a lot of money on expensive hardware because it can run and test medium-complexity video games with ease.
- b) RAM: 4GB or higher. Justification: The system will run smoothly during testing and gameplay if it has 4GB of RAM or more. This amount of memory is adequate to run the game effectively, particularly when it is put through stressful tests with heavy loads. RAM is crucial for preserving a stable experience because it can prevent lag, crashes, or decreased performance.
- c) Graphics Card: GeForce GTX 1050 or higher from NVIDIA. Justification: The GTX 1050 is an affordable graphics card capable of handling the majority of contemporary gaming graphics rendering tasks. It can easily run games at medium settings, mimic real-world situations, and satisfy the hybrid game's graphical requirements. The card was picked because it strikes a balance between cost and functionality.
- d) Operating System: Windows 10. Justification: The popular operating system Windows 10 offers complete compatibility with a large number of development tools. The system is the recommended option for game development environments because it supports compilers like GCC and libraries like SFML.

- 2) SFML Library - Version: 2.5.1. Justification:

Because of its lightweight, effective, and user-friendly features, the SFML (Simple and Fast Multimedia Library) was chosen. SFML makes handling user inputs, managing audio components, and rendering graphics easier. This makes it a great option for quick game development.

- 3) C++ Compiler - GCC (GNU Compiler Collection) or equivalent. 1. Reasoning: GCC is the most recent C++20, and with the most recent C++ there is no reason not using modern language features. The compiler automatically compiles the code in effective mode and allow us to create a hybrid game with the most C++ programming practice.

B. Overview

- 1) Installation and Setup: The first step in making a dependable development environment is to install GCC compiler and SFML library on the system you are working. This configuration was essential because you want to guarantee that it is compatible with all required development tools. After installation, the source code of the game has been downloaded from version control to allow access to the most recent stable version. These strategies encourage a stress-free working environment for everyone and that means everyone was using the best available features and bug fixes.
- 2) Compilation: The game code is compiled with the best compiler flags to guarantee complete compatibility with SFML and the latest C++20 standard. All library dependencies were thoroughly checked up, and include paths were modified if required to fix compilation errors. The procedure also ensures that the compiled code runs across different environments and systems as possible.
- 3) Execution: After successful compilation and several performance tests, the game is launched. The system's performance in overall, and in graphics, were also examined in these tests. Input response latency was measured also to improve user experience and ensure player actions are translated seamlessly into in game reactions.
- 4) Gameplay Testing: A number of gameplay sessions are held to test the fundamental features, such as controls, collision detection, AI behaviour and level progression. Special payers of improving areas and difficulty levels are constantly monitored in terms of the degree of player engagement. The gameplay elements go through this iterative process which improves the whole experience.
- 5) Technical Evaluation: Resource monitoring tools are provided to monitor CPU, RAM and GPU during gameplay. Performance issues were found by simulating heavy workloads through stress tests, but with a small frame rate drop and stability issues. Its was noted and fixes were ranked by what order they would change and then solved.
- 6) Bug Tracking and Reporting: In case of any problems related to testing that includes graphical

glitches and inaccurate collision detection, that is well documented. Bug reports contain clear reproduction instructions as well as resolution priorities. This is an effective bug tracking procedure which guarantees the timely fixes and enhancements.

- 7) Feedback Collection: Qualitative input from team members and playing testers for the overall functionality, aesthetics, and gameplay experience have been obtained. This feedback provided iterative design and development guidance in order to ensure an enjoyable experience and to meet player expectations with the finished product. The feedback process was necessary to improve the mechanics of the game and get it ready for release.

C. Apparatus Design

- 1) Test Computer System: It was created especially to meet the gaming audience's minimal system requirements, guaranteeing that the game is playable and functions well for a wide range of users. This allows users to use similar hardware capabilities such as common gaming system so that people with entry level systems all the way to those with high end setups will able to run the game. It ensures that the game will be accessible to a larger set of players and the game plays out the best.
- 2) Development Tools: We made the choice on SFML (Simple and Fast Multimedia Library) because of its great usability, efficiency, and adaptability in handling essential tasks in fulfilling crucial game development. C++ does not have such an inbuilt feature so it is the perfect choice for this project since it allows smooth rendering of 2D graphics, controls the audio elements, and it handles the input events. Additionally, as GCC (GNU Compiler Collection) can compile cross-platform applications and it is dependably compatible with the C++ 20 standard, the C++ compiler is selecte. This ensures portability and performance by ensuring that the provided game can be used on multiple range operating systems.
- 3) Experimental Conditions: Experimental conditions were controlled so as to obtain reliable, consistent results. The similar way was used to test system stability, gameplay, and performance. Due to all these measures against any outside impediments that would influence the results of our experiment, we ensured that the settings were set up in a manner in which this experiment was conducted while not allowing for any change in the results due to any external entities (remote testing). This strategy enabled a comprehensive and precise evaluation of the game performance in a wide range of situations.
- 4) Feedback from Extensive Play Testing: Character speed, collision tolerance, and enemy entity response time are a few of the gameplay elements that were selected based on feedback from enormous play testing. The aim was to find

the sweet spot between complexity and fun, in order to give a good gaming experience. Also, an average quality default was selected for the graphical settings, in order to reflect what most users with standard system configurations would experience. To sustain performance under room pressures, stressful conditions were tested to the max, repeatably loading heavy assets and stressing with frequent input events.

D. Experimental Conditions

- 1) System Stability: In order to assess the game's overall performance, system stability was continuously observed under both stress and idle conditions. To make sure the game functions properly under various load conditions, resource usage—including, RAM, CPU and GPU the usage was monitored. To find any system slowdowns, crashes, or memory leaks, stress tests were also performed, simulating intense gameplay moments. We could make sure the game maintained steady performance and stability over long gameplay sessions by monitoring how it would react to high performance demands.
- 2) To test long-term stability Gameplay Duration is used each gameplay session was made and engagement to last 10-15 minutes. These sessions gave important information about how the game. It would work over time and therefore it is always to be responsive, engaging and entertaining. And also allowed it to be observed within this time frame. Player behaviour, gameplay patterns, and any slow or performance degradation which includes input lag frame rate drops. This time frame was selected it gives a realistic picture of the overall. It is a representative of gaming experience and is used as such normal gaming sessions.
- 3) Bug Reproduction: Bugs that were reported were run under different system configurations and across different conditions as a means to prove their existence and to determine the extent. Hardware and software setups were tested to find out whether some had a negative effect on some symptoms or whether any of them aggravated certain issues in order to find a compatible configuration that worked best for both hardware and software. To determine whether the problems were repeatable in different environments, this process attempted to reproduce the identical conditions which had reported the bugs. This is also the way it helps in providing a fair understanding of current problems and how to rank the fixes of these problems according to their frequency and severity.
- 4) users of varying levels of gaming experience from casual to expert were among the play testers. This diverse group of people gave us feedback the game's controls, difficulty level, and player engagement. We were able to assure the game would be playable on everyone's hand's skill set. We also analyzed user feedback to create changes toward making more inclusive game and making sure game can enjoy by all kinds of players.

E. Experimental Conditions

- 1) System Stability: To assess game performance, system resource usage was tracked under both stress and idle conditions.
- 2) Gameplay Length: To evaluate long-term stability and engagement, each gameplay session lasted 30 to 45 minutes.
- 3) Bug Reproduction: To verify the existence and extent of reported bugs, tests were conducted across various system configurations.
- 4) Play Tester Demographics: To ensure inclusivity and collect a range of feedback, testers with varying gaming experiences were included.

RESULTS AND DISCUSSIONS

Combining the fast-paced action of moving around the maze and the tactical side of hitting the space and capturing territory, this exciting and dynamic hybrid Pac-Man and Xonix game offers perfect game play. At the beginning of the game, the player plays through a large, complicating setting in which he controls a character. The main goal is to draw around the empty spaces in order to take over as much territory as one can. To define the borders of the open areas and make them into the territory that can be reflected in the game: the players have to very carefully trace the boundaries as they quietly wander the maze. Each successful enclosure increases the players control over the maze and advance them along.

Additionally, the player has to contend with Pac-Man inspired enemies. As you run through the maze, these enemy entities follow you so closely that you tense on a very particular sense of urgency and danger. Any enemy that captures the player will result in losing a life and bringing the player back to the most recent checkpoint. This threat is ongoing so the player is forced to stay alert in order to strike the right balance, of enclosing territory and avoiding the pursuer. The trick is to set boundaries without falling for the captivity of those chasing after you.

The fact that the enemy's behaviour is dynamic makes the game even more difficult. Each enemy in fact moves in their own way, so the players have to adapt their tactics on the run. This unpredictable nature further means that the player has to think fast and try thinking their moves carefully because they are on edge. The player must have skills to navigate and have a good enough reflex in order to not lose progress and to get out from confined spaces.

Pac_nix combines clever, evasive gameplay of Pac Man and territory capturing game play from Xonix, creating an interesting, unique experience. Because players are expected to change with a constantly changing environment of threats and objectives every level feels new and interesting. It is for accuracy, strategical thinking, mastery of offensive and defensive. Being not quite exactly what it seems, and with a perfect blend of known and new mechanics, The game promises hours of fun.

The game progresses once a preset segment of the maze is closed around it. Things get harder as you move past, enemy hits are faster, and the enemies repeat more than once. As territory is enclosed, thus encouraging a greater

amount of territory to be enclosed, players are forced to continually improve upon their tactics to combat the enemies and carry out their objectives. If the player loses all of their lives (according to each level), the game is over, their journey is over. The percentage of territory enclosed and the highest level attained during the game will now be used to determine the overall score. Players are encouraged to get better with every trial thanks to this scoring system, which also makes it more competitive and pushes them to perform better.

Players must grasp the various characteristics in order to properly modify their tactics. Players must learn the strategies to enclose territory while dodging enemy threats in addition to perfecting evasion in order to advance through the game. The gameplay is kept interesting and dynamic by the growing the complexity of enemy behaviour, which forces players to be quick-thinking and constantly modify their strategies in order to survive and thrive. As players progress, the game's overall excitement is increased by the layers of depth and difficulty that this variety in enemy design adds.

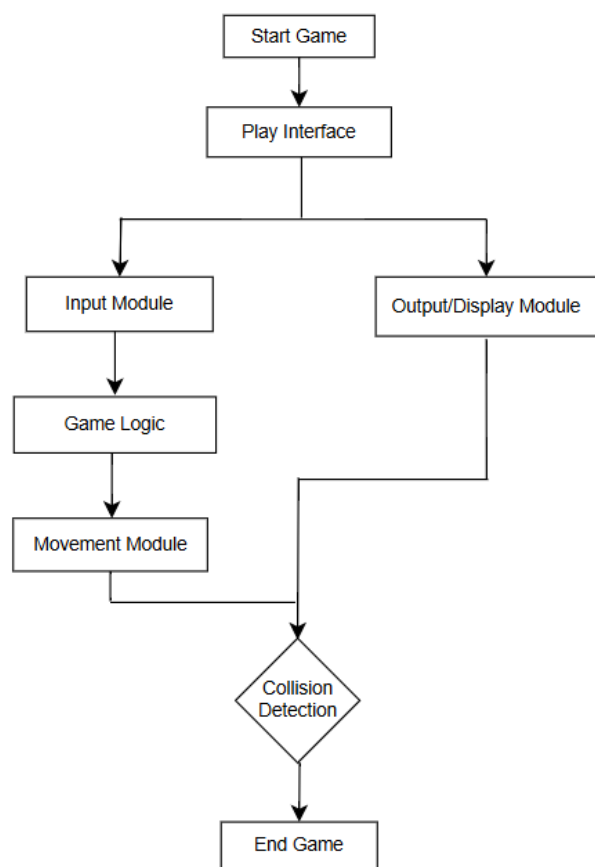


Fig. 1. Program workflow using a Flowchart.

Figure 1 flowchart for the hybrid game outlines several essential modules. The Play Interface focuses on the visual layout, combining maze-like elements with territorial grids. The Input Module handles player controls, such as directional movement like up, down, right and left. The Output Module displays key elements during gameplay, including the player's score, enclosed territories, timer, and lives left of the player. The Game Logic governs the core rules, such as tracking enclosed territories, calculating scores, and defining progression criteria. The Movement

Module illustrates the navigation of both the player's character and enemy adversaries within the game. Lastly,

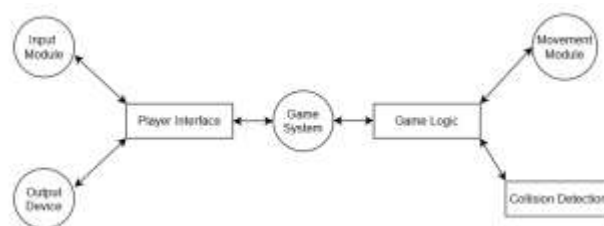


Fig 2. Data flow Diagram

The DFD is shown in Figure 2, where the context diagram's single process node is divided into smaller processes. The diagram will require more data flows and data stores to connect these processes as they are added. This can involve integrating data stores and the room selection and inquiry procedures into the reservation system in the hotel reservation example.

Colour	Name	Aggressiveness	Territory
Yellow	Player	20%	Northwest
Blue	Default Enemy	50%	Northwest
Dark Blue	Destroyer Enemy	70%	Northeast
Green	Hunter Enemy	90%	Center

Table 1 Characteristics of Entities

The distinguishing features of the enemies in the hybrid Pac-Man and Xonix game are highlighted in Table 1. Every enemy has distinct movements and behaviours that have been carefully created to captivate and test players. The way these enemies interact with the player is greatly influenced by their programmed characteristics, which include speed, intelligence, and strategic approaches. While some enemies use aggressive strategies, pursuing the player nonstop, others move erratically or strategically in an attempt to corner or trap the player.

Level	Lives	Enemy		
		Default	Destroyer	Hunter
1	3	1		
2	4	2		
3	4	2	1	
4	2	2	1	
5	2	2	1	1

Table 2 Game Progress

Table 2 provides a comprehensive summary of the enemy's names, their numbers, and the cumulative lives assigned for each level up to level 5. The game progressively increases its difficulty and intensity by introducing an increasing number of enemies up to level 5, each with unique names and a specific number of lives that determine how many times they must be defeated to be removed. This design adds a strategic layer, requiring

players to carefully plan their moves to navigate the maze and counter the growing enemy threats. As the number and lives of enemies grow, their behavior becomes more complex, compelling players to constantly adapt their tactics.

Each level presents more difficulties, necessitating better tactics to avoid and beat the enemies while accomplishing level goals, thanks to the gradual increase in the number of enemies and their overall lives. As players progress through increasingly complex levels, the changing complexity keeps them interested, improves their problem-solving abilities, and increases their sense of achievement.



Fig. 3. Menu Page.

Figure 3 is the Home Page, which serves as the entry point to the game. The title "Pac-Nix" is prominently displayed at the top, inviting players to delve into its world. Below the title, four options are presented: "Start," which allows players to begin a new adventure and explore uncharted territories; "Continue," enabling them to resume their previous progress; "Credits," offering recognition to the developers of the game; and "Exit," providing a choice to temporarily leave the gaming session. This interface sets the stage for an immersive experience.



Fig. 4. Gameplay Window.

Figure 4 players immerse themselves in intricate mazes, reminiscent of the iconic Pacman, while simultaneously attempting to claim territories inspired by the gameplay mechanics of Xonix. These mazes are designed to challenge players, requiring them to think critically and act swiftly to navigate the twists and turns effectively. The integration of territory capture introduces a competitive

element, pushing players to devise strategies that balance offense and defense.

Success in this environment hinges on a seamless combination of sharp reflexes and thoughtful planning, making each decision vital in influencing the outcome. As players advance, the complexity of the mazes and the strategic depth required to conquer territories increase, demanding adaptability and skill.



Fig. 5. Pause Window.

Figure 5, the third window is the pause window which gives players two crucial choices. Players can return to the Home Page by selecting the option, "Back," which gives them a chance to rethink their tactics and get ready for a more powerful comeback. With the second option, "Save and Quit," players can safely end the session and preserve the game progress, allowing them to return to the challenge later without losing their gains. This feature gives players more flexibility and convenience while preserving the continuity of their game's progress, enabling them to efficiently manage their journey.



Fig 6. Game Over Window.

Figure 6 is the final window which shows loss, giving players an option that represents to move back to previous window. The defeated players are asked by a message to click "Back," which will take them back to the Home Page.

Through this research I've learnt a lot including how important it is to optimise performance and take user feedback into account when creating games. During testing, I ran into issues with irregular frame rate drops and excessive system resource usage. These problems made it clear that the game's efficiency was supposed to be improved. Suppose, I want to better control memory usage and optimise how graphics are loaded. By fixing all these issues, the game will run more smoothly and provide players with a smooth and entertaining experience.

Furthermore, although the game mechanics work as intended, some aspects, such as level progression and enemy entities, need to be adjusted. According to player feedback, some levels were either too easy or too hard, which frustrated or disengaged few players. This realisation emphasised how crucial it is to strike the ideal balance in gameplay to appeal to a wide range of players, from a wide range of casual to experts. Maintaining interest and making sure the game stays fair while challenging requires striking this balance.

These findings are crucial because they have a direct impact on player satisfaction and the overall gaming quality. Providing a smooth and entertaining experience is crucial for Pac-Nix to succeed in this fierce competitive gaming market. In addition to drawing players, a top-notch game builds a solid reputation and promotes sustained market success.

I've grown to appreciate feedback more and recognise areas that need work. This ability will be very helpful for my upcoming projects because it allows me to approach development with a growth-oriented and improvement-focused mindset. I am certain that the game will be more ready for release and have a higher chance of success if the problems found during testing are fixed and the conclusions I've learnt are put into practice.

All points considered here, this experience has opened my eyes to the value of technical optimisation, well-balanced gameplay design, and the effectiveness of constructive criticism. In addition to improving this project, these lessons will influence how I approach future endeavours in the field.

A. Applications

- 1) Entertainment: Provides fun and relaxation.
- 2) Skill Development: Improves problem-solving, reflexes, and focus.
- 3) Education: Can include puzzles to teach logical thinking.
- 4) Learning Tool: Demonstrates coding and debugging techniques.
- 5) Prototype: Serves as a base for future commercial projects.
- 6) Portfolio: Showcases technical and creative skills.

- [1] Kuldeep Vayadande, HARshwardhan More, Omkar More, Subham Mu- lay, Atharva Pathak, Vishwam Talnikar, PacMan: Game Developement using PDA and OOP, IRJET, Volume 9, Issue 1, 2022.
- [2] Silva, Adriana de Souza e, Glover, Ragan LeAnn, Hybrid Play, Crossing Boundaries in Game Design Players Identities and Play Spaces, Rout- ledge, 2020.
- [1] Catalin Dima, Mariem Hammami, Yousaf Oualhadj, Regine Laleau, Deciding the synthesis problem for hybrid games through bisimulation, 2024.
- [2] Hybrid Games, Academia.
- [3] Istvan Szita and Andras Lorincz, Learning to Play Mistreatment, Low complexity Rule-based policies, Journal of Computer Science Analysis, Volume 30, pages 659-684, 2002.
- [4] Philipp Rohlfshagen, Jialin Liu, Deigo PerezLiebana, Simon M.Lucas, Pac-Man Conquers Acasemia, IEEE Transactions on Games, Volume 10, Issue 3, 2018.
- [5] Kushal Dey, An Analysis of Gaming Industry in India, IJSR, Volume 10, Issue 5, 2021.
- [6] Namco, Toru Iwatani, Pac-Man
- [7] Anne Ladyem McDivitt, Hot Tubs and Pac-Man and the Early Video Game Industry, Johns Hopkins University Press, Volume 64, Issue 4, 2023
- [8] Artem Novikov, Vladimir Yanovsky, Exploring the limits of maze size simulations and performance, Khmelnytski National University, Volume 341, Issue 5, 2024
- [9] Phan, Angelic, The Rise and Infiltration of Pac-Man and Street Fighter, University Honors Theses, Paper 829, 2020
- [10] Irene Pajarillo-Aquino, The effect of online games to academic perfor- mance of students, IJARMSS, IISN 2278-6236, Volume 8, 2019.
- [11] Ilan Rav, Xonix, DOS Games, Vintage computer games, Action games, 1984.