A Multivocal Review on Derivation Games: A Software Reuse Study

Diego Castro*†, Claudia Werner*
*Program Systems Engineering and Computer Science - COPPE
Federal University of Rio de Janeiro
Rio de Janeiro, Brazil
†Federal Center for Technological Education Celso Suckow da Fonseca (Cefet/RJ)
Rio de Janeiro, Brazil
email: {diegocbcastro, werner}@cos.ufrj.br

***Abstract*—Games have emerged as a prominent form of entertainment, hence establishing the gaming business as a highly lucrative sector. Nevertheless, the process of developing a game can be extremely complex, involving several activities, components, and team members, which can result in an extremely long development period for certain games. The gaming community is engaging in constructing its own games as a response to the delay; its behavior is similar to opportunistic Software Reuse (SR), which is also called modding. The application of mods in game development can provide several benefits, including enhanced longevity of games, reduced production expenses, and accelerated creation of diverse games within reduced timelines. However, the current mod development process lacks a systematic framework or platform that can assist in this progress. Consequently, the aim of this research is to conduct an evaluation that clarifies the primary characteristics, advantages, challenges, and approaches employed in the development of mods, with the goal of comparing these findings with the RS methods that are recommended by the market.**

Keywords-Software Reuse; Game; Mods; Derivation; Structured review; Multivocal review.

I. Introduction

This work is an extension of the paper presented by Castro and Werner [1] at the 2023 IARIA Annual Congress on Frontiers in Science, Technology, Services, and Applications (IARIA), Valencia, Spain.

The evolution of games has been remarkable, establishing itself as one of the most prevalent kinds of entertainment in modern times. This industry generates billions of dollars annually in revenues and investments [2, 3]. As a result of its exponential development over time, this sector has amassed a substantial user community, attracting people of all genders and ages. However, despite the substantial expansion and the extensive fan base, the game development procedure has been exhaustive and deficient in organization, resulting in delays for several titles' release dates [4]. This delay in launching a certain game could cause anxiety and irritation in this community, contributing to search for new games or even their creation of games [5].

By doing a brief search, it is possible to identify several websites that offer modified versions of games. A modification, also referred to as a "mod," involves one or several alterations or adjustments made to a game, which could be related to its mechanics, dynamics, or any other basic element. Its classification may differ contingent upon the degree of modification: these categorizations involve terms such as patches, tweaks, add-ons, and other designations [6, 7, 5]. This method of modifying games can result in a variety of advantages for the company that created the original titles. Among the primary advantages there are: an increase in the number of users, the number of sales, and the longevity of the game [2].

The development of mods carried out by the gaming community can be directly compared to the opportunistic Software Reuse (SR) approach, where a user reuses something previously created to develop new software. Therefore, the objective of this study is to review existing literature to determine the most common ways of mod development currently in use, in order to improve these practices using recommended SR methodologies. As previously mentioned, the gaming community has a substantial user base. Therefore, the search methodology employed in this study aimed to gather material from both formal research sources and informal channels, taking into account the vast size of this community. Due to this, the research method used was Multivocal review which incorporates data from both white (academic papers, books, etc.) and gray (blogs, websites, videos, etc.) literature [8]. This strategy is typically utilized when there is substantial community support for the study subject and it is necessary to verify practical knowledge on a particular subject

The subsequent sections of this paper are outlined as follows: Section II provides a concise overview of the research procedure used in this study. Section III includes a comprehensive analysis of the data encountered during the search process. Finally, Section IV offers a concluding summary of the paper.

II. Theoretical foundation

A. Games

Multiple definitions of games can be found in the literature. Most of them, however, revolve around the following definition: Games can be defined as activities that utilize a conceptual setting where choices, actions, and rules are established in order to engage in a recreational effort, such as entertainment or pleasure [9, 10].

Identifying the defining characteristics of a new game is a more complex task than it may initially seem. According to the previous definition, it is possible to identify some characteristics that differentiate them, including new rules, actions and choices made by players. Therefore, a mod can be identified as a game that has received certain modifications in some of these previously mentioned characteristics [6, 7].

B. Software Reuse

The practice of reuse is widely employed across various industries, such as manufacturing, automotive, and electronics. The term Software Reuse (SR) was first introduced at a NATO conference in 1968 [11] and is one of the disciplines of Software Engineering (SE). Software reuse is the practice of creating new systems by utilizing existing software artifacts and knowledge, rather than building everything from scratch [12].

SR has several areas of study within its field of research, such as: component-based development, Model Driven Development (MDD) and Software Product Line (SPL).

- **SPL:** is a set of strategies, techniques, and tools used to systematically develop similar systems that share a common core but have unique features. The application of these subjects is expected to lead to a reduction in development time, easier maintenance and evolution of systems, increased programmer satisfaction, and improved code quality [12].
- **Software Component:** can be conceptualized as an autonomous and interchangeable piece of code that fulfills a specific function and is reused during the development of a new program. By implementing this approach, several benefits are expected to be achieved, such as the unification of functions, encapsulation, improved code quality, and more agile development [13].
- **MDD:** aims to separate feature design from implementation specification. This technique facilitates software development by utilizing modeling and implementing models of its implementations. The primary objective of this method is to allow developers to concentrate solely on the application's business specifications, rather than the underlying platform on which it will run [14].

III. Research protocol

As mentioned, a Multivocal Review (MR) is a more complete examination of the literature that aims to elicit as much information as possible about a specific subject; hence, it incorporates data from both white (academic papers, books, etc) and gray (blogs, websites, videos, etc) sources. A MR may be separated into two stages: the first stage involves the search for academic knowledge (in this case, a Mapping Literature Review - MLR - was used), and the second stage involves the search for gray literature [8]. MLR is a systematic methodological review of research that investigates and categorizes studies in a specific field of study and presents an overview of a certain subject systematically [15].

In the initial phase of the investigation, four search databases were utilized, following the recommendation of Kitchenham et al. [15]. The search string was executed on the main search engines:

1) **Scopus:** www.scopus.com
2) **ScienceDirect:** www.sciencedirect.com
3) **IEEEXplore:** www.ieeexplore.ieee.org
4) **EI Compendex:** www.engineeringvillage.com

In order to facilitate the execution of this study, a fundamental search string was formulated based on the PICOC framework, which covers the following components: Population, Intervention, Comparison, Outcome, and Context [16]. Combining domain-specific keywords with the logical operator ”OR” and fields with the logical operator ”AND” produced the search string. This string was utilized for the duration of the search. To validate the search string, two control papers (Modding as part of game culture [5], Serious mods: A case for modding in serious games pedagogy [17]) were used to generate and execute the string in the Scopus database, the first database to which the string was applied. This validation technique seeks to ensure the quality of the search string by returning only relevant articles and author knowledge [18].

According to [19] and [20], snowballing processes can compensate for the absence of other search engines and supplement the approach by doing research via the references and citations of the papers. Therefore, to minimize the loss of some papers and increase the search range, the forward and backward (one-level) snowballing procedure was used, which checks the references and citations of articles seeking relevance [21]. The procedures, inclusion and exclusion criteria and quality criteria will be described below. The research questions of each of the reviews will be described in the following sections.

1) **Snowballing Backward**: refers to the identification of new papers based on the works that were referenced in the paper that was analyzed [21];
2) **Snowballing Forward**: refers to the identification of new papers based on the works that referenced the paper that was analyzed [21].

The research execution procedure consisted of the following steps:

1) Execute the search string. For searches in gray literature, it was searched for each search string up to page 10 of google. The search strings were formed by combining the keywords of population and intervention;
2) Apply the inclusion / exclusion criteria based on the title;
3) Apply the inclusion / exclusion criteria based on the abstract;
4) Apply the inclusion / exclusion criteria based on the full text;
5) Apply the quality criteria; Apply snowballing backward;
6) Apply snowballing forward. For searches in gray literature, the snowballing was performed on site references, on links contained within the site.

The inclusion criteria, exclusion criteria, quality criteria, and research questions used in the study were:

Inclusion Criteria:

- Viability Study: The document must be in the context of Mods;

Table I
ANALYSIS OF THE PAPERS ABOUT MUTATORS AND GAMES (MAIN STUDY).

	Scopus		**ScienceDirect**		**IEEEXplore**		**EI Compendex**	
Activity	**Result**	**Number of paper**	**Result**	**Number of paper**	**Result**	**Number of paper**	**Result**	**Number of paper**
First Execution	512 added	512	101 added	101	183 added	183	127 added	127
Number of papers	**1440 papers**							
Repeated Papers	51 withdraw	461	7 withdraw	94	23 withdraw	160	0 withdraw	127
Remove conference	48 withdraw	413	7 withdraw	87	0 withdraw	160	0 withdraw	127
Papers in another language	10 withdraw	403	0 withdraw	87	0 withdraw	160	0 withdraw	127
Number of papers	**1110 papers**							
Remove by title	324 withdraw	78	77 withdraw	10	139 withdraw	21	96 withdraw	36
Number of papers	**207 papers**							
Remove by abstract	59 withdraw	14	10 withdraw	0	18 withdraw	3	20 withdraw	16
Number of papers	**56 papers**							
Papers not found	0 withdraw	14	0 withdraw	0	0 withdraw	3	0 withdraw	16
Remove by quality criteria	2 withdraw	12	0 withdraw	0	0 withdraw	3	6 withdraw	10
Remove by full paper	0 withdraw	12	0 withdraw	0	1 withdraw	2	10 withdraw	0
Extracted Papers	**14 papers**							

Table II
SEARCH STRING OF MUTATOR AND GAMES.

P	*Game*
I	Mutator, variant, mods, modification, conversion, add-on, tweak, modding
C	Not applicable
O	Tools, approach*, method*, ideas, framework*, mechanics, interpretation*
C	Creation, production, development, elaboration, generation, practice*
(*game*) AND (mutator OR variant OR mods OR modification OR conversion OR add-on OR tweak OR modding) AND (tools OR approach* OR method* OR ideas OR framework* OR mechanics OR interpretation*) AND (creation OR production OR development OR elaboration OR generation OR practice)) AND (LIMIT-TO (SUBJAREA, "COMP") OR LIMIT-TO (SUBJAREA , "ENGI"))}	

- The document must be in the context of Games and Software Reuse;
- The document must provide data to answer at least one of the research questions;
- The paper must be written in English.

Exclusion Criteria:

- Conference call;
- Studies that can not be fully accessed;
- tudies that are not in the area of Computer Science or Engineering.

Quality Criteria:

The quality criteria employed are derived from Lincoln and Guba, with the objective to evaluate the author's credibility, the transferability of ideas to the new paper, the reliability of the information, and the confirmability of the information [22].

- Is the publishing organization reputable?
- Has the author published another work in the area?
- Does the author have expertise in the area?

- Is the article clear?
- Are the references documented?
- Does this enrich the research?

Research Questions:

- **Q1:** What modifiers are used to create games from other games?
- **Q2:** What characteristics are needed to derive a game?
- **Q3:** What are the advantages and difficulties of creating games from others?
- **Q4:** What tools strategy or frameworks support these changes?

The first stage returned a total of 923 papers. When the publications were examined using the inclusion and exclusion criteria, this number was reduced to 14. From these studies, the snowballing process was carried out, and a total of 245 more papers were evaluated. After this approach, 9 more papers were included, totaling 23 papers read and assessed. Tables II, I and III show the search string and the papers' analysis process. Figure 1 demonstrates the steps that were taken when performing the search. It is worth remembering that the paper analyses were broken into two tables, one for the main study and another for the snowball processing.

Based on the findings of the initial phase, it was determined that the gaming community is quite active in terms of development, enhancements, and modifications. Consequently, a new phase was introduced to the study. In addition to the investigation, a search for gray literature was conducted.

The gray literature search encompassed up to page 10 of Google for each of the search keywords, resulting in 700 links that required validation. The inclusion and exclusion criteria were implemented after visiting each link, resulting in the selection of 21 links for the quality criteria step. Ten links were selected and approved based on the following criteria. The snowball effect was achieved by utilizing backlinks (website reference connections). As a consequence, the entire procedure was restarted for the authorized connections, and 335 additional links were validated. Lastly, 12 additional documents were added in the search at the snowballing process. The papers that were chosen for this research are listed in Table IV, along with the questions that each document may answer.

IV. Results analysis

The rise of the mod trend is closely related to increased accessibility to personal computers and the expansion of the internet, which is disseminating an increasing amount of content [23]. The community and academy are increasingly generating game adaptations, assisting game producers in a variety of ways, including recruiting new players, extending the life of a game, providing new views for the game, and resolving bugs. Modifications, in general, are referred to as mods and may be thought of as modifications to an original game [6].

Modifications to products in the gaming industry done by gamers are now often referred to as modding. Modders employ a variety of strategies in their creations, ranging from basic rearranging of game world parts to complete conversions that can be somewhat independent of the original game [23]. This section will discuss the many sorts of modifiers discovered, their benefits and drawbacks, and lastly, the essential criteria for constructing an adapted game, as well as if there are tools available to aid in this process.

Table III
Analysis of the papers about mutators and games (Snowballing Study).

	Snowballing Backward		**Snowballing Forward**	
Activity	**Result**	**Number of paper**	**Result**	**Number of paper**
First Execution	381 added	381	136 added	136
Number of papers	**1440 papers**			
Repeated Papers	141 withdraw	240	20 withdraw	116
Remove conference	0 withdraw	240	0 withdraw	116
Papers in another language	4 withdraw	234	3 withdraw	113
Number of papers	**1110 papers**			
Remove by title	172 withdraw	62	86 withdraw	27
Number of papers	**207 papers**			
Remove by abstract	40 withdraw	22	18 withdraw	9
Number of papers	**56 papers**			
Papers not found	0 withdraw	22	0 withdraw	9
Remove by quality criteria	7 withdraw	15	0 withdraw	9
Remove by full paper	11 withdraw	4	4 withdraw	5
Extracted Papers	**9 papers**			

Q1: What modifiers are used to create games from other games?

Increased accessibility to personal computers and the expansion of the Internet, which is disseminating an increasing quantity of content, are closely related to the rise of the mod trend [23]. The community and academy are increasingly generating game adaptations, which help game developers in a variety of ways, such as recruiting new players, prolonging the life of a game, providing new perspectives for the game, and fixing bugs. In general, modifications are referred to as mods and can be viewed as alterations to the original game [6]. In general, a mod is an original game that has had one or N alterations or modifications made to its mechanics, dynamics, rules, or some of its components [4].

Mods are as diverse as the games themselves. They vary in size and complexity and can make minor adjustments to the original game or completely alter its visual design

Table IV
TRACEABILITY MATRIX OF MUTATORS AND GAMES.

Title	Year	Q1	Q2	Q3	Q4
First Stage - White Literature					
Building the Perfect Game – An Empirical Study of Game Modifications [2]	2020	X	X	X	X
To mod or not to mod—an empirical study on game modding as customer value co-creation [4]	2020		X	X	
Modding tabletop games for education [24]	2019	X	X	X	X
Migrating Java-based apo-games into a composition-based software product line [25]	2019		X	X	X
Product line architecture recovery with outlier filtering in software families: the Apo-Games case study [26]	2019	X			X
Apo-games-a case study for reverse engineering variability from cloned Java variants [27]	2018			X	X
Multi-objective optimization for reverse engineering of apo-games feature models [28]	2018		X	X	X
Visual and computational modelling of minority games [29]	2017	X	X		X
Placing value on community co-creations: A study of a video game 'modding' community [30]	2017		X	X	X
Analysis of popularity of game mods: A case study [31]	2016		X	X	
Serious mods: A case for modding in serious games pedagogy [17]	2016	X		X	
Design of a math learning game using a Minecraft mod [32]	2014	X	X	X	X
Applying exception handling patterns for user interface customization in software games modification [33]	2013	X	X	X	X
An environment to support collaborative learning by modding [34]	2014		X	X	X
Reporting about the Mod software process [35]	2012		X		X
A Role-Playing Game for a Software Engineering Lab: Developing a Product Line [35]	2012		X		X
Remix and play: Lessons from rul ts in texas hold'em and halo 2 [36]	2012		X	X	X
Modding as part of game culture [5]	2012	X		X	X
Utilizing a 3D game engine to develop a virtual design review system [37]	2011	X		X	X
Modding as an open source approach to extending computer game systems [7]	2011	X		X	X
When the game is not enough: Motivations and practices among computer game modding culture [38]	2010	X	X	X	
Modding as a basis for developing game systems [6]	2011	X		X	X
Of mods and modders: Chasing down the value of fan-based digital game modifications [39]	2009	X		X	X
Am I Mod or Not? - an Analysis of First Person Shooter Modification Culture [40]	2005	X	X	X	X
Second Stage - Gray Literature					
Unofficial patch [41]	2021	X		X	X
Mod (video gaming) [42]	2021	X			X
Appropriation & Motivation in Game Modification [43]	2020	X	X		X
Video game conversion [44]	2020	X			
Players as Content Creators the Benefits of Game Modding According to Polish Users. [45]	2019	X	X		X
Mod (video games) [46]	2018	X	X	X	X
Understanding Game Modding through Phases of Mod Development [47]	2015	X	X	X	
Does game modding require programming? [48]	2015	X			X
Computer game modders' motivations and sense of community: A mixed-methods approach [49]	2014	X		X	
Game Mods: Design, Theory and Criticism [50]	2013	X	X		
Computer game mods, modders, modding, and the mod scene [51]	2010	X	X	X	X
On modder labour, commodification of play, and mod competitions [48]	2007	X	X	X	X

[40]. Modding is the process and technique of modifying or adapting video games. It is frequently a "Do It Yourself" (DIY) strategy that teaches social and technical skills affiliated with innovation by reusing the concept of an existing game, which can be compared to opportunistic Software Reuse [27, 12]. Numerous aspects of the game, including the user interface, game items, bug fixes, characters, and regulations, are modifiable [2]. By altering the rules of a game, for instance, players are able to construct a unique gaming experience [36, 52].

Developing mods is possible by applying mutators to a game. A mutator is a modification to an existing game; for instance, applying mutator M to game G results in the creation of a new game named G [M] [53]. Depending on the number of mutators utilized, a game may be classified in a variety of ways. There are numerous adaptations and modifications, each serving a distinct purpose [2, 6]. Each of them will be described in more detail in the following [1-3][8][16-27].

1) **Interface customization:** The interfaces are designed to emphasize the visual component of the game in order to enhance the experience. This customization entails making changes to the visual element, such as remodeling the accessories, skin, shader, or animation of a character or a game map, altering the game's colors, or altering the information displayed on the screen.

2) **Partial Conversions:** Add a new map, a new character, and a new item; increase the game's pace; add small mechanisms, bots, and rules. It is still possible to classify partial modifiers according to the modifications they execute.
    a) **Mutators/tweaks:** Modify or add restricted features that have no effect on the game's functionality or mechanics. They may include modifying the game's theme song, increasing the game's speed, or modifying some graphic elements and minor rules.
    b) **Add-ons:** They serve as supplementary elements within the game, performing minor adjustments such as modifying the theme's music, accelerating the game, or adjusting minor graphical components and rules.
    c) **Mods:** They are the intersection of the previous two, as they retain the capacity to change rules and configurations.
3) **Total Conversions:** Certain changes are so drastic that they result in the creation of new games. A well-known conversion is the CounterStrike mod, which was based on Half-Life. In general, the number of modifiers used differentiates a partial conversion from a complete conversion. When a significant number of modifiers are applied to the point where something new is generated, a complete conversion occurs.
4) **Machinima:** It could be seen as the outcome of changes that influence the visual replay of game usage sessions. In this type of modification, games are used for other purposes, such as telling a story, making a movie, or replicating a gaming experience.
5) **Patch:** They frequently concentrate on addressing unresolved problems and creating technical enhancements. This modification is known as an unofficial or fan patch when it is created by a community.

Q2: What characteristics are needed to derive a game?

A game is a type of software development in which designers, developers, and software engineers work together to create an experience for players to live through the game [30]. Once the game is out, the contributors devote their time to updating and adding content to the main game. Modifications may include new game models, textures, music, and mechanisms, as well as complete remakes [30].

There are two primary methodologies for mod development. The first scenario occurs when there is a need for expansion in a particular game by introducing new elements, while the second scenario occurs while seeking games that offer similar characteristics to those wanted in the game under development [24]. Both need the same characteristics.

A game is made up of components that work together to generate the final output. The required qualities for their construction can be determined by defining games. Games are activities that occur in an abstract environment where decisions, actions, and rules are developed with the objective of accomplishing a leisure activity in the form of entertainment or amusement [54]. On this premise, the following aspects must be decided prior to the construction of any game: rules, actions, behaviors, objective, game loop, difficulty, and rewards [24, 9].

Each of the characteristics necessary for the interpretation and evolution of a game will be exemplified below. These features were divided into four broad categories that capture the attributes of the games at a higher level of abstraction. It should be noted that game mechanics were previously divided into actions and behaviors [17-19][23][24][29][32].

1) **Avatar**
    a) **Operation rules**: Rules about the player. E.g.: the player can only carry one weapon at a time [4][14][18][28][33-35];
    b) **Transition rules/states**: Understanding the character's state transitions. E.g.: the player can only shoot if he/she has a weapon in his/her hand [24, 32, 4, 28, 35, 26];
    c) **Actions**: Commands that can be executed by the character. E.g.: shooting and walking [2, 29, 32, 35, 4, 26].
1) **Game world**
    a) **Levels**: The game's stages. Strongly influenced by the gameplay that can change from one stage to the next [24, 28, 32, 37, 35, 4, 26];
    b) **Rules of objects**: Rules of the objects contained in the world. E.g.: when an object must be locked or unlocked [2, 24, 29, 55, 4, 26];
    c) **Behavioral rules**: Rules of behavior that the world can exhibit. Eg .: if the player collects a specific item it can start to rain [24, 29, 37, 56, 30, 26];
    d) **Temporal states**: It works like a state machine; depending on the world's state, it can only go to a specific one [2, 24, 37, 56, 26];
    e) **Mission**: What you want to achieve/complete [2, 24, 32, 37, 30, 4, 26];
    f) **Obstacles**: What you must overcome in the game, its difficulties [37, 17, 35, 56, 30, 4, 26].
1) **Game play**
    a) **Winning and losing conditions**: Conditions to win or lose the game [24, 17, 56, 40, 4, 26];
    b) **Strategic dilemmas**: Strategies that can be used in the game. E.g.: combo attacks [24, 17, 56, 40, 4, 26];
    c) **Chains of actions**: Chain of actions that can be combined. Eg.: player action with a map action [24, 35, 56, 4].
1) **General features**
    a) **Rules**: Encapsulates the logic inside the system [2, 24, 28, 29, 34, 36, 4, 26];
    b) **Score**: The points obtained by the player throughout the game [24, 28, 35, 56, 4, 26];

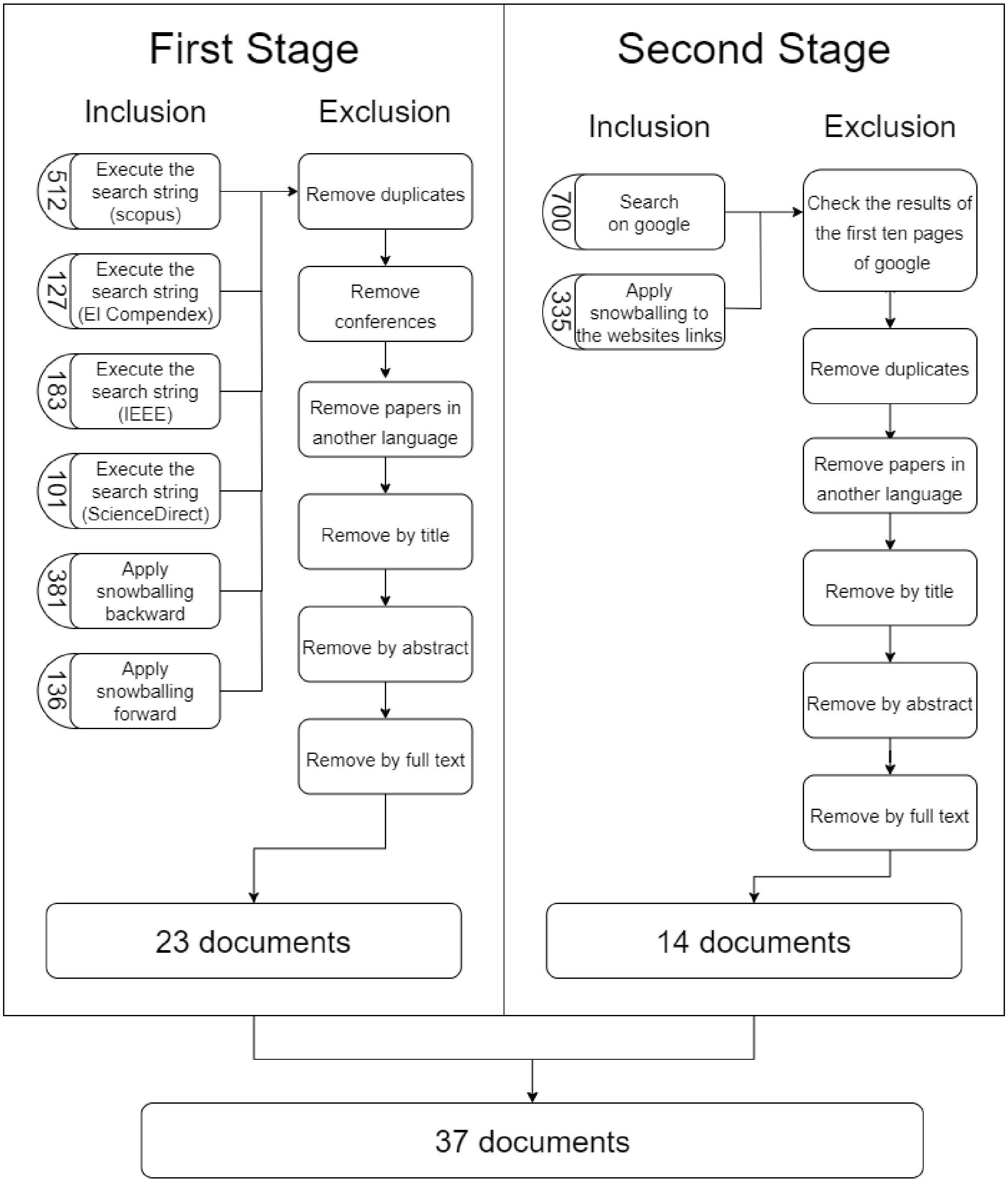
First Stage
Inclusion
Exclusion
512
Execute the search string (scopus)
127
Execute the search string (EI Compendex)
183
Execute the search string (IEEE)
101
Execute the search string (ScienceDirect)
381
Apply snowballing backward
136
Apply snowballing forward
Remove duplicates
Remove conferences
Remove papers in another language
Remove by title
Remove by abstract
Remove by full text
23 documents
Second Stage
Inclusion
Exclusion
700
Search on google
335
Apply snowballing to the websites links
Check the results of the first ten pages of google
Remove duplicates
Remove papers in another language
Remove by title
Remove by abstract
Remove by full text
14 documents
37 documents

Figure 1. Search flow of research on mutators and games.

c) **Behaviors**: Commands that are executed by the system [2, 24, 28, 56, 4, 26];
d) **Goal**: What you want to achieve/complete [2, 24, 32, 37, 4, 26];
e) **Challenge**: What must be accomplished to achieve the goal [2, 24, 28, 17, 4, 26];
f) **Rewards**: reaching the goal [2, 24, 29, 34, 4, 26];
g) **Game loop**: Flow of engagement of the game. It is the execution of the game where the player seeks a goal by executing a challenge and being rewarded with something [2, 28, 35, 56, 4];
h) **Interface**: The visual of the game, the game's sprites, and graphics [28, 29, 32, 34, 37, 17, 26];
i) **Entities**: Objects and elements instantiated within the game [28, 29, 56, 40, 26].

Q3: What are the advantages and difficulties of creating games from others?

Generalizing mod developers' intentions is difficult. There are several elements that contribute to a user producing a mod.

Attempting new things, resolving bugs, creating new characters, increasing the difficulty of the game, gaining advantages in the game, extending the game's life cycle, the software was originally designed for a significantly different environment and may require improvement, the official developer is unable to deal with the problems, and so on are among the most significant ones [2, 47].

Modifiers, like games, are complex and time-consuming to create. The time it takes to create a mod varies greatly. The construction process might vary in duration, ranging from a few days to a somewhat longer period, while offering the advantage of using reusable components. As previously mentioned, creating a game can be incredibly time consuming and can take years. However, the time necessary to release a mod is far shorter [2]. Mods allow the community to add to the original game. Depending on the nature of the mod, it may only require one or several releases. For example, a mod that improves the texture of a game may only require one version.

The potential to increase the longevity of games is another advantage that can be ascribed to the employment of modifiers. Every game has an effective life cycle. Modifiers, on the other hand, can extend the life of the game by adding additional instructions, characters, levels, and other factors, giving players more areas to explore [2, 4, 38]. Using the same logic, modifications may help boost sales, income, and profits for original games, as many people purchase the original game in order to play the mod [2, 52, 47].

Another significant advantage of modifications is their ability to draw new players to the game, so extending its longevity. For example, Dota 2 was a Warcraft mod that reached 450,000 daily players five years after its debut and 16 years after the original game's release. As a result, the game's player base and longevity grow [2, 39].

Along with the benefits described above, several additional are still directly tied to the community member who worked on the modification. Among the primary ones that stand out are the user's expression, communication between the firm and the end-user, and diversity of the game via end-user ideas [4]. Mods have become such a common practice in the community that they may now be considered a form of culture, allowing the user to incorporate their experiences into the game [5, 7].

When a player buys a game, he or she obtains a license to use the product. This license is structured in the manner of a copyright-based agreement [51]. Mods made by the community are susceptible to contract violations, placing their authors at considerable danger of being held liable for their actions, which frequently involve scamming games, exploiting product faults, or committing copyright infringement [6].

Last but not least, this mod technique has gotten so widespread that many large game production firms have opted to enlist members of the gaming community in order to cut development costs and risks. This cooperation enables the players to pool their unique perspectives and extensive experience and skills, therefore boosting the quality of the innovation without incurring additional resources, as well as certifying that it was what the community expected in terms of the game [30, 33].

Table V
ADVANTAGES AND DISADVANTAGES OF USING MODS.

Advantages / Disadvantages	Papers
Communication between the company and the end-user	[2, 41, 47, 49]
Diversification of the game through end-user ideas	[2, 45, 39]
Decreases the risk of game bugs	[47, 4, 41, 51, 38, 33, 49, 2]
Create new instructions, characters, levels, and other elements, providing players with new aspects to explore	[4, 5, 7]
Increase the game life cycle	[2, 4, 38, 51, 47, 33]
Shorter development time	[30, 33, 4]
Lower development cost	[51, 47, 33, 49, 41]
Increase the number of players of the original game	[2, 47, 49]
High initial investment	[27, 28]

Despite all the advantages described so far, some difficulties and challenges must be observed when creating mods. The first and main problem is the initial investment to produce a mod, which is necessary to understand the source code, reverse engineering and extract its features [28, 27]. Following this line of reasoning, some studies have already been carried out using the product line. However, this approach also requires an initial investment to conceptualize the initial features of the project [28, 27].

Table V demonstrates the advantages and disadvantages of using mods. The green color shows the advantages and the red the disadvantages.

Q4: What tools or frameworks support these changes?

Numerous frameworks and tools facilitate the building of modifications. However, the most prevalent technique of mod development so far has been cloning and do-it-yourself. The modder selects the basic game to be updated, verifies the characteristics he/she wants to modify, and then produces the new game [27]. This less complex strategy is referred to opportunistic reuse or ad-hoc reuse, and it comprises cloning, copying, and straining. Opportunistic reuse provides immediate advantages and produces the desired outcome. However, the quality of the project is not a priority, significant reworking leads to unexpected behavior and an unstable software structure [26].

Typically, games are changed using tools that enable access to an unencrypted internal representation of the game program. While it may appear as though game developers would aim to discourage consumers from customizing their games, this is not the case. Developers of video games are increasingly providing software tools for customizing their products in order to boost sales and market share [47]. Software development kits (SDKs) for games/domains supplied to users by game development studios represent a modern business approach for engaging users and assisting in product innovation outside the studio [6, 45, 30]. In addition to SDKs, which are the most common way of accessing the game's source code, several other platforms provide access to the game's source code and allow modifications. Among the main ones are the Creation Kit, GECK, Construction Set, MODKit, REDKit, Modbuddy, and D'jinni [2].

Another possibility for the development of modifiers is through free software games, in which the end user has complete access to the game's source code and may modify it as desired [6]. However, this strategy is used by small businesses or anonymous developers.

There are firms that assist and encourage the production of modifications with the goal of reducing problems, improving the game's quality and consistency, and generating new ideas. This technique leverages the users' ideas and wants to generate improvements for the game sold. The Unreal engine was created to provide access to all of its technology's components. This enabled it to host multiple events dubbed Unreal Tournaments, in which the developer may express his/her creativity while developing his/her mods [40, 2]. Other companies permit the construction of modifications as well, although without providing direct access to the components. For instance, Blizzard Entertainment's World of Warcraft has a UI modification tool that enables add-ons to modify the user interface panel, resulting in an enhanced gameplay experience. But these add-ons do not modify or convert the game into something entirely different since Blizzard seeks to ensure that players have access to the same configuration and mechanics as the original game [51].

In adiction, there are developers that produce mods by reverse engineering the source code of the original game. It is worth noting that this method for developing modifications is unlawful and violates the copyright of the original games [41].

Finally, single mod distribution platform may include several modifications for a single game. They all, however, adapt the same basic game. These platforms must be demonstrated in some way, such as demonstrating which files were changed in each mod, or if one mod is compatible with another, as both can change the same original game file [2].

Due to the large number of game variations generated based on an original game, maintenance can become difficult, and businesses may consider transitioning to a line of software products, referred to as an extractive method, to assist with mass game production [27, 35, 25, 28, 26].

Finally, there are several sites that support mods, integrating the community and providing thousands of games. Among the main games are gamemodding (http://www.gamemodding.net), moddb (https://www.moddb.com), modsonline (http://modsonline.com/), among others [2, 56].

V. Discussion

The research revealed that the community has been utilizing methods of reuse to create games. Modding, the practice of game reuse development using existing games as a base, is widely utilized by both the gaming community and companies to boost their games or expand their game collection. Mods are a common means of self-expression in the community, however, they are typically created in a casually manner and are frequently linked to opportunistic reuse, frequently practiced out through clone-and-own techniques. Search results reveal other techniques for creating games with SR, like software componentization, enabling integration with popular game engines. Nevertheless, techniques like SPL and MDD are currently in the testing phases, with limited instruments for use.

Utilizing reuse in software development can provide several advantages, such as decreasing development time, enhancing software quality, and facilitating quicker and easier product expansion. Considering these remarkable benefits, it is feasible to relate them to the challenges in game development, such as long development time, risk of numerous defects, and complexity of generating new versions of the original game.

Various methods exist for implementing software reuse, ranging from clone-and-own to product lines. Each of these methods has distinct benefits and suits certain goals. Each of these topics will now be illustrated with a focus on developing games. All the following observations were made with consideration for the development of mods and game expansion.

- **Clone-and-own:**
  - **Context of use:** This method could be used for modest projects if expanding the game is not part of the initial plan.
  - **Advantage:** This approach provides an excellent initial return by reusing a significant portion of the existing game.
  - **Disadvantage:** Future maintenance and upgrades can generate significant expenses if the same changes and improvements are applied to multiple projects simultaneously.
  - **Example:** adapt particular elements of an open-source game.
- **Componentization:**
  - **Context of use:** This method is suitable for tasks of medium to big complexity. Expansion of the original game is premeditated, but new games can be created by reusing its components. The main objective is to create new games by utilizing its components again.
  - **Advantage:** this strategy is widespread and utilized by numerous engines. Any alteration to a component will result in all games utilizing that component being updated.
  - **Disadvantage:** does not allow expansion of games directly.
  - **Example:** in a First-Person Shooter (FPS) game, all moves and a lot of actions may be broken down into smaller parts because they are the same in all games.
- **MDD:**
  - **Context of use:** This method is appropriate for medium to large projects. The original game's expansion was not preplanned. This strategy is popular because it begins implementation only after a well-

developed model has been generated through numerous iterations.
  - **Advantage:** platforms based on this production approach enable anyone with basic programming knowledge to produce full-length games.
  - **Disadvantage:** unlike the others, it is not a development pattern, but a development process.
  - **Example:** use in projects with poorly specified criteria and a high probability of modification due to the ability to change the models before beginning the project.
- **SPL:**
  - **Context of use:** this method is appropriate for medium to large projects. The expansion of the original game was planned before.
  - **Advantage:** several games can be derived from a single game by selecting game features.
  - **Disadvantage:** few tools and platforms available.
  - **Example:** the product line makes it easy to make games that can be played on a variety of platforms. Think of a game that needs both a mobile and a desktop form.

VI. Conclusion

Game companies are growing in size, generating billions of dollars each year, releasing many titles each year, and attracting fans of all ages and genres. However, as has been seen so far, developing a game may be a long process that might take years to complete. However, the gaming community is rising daily. With such a vast user community, some members may experience anxiety or dissatisfaction at the prospect of having to wait so long for a game to be published.

With a little online search, it is possible to locate multiple games for sale and several websites that provide modifications for them. As previously mentioned, a mod may be defined as a modification made to a game and depending on the level of this modification this mod can receive different names, such as: patches, tweaks, add-ons, among others. This method of modifying games can result in a number of benefits for the company that generated the original games. The benefits are numerous, and some businesses even encourage this practice. Among the primary benefits are an increase in users, an increase in sales, and an increase in the game's longevity, among others. However, the study revealed that the process of developing a mod might be expensive and ad-hoc.

It is noted the presence of tools and frameworks that support modifications, ranging from clones to SDKs and tools made accessible by the game's developer, among other techniques. However, these tools are frequently associated with a number of difficulties, including a large initial investment, a steep learning curve, the requirement to comprehend the source code, and the fact that the majority of these techniques are limited to the creation of basic games.

Throughout the investigation, multiple papers were found that contrasted ad-hoc mod development with opportunistic reuse, in which software is built using similar strategies of small modifications and copying and pasting [57, 58]. Demonstrating once again the need of systematizing the process of mod development. Additionally, it was seen in these same papers an early application of Software Reuse (SR) for game building. However, it was employed superficially and exploratorily.

As is already known, SR can bring several types of advantages in the construction of software in general, from systematization of development to an increase in delivery speed and cost reduction. It was observed that these advantages were being highlighted in some articles through the research, but, however, the approaches demonstrated were of an exploratory nature, with the exception of software componentization, an approach that has already been used by programmers and is even found in some more current advanced engines [59, 60].

Initial discussions about the possible use of SR approaches in game development were made in the discussion section, however, it was decided to conduct another research that would complement this study. The proposed study sought to determine which SR techniques are being utilized to create games or mods.

References

[1] D. Castro and C. Werner, "A multivocal review on derivation games," *The 2023 IARIA Annual Congress on Frontiers in Science, Technology, Services, and Applications, IARIA Congress*, pp. 144–149, November 2023.

[2] D. Lee, D. Lin, C.-P. Bezemer, and A. E. Hassan, "Building the perfect game–an empirical study of game modifications," *Empirical Software Engineering*, pp. 1–34, 2020.

[3] S. Pashkov, "Video game industry market analysis: Approaches that resulted in industry success and high demand," 2021.

[4] K. Bilińska, A. Dewalska-Opitek, and M. Hofman-Kohlmeyer, "To mod or not to mod—an empirical study on game modding as customer value co-creation," *Sustainability*, vol. 12, no. 21, p. 9014, 2020.

[5] A. Unger, "Modding as part of game culture," *Computer Games and New Media Cultures*, pp. 509–523, 2012.

[6] W. Scacchi, "Modding as a basis for developing game systems," *Proceedings of the 1st international workshop on Games and software engineering*, pp. 5–8, 2011.

[7] ——, "Modding as an open source approach to extending computer game systems," *Proceedings of IFIP International Conference on Open Source Systems*, pp. 62–74, 2011.

[8] V. Garousi, M. Felderer, and M. V. Mäntylä, "Guidelines for including grey literature and conducting multivocal literature reviews in software engineering," *Information and Software Technology*, vol. 106, pp. 101–121, 2019.

[9] J. Ehrmann, C. Lewis, and P. Lewis, "Homo ludens revisited," *Yale French Studies*, no. 41, pp. 31–57, 1968.

[10] R. Hunicke, M. LeBlanc, and R. Zubek, "Mda: A formal approach to game design and game research," in *Proceed-*

ings of the AAAI Workshop on Challenges in Game AI, vol. 4, no. 1, 2004.

[11] B. Randell, "Software engineering in 1968," in *Proceedings of the 4th International Conference on Software Engineering*, ser. ICSE'79. IEEE Press, 1979, p. 1–10.

[12] C. W. Krueger, "Software reuse," *ACM Computing Surveys (CSUR)*, vol. 24, no. 2, pp. 131–183, 1992.

[13] J. Sametinger, *Software Engineering with Reusable Components*. Springer Science & Business Media, 1997.

[14] N. Maia, A. P. T. Bacelo, and C. M. L. Werner, "Odyssey-mda: A transformational approach to component models." *Proceedings of Internacional Conference on Software Engineering and Knowledge Engineering*, pp. 9–14, 2007.

[15] B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering–a systematic literature review," *Information and software technology*, vol. 51, no. 1, pp. 7–15, 2009.

[16] M. Petticrew and H. Roberts, *Systematic reviews in the social sciences: A practical guide*. John Wiley & Sons, 2008.

[17] V. McArthur and R. J. Teather, "Serious mods: A case for modding in serious games pedagogy," *Proceedings of IEEE Games Entertainment Media Conference (GEM)*, pp. 1–4, 2015.

[18] T. M. Moraes and A. d. SOUZA, "Revisão sistemática sobre a comunicação dentro do processo de desenvolvimento de software," *Universidade Federal de Goiás-GO*, p. 57, 2011.

[19] R. C. Motta, K. M. de Oliveira, and G. H. Travassos, "Characterizing interoperability in context-aware software systems," *Proceedings of VI Brazilian Symposium on Computing Systems Engineering (SBESC)*, pp. 203–208, 2016.

[20] S. Matalonga, F. Rodrigues, and G. H. Travassos, "Characterizing testing methods for context-aware software systems: Results from a quasi-systematic literature review," *Journal of Systems and Software*, vol. 131, pp. 1–21, 2017.

[21] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," in *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, 2014, pp. 1–10.

[22] Y. Lincoln and E. Guba, "Naturalistic inquiry. encyclopedia of research design. 2455 teller road, thousand oaks california 91320 united states," 2016.

[23] O. Sotamaa, "On modder labour, commodification of play, and mod competitions," *First Monday*, vol. 12, no. 9, 2007.

[24] D. Abbott, "Modding tabletop games for education," *Proceedings of International Conference on Games and Learning Alliance*, pp. 318–329, 2018.

[25] J. Debbiche, O. Lignell, J. Krüger, and T. Berger, "Migrating java-based apo-games into a composition-based software product line," in *Proceedings of the 23rd International Systems and Software Product LineConference, vol. A*, 2019, pp. 98–102.

[26] C. Lima, W. K. Assunção, J. Martinez, W. Mendonça, I. C. Machado, and C. F. Chavez, "Product line architecture recovery with outlier filtering in software families: the apo-games case study," *Journal of the Brazilian Computer Society*, vol. 25, no. 1, pp. 1–17, 2019.

[27] J. Krüger, W. Fenske, T. Thüm, D. Aporius, G. Saake, and T. Leich, "Apo-games: a case study for reverse engineering variability from cloned java variants," in *Proceedings of the 22nd International Systems and Software Product LineConference, vol. 1*, 2018, pp. 251–256.

[28] W. D. Mendonça, W. K. Assunção, and L. Linsbauer, "Multi-objective optimization for reverse engineering of apo-games feature models," in *Proceedings of the 22nd International Systems and Software Product LineConference, vol. 1*, 2018, pp. 279–283.

[29] R. Damaševičius and D. Ašeriškis, "Visual and computational modelling of minority games," *TEM J*, vol. 6, no. 1, pp. 108–116, 2017.

[30] L. Poretski and O. Arazy, "Placing value on community co-creations: A study of a video game'modding'community," in *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, 2017, pp. 480–491.

[31] T. Dey, J. L. Massengill, and A. Mockus, "Analysis of popularity of game mods: A case study," in *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts*, 2016, pp. 133–139.

[32] R. Al-Washmi, J. Bana, I. Knight, E. Benson, O. A. A. Kerr, P. Blanchfield, and G. Hopkins, "Design of a math learning game using a minecraft mod," in *European Conference on Games Based Learning*, vol. 1. Academic Conferences International Limited, 2014, p. 10.

[33] T. Tengtrirat and N. Prompoon, "Applying exception handling patterns for user interface customization in software games modification," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. 1, 2013.

[34] S. George, É. Lavoué, and B. Monterrat, "An environment to support collaborative learning by modding," in *European Conference on Technology Enhanced Learning*. Springer, 2013, pp. 111–124.

[35] S. Zuppiroli, P. Ciancarini, and M. Gabbrielli, "A role-playing game for a software engineering lab: Developing a product line," in *2012 IEEE 25th Conference on Software Engineering Education and Training*. IEEE, 2012, pp. 13–22.

[36] G. Cheung and J. Huang, "Remix and play: lessons from rule variants in texas hold'em and halo 2," in *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, 2012, pp. 559–568.

[37] M. F. Shiratuddin and W. Thabet, "Utilizing a 3d game engine to develop a virtual design review system," 2011.

[38] O. Sotamaa, “When the game is not enough: Motivations and practices among computer game modding culture,” *Games and Culture*, vol. 5, no. 3, pp. 239–255, 2010.

[39] H. Postigo, “Of mods and modders: Chasing down the value of fan-based digital game modifications,” *Games and Culture*, vol. 2, no. 4, pp. 300–313, 2007.

[40] D. B. Nieborg, “Am I mod or not?—An analysis of first person shooter modification culture,” in *Creative Gamers Seminar—Exploring Participatory Culture in Gaming, University of Tampere, Finland (14–15 January)*, 2005.

[41] J. Kolbert, “Unofficial patch,” https://en.wikipedia.org/wiki/Unofficial_patch, online; accessed 09 March 2021.

[42] R. F. Patterson, “Mod (video gaming),” https://civilization.fandom.com/wiki/Mod_(video_gaming), online; accessed 05 March 2021.

[43] C. Weeke, “Appropriation & motivation in game modification,” *Erasmus University Thesis Repository*, 2020.

[44] M. Magioladitis, “Video game conversion,” https://en.wikipedia.org/wiki/Video_game_conversion, online; accessed 07 March 2021.

[45] M. M. Hofman-Kohlmeyer, “Players as content creators. the benefits of game modding according to polish users,” *International Scientific Journal News*, vol. 2, pp. 8–26, 2019.

[46] N. Cleland, “Mod (video games),” https://en.wikipedia.org/wiki/Mod_(video_games), online; accessed 04 March 2021.

[47] S. Agarwal and P. Seetharaman, “Understanding game modding through phases of mod development.” *Proceedings of ICEIS*, pp. 114–121, 2015.

[48] R. Ramadan, “Does game modding require programming?” https://www.quora.com/Does-game-modding-require-programming, online; accessed 09 March 2021.

[49] N. Poor, “Computer game modders’ motivations and sense of community: A mixed-methods approach,” *New media & society*, vol. 16, no. 8, pp. 1249–1267, 2014.

[50] E. Champion, *Game mods: design, theory and criticism*, Lulu.com, 2013.

[51] S. Walt, “Computer game mods, modders, modding, and the mod scene,” *First Monda*, vol. 15, no. 5, 2010.

[52] W. Scacchi, “Computer game mods, modders, modding, and the mod scene,” *First Monday*, 2010.

[53] B. T. Claude Chaunier, Torben Mogensen, “Mutators,” http://www.di.fc.ul.pt/~jpn/cv/mutators.htm, online; accessed 04 March 2021.

[54] G. Xexéo, A. Carmo, A. Acioli, B. Taucei, C. DIpolitto, E. Mangeli, J. Kritz, L. Costa, and R. Monclar, “What are games?” *LUDES. Rio de Janeiro*, vol. 1, pp. 1–30, 2013. (In Portuguese).

[55] J. Åkesson, S. Nilsson, J. Krüger, and T. Berger, “Migrating the android apo-games into an annotation-based software product line,” in *Proceedings of the 23rd International Systems and Software Product LineConference, vol. A*, 2019, pp. 103–107.

[56] G. A. Cignoni, “Reporting about the mod software process,” in *European Workshop on Software Process Technology*. Springer, 2001, pp. 242–245.

[57] W. Fenske, J. Meinicke, S. Schulze, S. Schulze, and G. Saake, “Variant-preserving refactorings for migrating cloned products to a product line,” in *2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, 2017, pp. 316–326.

[58] M. Sierra, M. C. Pabón, L. Rincón, A. Navarro-Newball, and D. Linares, “A comparative analysis of game engines to develop core assets for a software product line of mini-games,” in *International Conference on Software and Systems Reuse*. Springer, 2019, pp. 64–74.

[59] N. Aouadi, P. Pernelle, C. B. Amar, T. Carron, and S. Talbot, “Models and mechanisms for implementing playful scenarios,” in *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*. IEEE, 2016, pp. 1–8.

[60] E. Albassam and H. Gomaa, “Applying software product lines to multiplatform video games,” in *2013 3rd International Workshop on Games and Software Engineering: Engineering Computer Games to Enable Positive, Progressive Change (GAS)*. IEEE, 2013, pp. 1–7.