Port or conversion? An ontological framework for classifying game versions.

Paweł Grabarczyk
IT University of Copenhagen
Rued Langgaards Vej 7
2300 København
pagrab@gmail.com

Espen Aarseth
IT University of Copenhagen
Rued Langgaards Vej 7
2300 København
aarseth@itu.dk

ABSTRACT
The notion of a "game port" appears in both: the popular discourse surrounding video games as well as the academic discourse. As pointed out in the literature, it is notoriously vague and often overlaps with other similar concepts, such as "conversion". We show that the main reason for the murkiness of the notion of a "game port" is its close connection to the marketing narrative which changed during history. We argue, that despite the problematic nature of the concept of a "game port" it is indispensable in context such as game analysis and game preservation. For this reason, we propose a formal classification of different game versions suitable for future use in these selected contexts.

Keywords
Port, version, adaption, clone, remaster, reskin, conversion

INTRODUCTION
In this paper, we present a model for discerning between different types of game versions. When is a port a port, and not a clone; what is the difference between a remaster and a reskin? This formalizing approach to what, until now, have been vernacular and fuzzy categories should be of help to all who needs a more nuanced language for describing relations between game objects that share large parts of their identities with other game objects.

On the face of it, the notion of a "port" seems rather straightforward. Any software can be said to be "ported" if it has been reproduced on another platform by carrying over some parts of code - as opposed to being written "from scratch". This understanding of game "ports" relates to the notion of software "portability" used in computing from the 1970s (Tanenbaum et al. 1978). Unfortunately, the actual usage of the term often deviates from this simple definition. As pointed out by (Švelch 2018) the concept of a "port" has been typically used in a much more liberal way to describe cases where no parts of code was literally "ported". Švelch suggests that we should reserve the notion of a "port" only to cases where code was reused and use the notion of a "conversion" for the cases where the code has been written anew. This is a good first step in tidying up the conceptual confusion surrounding the notion of "ports" but demands further investigation. Švelch's idea seems to function rather as a necessary condition than as a definition, because it assumes that we already established two objects to be versions of

Proceedings of DiGRA 2019

the same game. It is fairly customary for different games produced by the same studio to reuse parts of the code or other assets, so game ports have to share more than that.

Another reason why a further study of ports and conversions is needed is that both concepts belong to a bigger network of partially overlapping, ambiguous notions, such as "game translation", “game adaptation” or "game clone". Other concepts, such as "game version" seem to function as a more general category which encompasses "game ports". Contrary to this, notions such as "remaster" or "remake" seem to refer to more specific types of game ports. Additionally, a number of different notions, such as "emulation", "mod" or "game hack" demand to be related to the notion of "game ports" as it is not clear if they could also be considered to be forms of "game ports".

Last but not least, the success of several crowdfunding campaigns resulted in the rise of popularity of board adaptations of video games (*Dark Souls* (2011) and *Starcraft* (1998) are notable examples of these). Even though the phenomenon itself is not new (there was for example a board game adaptation of *Pac-Man* (1980) released in 1980), the relation between the original video game versions and their board counterparts is much more complex in contemporary cases because the adaptation has to be much less straightforward - after all, most of the early arcade games which confined their game space to a single screen, could easily be understood as a digital form of a board.

This conceptual confusion as well as the inconsistencies of usage throughout history (which we explore in section 2) make the notion of "game ports" ill-suited for academic use, so it might be tempting to abstain from it altogether. And yet, we believe that there are several reasons to retain it, but in a more precise form. The notion of "game ports" belongs to the ordinary vocabulary of video game players and journalists but it can also be said to be tacit in the scholarly convention which requires the author to disclose the platform on which the game has been played. Additionally, the notion of ports functions as an opposite of the notion of "game exclusives" which belongs to an important marketing narrative in history of video games. Moreover, in the cases of games produced simultaneously for multiple platforms, one of these platforms typically functions as the leading one and the other versions are not classified as ports mostly for marketing reasons (as they may be then seen as inferior). The result of this is that the majority of existing games can be classified as ports. This makes our inability to explain what ports are even more jarring.

Most importantly, even if we dispose of the notion, we will still not be able to escape the distinctions this notion was supposed to define. Any time a given game has to be classified in a database system or is presented at an exhibition a decision as to which of the various versions of the game is to be used has to be made (Newman 2012). Some of these versions vary greatly and there are no clear criteria as to what should we disregard as unimportant variations. One good example of such a practical problem can be found in (Stuckey 2014) who points out that choosing between different versions of the classic text adventure game *The Hobbit* (1982) is far from trivial. Which version of the game should we choose if we want to present it to the public? The most popular, the best looking one or maybe the first one produced? Many times this decision is also dictated by purely pragmatic reasons - we choose the version which will be the easiest for us to run on the hardware we have or one which will be the most accessible to the audience. But then, it would be good to know what the choice we made entails - which aspects of the other versions of the game does our version retain and which of them does it omit.

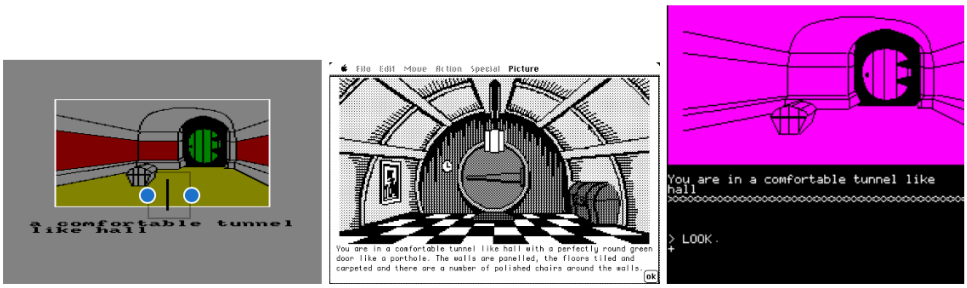

Figure 1: Different faces of The Hobbit. From left to right: ZX Spectrum, Apple GS, PC CGA.

Academic usage of the term leads to analogous considerations. Any time an example of a game is used in an academic context, a tacit ontological decision about the identity criteria for the game has to be made. If Doom (1993) is referenced in a paper, should we assume that what is written pertains only to the version acknowledged in the ludography, any game entitled "Doom" or to a subset of them? In some cases, the differences between conversions, ports or remakes may not be substantial but in other cases they may become crucial. For example, in the case of platform studies (see Fernández- Vara & Montfort 2014 as well as chapter 2 of Altice 2015), comparison of ports gives us much more information about the specificity of the platforms they run on, than the comparison of conversions or remakes, which depend largely on the skills of the programmers and their ability to reverse engineer the original.

In both cases - the academic and the practical - abstaining from having our own criteria delegates the decision to the marketing narrative of the time. We treat two pieces of software as versions of "the same game", because this is how it was sold to us. The decision as to what counts in a given context should be left to the author/creator, but in order for her decision to be well-informed, she has to be able to realise what the differences between versions consist of. The classification presented in section 5 aims to do just that.

THE HISTORICAL INCONSISTENCY OF “PORTS”

The usage of the notion of "game port" changed significantly over time to the point of being inconsistent and seems to be closely tied to the marketing narrative of game producers. Different eras of computing resulted in different technical challenges which resulted in conceptualizations that are not well suited for general use. We could say that the notion of "game port" had to be adapted itself.

In the 1980s the notion of game ports remained in everyday use, because it was a part of a more specific notion of "arcade-perfect” or “arcade-accurate” ports (Altice 2015) which remained to be the elusive target of programmers of this era. The arcade machines of the time were much more advanced than any home machines. Typical arcade machine contained hardware which the users could only hope to find in home machines in the next 5-10 years. For this reason, recreating the exact same looking game for homes was simply impossible and the best the players could expect was a facsimile of the arcade game. Moreover, the existing home platforms differed significantly amongst each other so a detailed comparison of "ports" was often presented in game magazines. The situation changed slightly in the 1990s when SNK introduced its NEO GEO console which was identical to the arcade machine and could play the same games. The impact of NEO GEO was small because the console was sold for a very high price - 649 USD which is an equivalent of 1200 USD in 2019.

Additionally, since the hardware of the arcade board was identical, the games played at home were not really ports but copies of the same software.

Mid and late 1990s enabled the developers to get closer to the ideal of an “arcade-perfect port”, because the arcade boards of many successful games contained enhanced versions of home hardware. Two notable examples of this are Sony Playstation (a simplified equivalent of the Namco System 11 arcade board) and Sega Dreamcast (an equivalent of Sega's NAOMI arcade board). These similarities resulted in very faithful ports because the hardware was close enough to enable portability but still not identical, so the machines were not simply compatible (as was the case with NEO GEO). Interestingly, when "arcade perfect ports" of older games were finally delivered to home consoles and computers, it was sometimes achieved through emulation rather than porting. Some of the entries in the popular Namco Museum series are some of the first examples of this phenomenon.

The inability to create arcade perfect ports for the home machines as well as different expectations of home users resulted in many odd cases where home versions of games ended up being completely different from originals. Two interesting examples of this can be found in Dragon's Lair (1983) and Bionic Commando (1987). The arcade version of the former functioned as an (barely) interactive movie in which the player watched a full-screen cartoon and reacted with a single input of a joystick (similarly to more modern "quick time events"). The home version of Dragon's Lair created for NES and SNES functioned as traditional platform games. The arcade version of Bionic Commando was a linear platform game, which has been released for various systems such as Amiga, Atari ST and Commodore 64 but the NES version differs from the original so much that it is often considered to be another game because of its non-linear nature and different level layouts. This invalidates the easiest solution we could have hoped to employ: we cannot simply follow the naming conventions developers use and assume, that software produced by the same company under the same name are all different versions of the same game.

Figure 2: Many faces of Bionic Commando. Left to right: Arcade, Amiga, NES.

Figure 3: Many faces of Dragon's Lair. Left: Arcade, Right: NES, Bottom: SNES

Many difficulties relating to the usage of the notion "game ports" are direct results of the marketing narrative. Since the ability to play popular arcade games at home functioned as a strong incentive for buyers, games which were recreated from scratch - games in which nothing was literally "ported" - were still called "arcade ports". This pertained especially to the 1980s home computer market where the developers of home versions never saw the original code or assets and simply reproduced the games based on their look and feel (Edwards 2016). In some cases they were able to hack into the contents of the memory of the arcade machines so it was possible to extract parts of the graphics. In other cases, the only thing they were given was a VHS tape with the gameplay. (Ager 2016)

It may be hard to imagine today, but in some cases games created on the basis of a cursory knowledge of the original could have ended up being quite faithful. This was especially possible in the case of games with a very simple structure. Consider Snake - a game popularised by its 1997 incarnation developed for Nokia phones, but one which originated from the 1976 arcade game Blockade (1976). Snake remains of the most popular programming exercises, because the game can be quite easily recreated using less than 100 lines of code. Are all of these incarnations ports, just because they share the same concept, game logic and quite often the same implementation? Tetris (1984) is probably an example of a game which has been recreated for most existing platforms - which of the versions of Tetris should be considered as ports? Problems of this type are the direct result of the tension between the conceptual distinction of ports and conversions with the practical side of game production process. The whole idea of a "port" stems from the common sense observation that some of the assets used in production could be reused in the process of adapting the game to another platform. In cases where the assets were extremely simple - the code was very short, the graphics

consisted of geometrical shapes, the sound was either non-existing or consisted only of single beeps, the difference between reusing assets and recreating them from scratch is negligible. This problem may be considered to be marginal if it pertains only to a finite number of early video games, but it resurfaced recently because of the popularity of simple web-based and mobile games, some of which may be even simpler than classics of the past, such as Flappy Bird (2013)

Another can of worms opens when we start talking about the difference between "licensed" and "unlicensed" ports. Consider the infamous example of Atari 2600 Pac-man. The game is definitely a very poor version of Namco's 1980 original, even though it was officially licensed. As can be seen in a homebrew port created in 2015 by an amateur programmer (who used the nickname DINTAR816[1]) the limitations of Atari 2600 hardware were not the main reason for the inferiority of the version published in 1982, as DINTAR816's version retained a lot of the original look, sound and mechanics. How should we classify these versions? Looking into the production process does not help. No assets used in the Namco original were actually ported to the 1982 version, because its author, Tod Frye, did not have access to them. In reality, they might not even have been very useful, because of the differences in hardware between the Atari 2600 and the Namco arcade board. Instead, Frye played the original and tried to reverse engineer the game. To do this, he had to assume which aspects of the game were essential, and which could be easily compromised. For example, Frye decided that the maze layout was not very important and created his own version of it.[2] This decision proved to be very controversial. The discussions on the Atari Forum where DINTAR816's version was revealed and presented show that the author also reverse engineered the game. Still, the process of reverse-engineering Pac-man in 2015 arguably demanded much less guesswork than the one in 1982. The game logic is fairly well known and the process of reverse engineering is much easier because of the ability to run the original rom on emulators and looking into it through debuggers. Does either of these versions deserve to be called a port? Should we give precedence to legal agreements between the developer and the original creators or to the faithfulness of the result?

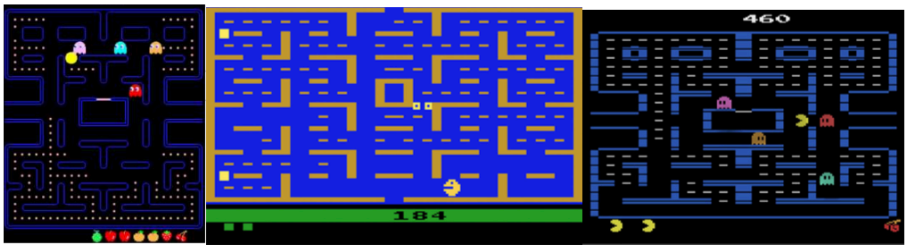

Figure 4: Many faces of Pac-Man. From left to right: Arcade, Original Atari 2600 version, DINTAR816's version.

[1] The game as well as discussion of its technical achievements can be found at Atari Age forums: http://atariage.com/forums/topic/229152-new-pacman-for-atari-2600/ (retrieved 11.02.2019)

[2] This is disclosed in an interview which can be accessed at https://www.youtube.com/watch?v=RqezF_Lv05Y (retrieved 11.02.2019)

In fact, neither of these choices are satisfactory. To reserve the notion of a port only for cases where the result has been approved by the publisher reduces the academic discourse to marketing narrative and seems to goes against well established notions of "licensed" and "unlicensed ports". What is more, it replaces a murky concept with an even murkier one as game marketing used the notion in a completely inconsistent way, compliant only with the sales needs of the time. In most cases pointing out the fact that the game was a port of an existing successful game worked as a sales push, stimulating producers to overuse the term. In other cases it was equally economically important to deny that the game is simply a port of an existing game, because marketing efforts focused on assuring the customers that the game was brand new. An example of the latter can be found in some of the recent Wii U games repurposed by Nintendo for their new console, Nintendo Switch. The confusion which this marketing strategy creates can be seen in the fan debates inspired by the release of games such as Splatoon 2 (2018) or Super Smash Bros. Ultimate (2018).

The production process of the games advertised as ports was sometimes even more convoluted. Good example of this is Thunder Blade (1987) which was ported to multiple home platforms in the late 1980s. The version published for Atari ST was recreated from scratch (using reverse engineering) but the result was later ported (by means of actual reusing of assets and code) to the competing platforms, such as Amiga. Are these games ports of the arcade game or ports of a conversion of ST game?

The criterion of the "faithfulness" of a port is not without its caveats either, since it typically related to the fact that arcade machines were typically much more capable than home consoles and computers of the time. It seems that in many cases the expectation towards the ports was less about being "faithful" and more about utilizing the potential of the machine the game was ported to. This can be easily seen in cases where, contrary to the arcade adaptations, the ports were created for machines superior to the original platforms. For example, Commodore 64 ports of successful ZX Spectrum games were often criticised for not utilising enough colour, even though the lack of colour was what made them more faithful to the originals. One notable exception to this rule pertains to game mechanics and structure. As can be seen in numerous reviews and comparisons, the expectation towards ports was that they preserved the exact mechanics of the original. Even small deviations were typically seen as a detriment of the new version. For example, the port of Marvel vs Capcom (1999) for the Playstation 1 contained several compromises due to the memory limitations of Sony's console. Even though the compromises of the visual aspects were typically pointed out, the main reason the port was not well received was the removal of the ability to change the characters during a tag battle. As noted by Jeff Gerstman, who reviewed the game for the popular site Gamespot: "The PlayStation version may have the same moves as the original game, but the shell surrounding those moves is completely different" (Gerstmann 1999). The reason for this bias towards mechanics is rather understandable. Compromises to the audiovisual aspects of the game were necessary because of the differences in hardware. Compromises to mechanics were considered unwarranted simplifications.

To summarize, the analysis of the historical usage of the notion of “game ports” shows that none of the simple criteria seem to be stable enough to be useful in the academic discourse and contemporary classifications. Using just the name is too coarse-grained because many games sharing the same name are different. Using the combination of a name and a platform indication seems to be too pedantic because in many cases games are too similar to study them separately (e.g. some Amiga and ST ports or PS3 vs Xbox360 ports). Adhering to the way games were advertised is too risky because the publishers had reasons to use the term “port” in an inconsistent way and looking at how the term was used historically shows that there is no clear pattern for classifying

something as a “port”. Invoking the notion of legality does not help, as it makes a fairly well-spread notion of an “unlicensed port” self-contradictory. Last but not least, sticking to the literal meaning of “porting” as “being carried over” is not consistent with the practice of reusing assets in many different games produced by the same studio.

ANALOGOUS NOTIONS

The discussion of the notions of game ports and conversions does not take place in a vacuum as they can be also compared to similar phenomena in different media. Unfortunately, as we will see, while some of the similarities are striking, the analogies are not as helpful as we might hope.

There first and the most obvious comparison comes from the field of linguistic translations. Translation tries to preserve the semantic, pragmatic and artistic aspects of the original text at the cost of changing its vocabulary and syntax. At first the analogy appears useful. Vocabulary and syntax can be seen as material and technological constraints similar to those which the developers of ports have to take into account. Semantics can be seen as crucial aspects of the game which have to be preserved in order for it to be recognised as being "the same game". Pragmatic and artistic qualities can be understood as the user experience which the developers try to reproduce. What is more, the act of porting a game can sometimes literary function as a translation in cases where the programmers have the access to the source code of the original and use it as a basis of their own code in similar programming language. On top of that, the notion of "translation" has sometimes been used as synonymous with “porting”. There are two reasons why this analogy is not perfect though. First of all, it does not allow us to account for the compromises typically needed to port a game to a much less powerful system. Even though languages differ in their expressive power (especially when it comes to their artistic repertoire) the differences are never as significant as those which the developers of ports of arcade games had to struggle with. Second of all, the notion of a translation is probably more analogous to the notion of "conversion" (in Švelch’s sense of the word) as translations do not typically reuse the parts of the original.

Another potentially fruitful analogy can be found in the field of music. After all, musical pieces are often rearranged, covered or sampled. On the face of it the comparison seems useful. Rearrangements can be likened to conversions while covers and remixes can be seen as ports. Note that the points where the comparison with translations failed do not apply here because producing covers for a simplified instrumentation can easily be likened to creating a port for an inferior hardware. The problem is that the ontology of different versions of the same musical piece is just as much in need of clarification as the ontology of games because there are no clear boundaries we could try to adapt to our case.

Last analogous notion which might be worth considering is the notion of a "replica". Are ports, conversions, remakes or clones replicas of the originals? As far as we know, this word has never been used in reference to game versions. It seems that the reason for it is that the notion of a replica is strongly connected to the idea of materiality, and becomes much harder to apply in the case of digital objects, such as computer programs. Secondly, what makes replicas different than game versions is that even though replicas seem to belong to the same category of objects as their originals, they are not considered instances of the same type, because of the historical context of their production. They are replicas, because they were produced in different times or by a different group of people. These considerations never mattered in the case of game versions as ports were often produced after many years or by different developers.

PROPOSED SOLUTION

The solution we propose to these conceptual problems is a formal model which can be used for a more rigorous classification of game versions. The main aim of the model is to facilitate the academic discourse and aid in various practical classifications (such as game preservation). It is important to point out that the ambition of the model is purely practical and that it should be understood as explication of the notion of a "game port" (and other neighbouring notions) and not as a normative definition. The idea of an "explication" comes from Carnap and refers to a very specific type of definition, similar to the regulative definition in law. The aim of the explication is to create a rigorous counterpart of a messy notion so that it retains as much of the existing usage as possible without falling into inconsistency. The resulting term should not be understood as the "true meaning" of the analysed word, but rather as a model which retains usefulness in selected applications. The hope of successful explications is to reclaim a given expression for selected usages and not to correct existing usages or replace the common word it originated from.

The model we present derives from a general ontological model presented in (withhold) and can be understood as an example of its specific application. The framework describes games using three ontological layers: **the physical layer**, **the structural layer** and **the presentation layer**. As the names imply, the physical layer relates to material substrate of the game. It is later subdivided into two categories: **the physical platform** the game runs on and **the physical interface** which the players use to control it. Typical example of platforms are: arcade machines, consoles, handhelds or home computers. Less obviously, ‘platform’ can be also understood as non-computational artefacts such as boards and card decks. Typical examples of physical interfaces are: mouse and keyboard, gamepads, joysticks or VR headsets. The difference between hardware platform and hardware interface may sometimes be blurry, but a rule of thumb is that the physical interface can be changed without changing the platform. Needless to say, this may be sometimes technically difficult - changing the physical interface of an arcade machine can be an example of that - but what is important is that it is, in principle, possible.

The division of the structural layer is the most complex one. We propose to divide it into three sub-layers: **computational core**, **computational shell** and **the mechanical layer.** The distinction between computational core and computational shell may not be obvious, but it is crucial from the point of view of the discussion of game ports. The basic idea behind this division is to differentiate between game’s code (computational core) and the additional software environment which is needed for the game to run (computational shell). Typically this environment is largely independent in the sense that it can also be used to run other games. Examples of computational shells range from operating systems and abstraction layers, such as SDL, to specific tools written for a number of games in order to increase their portability, such as Z-machine used in classic Infocom text adventures. The split between a core and a shell matters for two reasons. First of all, a game can be rewritten to become shell-independent. Porting a game from Windows to Linux can be a good example of this. Second of all, in some cases the porting process consists only of the changes made to the shell and not to the game. Infocom games do not need to be altered to run on any machine as the only thing that has to be implemented is the Z-machine, which functions as a virtual machine which runs them. The proliferation of exotic Doom ports stems from an analogous situation. Doom can now be played on digital cameras, printers, ATMs and cars, but the programmers did not have to alter the code of the game. The only thing they needed to do was to port the computational shell (in this case SDL). In some ways, ports obtained via this method are closer to the process of emulation than to the processes of porting or conversion.

The last structural sub-layer we discern - **the mechanical layer** consists of the game systems and mechanics. The notion of mechanics we employ is Miguel Sicart's understanding of the term, according to which game's mechanics consist of the set of game state changing actions the agents in the game can perform. The last layer - **the presentation layer** should be fairly self-explanatory. It consists of all of the aesthetic aspects of the game: the way it looks and sounds and the narration it presents to the player. In order to codify the similarities and differences between the games we employ a simple scale of three possible grades:

Sameness

When two games are sharing a given layer completely, we indicate it by saying that the layer is the same. What it means in practice is that the games either run on the same platform, use the same physical interface, contain the same code, run within the same software environment, contain the same set of mechanics or present the same audio/visual effects.

Similarity

We will be saying that two games are similar with respect to a given layer if they share the layer at least partially. They can run on a platform of the same type, share physical interfaces of the same type, share parts of the same code or, more generally, the computational structure of the code (which can be, for example, represented in pseudocode). When two games share a similar computational shell, it means that the shell itself has been ported to a different physical platform (different Z-machine implementations can be considered as examples of this). Analogically, similarity of mechanics should be understood as sharing a part of the set of mechanics. Similarity of the presentation layer means that the game shares some aspects of its visuals and audio (for example the same shapes, sets of colors or the same tune). This way of understanding similarity allows for identity as its special case. The analogy between identity and similarity in our sense and between mathematical signs of = and ≥ might be of help here.

Resemblance

We say that two games are resemblant with respect to a given layer, whenever we wish to indicate that the games share a given layer, but never to the extent of sharing it completely (they cannot be identical). Resemblance should be understood as a sharp version of similarity - similarity which does not allow for identity.

Difference

We say that two games are different with respect to a given layer, whenever they are not similar. In other words, they are neither identical, nor resemblant. What it means in particular cases, is that the physical platform belongs to a different type, the physical interface belongs to a different type, the code is structured differently, the shell is completely new (it is not a port of the original shell), the sets of mechanics contain different actions; the graphics and sound do not display non-trivial similarities.

Using the above grades we can now codify two extreme cases: on one side we can imagine two games being identical with respect to all of the layers. In this case, the games should be treated simply as two copies of the same game. On the other side we can imagine two games which do not share any of the layers. In this case the games can be described as completely different. The model we present functions as a possibility space between both of these extremes. Using different combinations of the possible overlaps between the games' layers it is possible to obtain explications of the notion of a port (and other notions related to it). A list of example explications can be found in the table below.

	Physical		Structural			Presentation
	Platform	Interface	Comp. core	Comp. shell	Mechanics	
Hardware port	**Resemblant**	**Any**	**Similar**	**Any**	**Similar**	**Similar**
Software port	**Same**	**Any**	**Similar**	**Any**	**Same**	**Similar**
Virtualised port	**Resemblant**	**Any**	**Same**	**Resemblant**	**Same**	**Similar**
Interface port	**Resemblant**	**Different**	**Similar**	**Any**	**Same**	**Same**
Game conversion	**Resemblant**	**Any**	**Different**	**Any**	**Similar**	**Similar**
Remaster	**Resemblant**	**Any**	**Similar**	**Any**	**Same**	**Resemblant**
Graphical mod/ rom hack	**Same**	**Same**	**Similar**	**Same**	**Same**	**Resemblant**
Gameplay mod/ rom hack	**Same**	**Same**	**Resemblant**	**Same**	**Resemblant**	**Same**
Total conversion	**Same**	**Same**	**Resemblant**	**Same**	**Different**	**Different**
Reskin	**Same**	**Same**	**Same**	**Same**	**Same**	**Different**
Game clone	**Resemblant**	**Any**	**Different**	**Any**	**Same**	**Different**
Game emulation	**Resemblant**	**Any**	**Same**	**Any**	**Same**	**Similar**
Game H emulation	**Similar**	**Any**	**Same**	**Same**	**Same**	**Same**
Game adaptation	**Different**	**Different**	**Different**	**Different**	**Resemblant**	**Resemblant**

Table 1: Explication of the family of notions related to the notion of “game port”

As can be seen in the table, the notion of a port is simply ambiguous and can refer to at least four different cases. The first and the most typical case is a **hardware port**. A game can be said to be a hardware port of another game, when it contains code which structure is at least partially borrowed from the original; contains at least a subset of the original mechanics and contains some of the audio-visual aspects of the original. Additionally, it has to be recreated for a physical platform which is structurally resemblant. This means that the platform cannot be identical but it that it has to share some of the elements of the physical layers of the original. The best way to understand it is to point out that the platform has to at least be built around von Neumann architecture. Whether the interface and the computational shell remains the same, similar, resemblant or different, does not matter. It can be virtually anything as long as the programmers manage it to run the program with a similar computational core. This is indicated in the table with the label "Any".

To give an example of a hardware port in this sense, consider Doom produced for the Atari Jaguar. The physical platform and the physical interface has changed (the game is now played on a Jaguar gamepad). The game has been programmed by the author of the original, John Carmack which made it easier to adapt the code of the original. The computational shell changed, because Atari Jaguar uses a proprietary operating system (but still belonging to the same type of shells). The game retains all of its mechanics but changes the structure slightly as some of the levels have been redesigned in order to boost the performance. It retained its look although it cannot be described as identical, because the abilities of Jaguar to display more colours was utilised.

In some cases a port may be created for the same platform but intended for use with another computational core. We propose to call it a **software port.** A typical example of a port of this type can be found in Linux ports of Windows games or in the case of Windows 95 ports of DOS games. Software ports retain the mechanical layer but may display differences in presentation due to the differences between rendering capabilities of different computational shells. For example – the game may look differently on Linux because of the difference in graphics drivers.

In cases where the only thing that was actually ported was the computational shell it is possible to talk about **virtualised ports**. Running Windows games on Wine in Linux or the series of Infocom games we mentioned earlier can be good examples of ports of this type. Similarly to the case of a "software port", virtualised ports retain the full set of the mechanics of the original (as the computational core remains intact).

It might be also sometimes useful to discern a situation where the physical interface of the new version belongs to a different type of interfaces and because of that the game has to be rewritten. This category of **interface ports** can be used to describe VR ports of existing games (for example Skyrim VR (2017)) but it is not limited to this as many of the early arcade ports suffered from the inability to reproduce the original controls. Missile Command (1980)(which originally used a trackball) and Robotron 2049 (1982) (which used dual joysticks) are good examples of this.

Leaving the notion of "a port" aside, it is now possible to reconstruct the notion of a **game conversion** (as used by Švelch). The only difference between a conversion and a hardware port is that both games do not share parts of the computational core layer. Examples of conversions can be found amongst any of the games which have been written anew by programmers who did not have the access to the original code (or consciously decided not to use it). One particular consequence of this explication of the notion of a conversion is that in a highly unlikely case where the programmer of the new version unknowlingly replicated the structure of the original, it would be classified as a hardware port by our framework. What it means is that games containing extremely

simple structures, such as aforementioned Snake, are classified as hardware ports and not conversions.

An interesting observation which the model lets us to make is that if we wanted to express the notion of a "**remake**" within it, it would end up being synonymous with the notion of a conversion. This shows that "remakes" function rather as marketing labels - they are conversions which emphasize their status as conversions (as opposed to ports). This tactic is useful from the marketing standpoint, when the development team tries to point out that the game is rewritten (for example when the engine is changed), but does not make any difference from the academic or classificatory purposes.

Another notion that the model helps us to express is the notions of a **remaster**. According to our model a remaster can be seen as a version of the game which retains all of its mechanics but changes the presentation. Secret of Monkey Island Special Edition (2009) serves as a good example here. An ostentatious way in which the game stresses that the mechanics remained the same is that it enables the player to switch between the new and the old presentation at will, without interrupting the gameplay.

Two corresponding notions of graphical mods/rom hacks and gameplay mods/rom hacks can be represented in the following way: A game is a **graphical mod or graphical rom hack** of another game if it keeps the same physical platform, interface, computational shell and mechanics, but changes the presentational layer (so it is not identical). This change may be made via the change of the computational core, or by some other means (in these cases the core stays the same). Mods which change the rendering of the game, swap its textures, sprites or tile-sets can be used as an example.

A game can be described as **gameplay mod/rom hack**, whenever instead of the presentational layer, it changes the mechanics layer (it is not identical to the original anymore). An interesting consequence of this is that the computational core has to *resemble* the original, instead of being *similar* to it. The reason for this is that it is not possible to implement different mechanics in the game without changing the code. Mods which balance the game differently can be seen as an example of this.

A very specific case of modding, referred in the community as **total conversion** can also be represented in the model. This can be described as situations when both - the game mechanics and the presentation has been changed so much, that they no longer resemble the original. Aliens TC (1993) mod of Doom can serve as an example of this.

A very close overlap of the layers enables us to define the notion of a **reskin**. A game can be described as a reskin of another game if they share all of the layers apart from the presentation layer which is different in order to make the game superficially new. A famous example of a reskin in this sense is Super Mario Bros. 2 (1988) for the NES which was based on a previous game Yume Koujou: Doki Doki Panic (1987).

Another problematic concept which can be represented in our framework is the notion of a game **clone**. Clones differ from the general notion of a game port in that their computational core and presentation layer has to be different (otherwise the creators would have faced the accusation of plagiarism), but the set of mechanics stays the same. An example of a game clone can be found in Zuma (2003) which borrowed all of the mechanics from Puzz Loop (1988).

The model gives us also the possibility of expressing the notion of **emulation**. Emulation can be realized for all physical platforms which share von Neumann architecture with any interface and any computational shell, as long as the

computational core remains intact - after all, the whole point of emulation is that the same software can be used in an intact form. The mechanical layer has to stay the same because of the lack of change in the computational core. The presentational layer may change, depending on how the emulator processes the output. If additional filters are used, the game may not look identical to the original.

As can be seen from the table above, despite similarity of the name **hardware emulation** produces significantly different results. The specific case we have in mind is emulation achieved via field-programmable gate arrays which allow for a full reproduction of the hardware logic of a given physical platform. In these cases all structural and presentation aspects of the game are retained and the only allowed difference is the physical interface. The reason why the physical platform has been described as *similar* is that it typically does not completely recreate the original (although it is, in principle, possible).

Last but not least, the model allows us to describe **adaptations** of video games to non-digital platforms (such as turning them into board games or card games). According to the framework this process tries to retain the resemblance of both: presentational and mechanical layers using different physical platforms and physical interface (since they are not digital, they have to belong to a different type), using different computational cores and shells. To see this consider an example of the board adaptation of Starcraft. The game uses resemblant iconography, using completely different physical objects (such as a board, plastic tokens etc) and delegates all of the computation needed for the rules of the game to function to the cognitive capacities of the players which, as far as neurobiology tells us, has a different computational structure than the current digital computers.

CONCLUSION

The intended way the model is to be used is that it presents the users with a matrix of possible differences between game versions and maps them to popular terms, creating explications of these terms. The explications can then be used in further research or in classifications. It should nonetheless be stressed, that the actual mapping should be treated as a suggestion or an invitation for further discussion as one of the advantages of the presented framework is that it is very easy to modify. It is more of a toolset or a fine-grained language that is meant to start the discussion, than a ready-made typology set in stone. It is also important to remember, that the model does not evaluate game versions. It does not tell us if a given port should be considered as a “good” or a “proper” one (although it can indicate whether it is faithful). The reason for it is simple – these types of evaluations depend on the games we consider and the reasons we consider them. To give one example – if a research paper or an exhibition relates to Mortal Kombat (1992) – a game known for its realistic depiction of gore, they should prioritize presentational sameness over anything else. In contrast to this, if they consider Tetris, any version that retains the original mechanics may do. Different tasks and different game selections result in distinct priorities and it is not up to the model to dictate them.

ACKNOWLEDGMENTS

This project has received funding from the European Research Council (ERC) under the European Union’s H2020 ERC-ADG program (grant agreement No 695528)

BIBLIOGRAPHY

Ager, M. 2016, [Interview], *Retro Gamer 165*, Future Publishing

Altice, N.,2015. I Am Error. The Nintendo Family Computer / Entertainment System Platform, MIT Press

Atari, 1980, *Missile Command*, Arcade, Atari

Beam Software 1982, *The Hobbit*, ZX Spectrum, Melbourne House

Bethesda Softworks 2017, *Skyrim VR*, PlayStation VR, Bethesda Softworks

Blizzard Entertainment 1998. *Starcraft*. PC, Blizzard Entertainment

Capcom, 1987, *Bionic Commando*, Arcade, Capcom

Capcom, 1999, *Marvel vs Capcom*, Sony PlayStation, Capcom

Cinematronics 1983, *Dragon's Lair*, Arcade, Cinematronics

dotGear 2013, *Flappy Bird*, iOS, dotGear

Edwards, M. 2016, [Interview] *Retro Gamer 162*, Future Publishing

Fernández- Vara, C., Montfort, N. 2014, *Videogame Editions for Play and Study*, TROPE 13-02 (Cambridge, MA: MIT Trope Tank, June 5, 2014), 6, http:// dspace . mit. edu / handle / 1721 . 1 / 87668

Fisher, J., 1993, *Aliens TC*, PC, Fisher, J.

From Sofware 2011. *Dark Souls*. Playstation 3, Namco Bandai

Gerstmann, J., 1999, *Marvel vs. Capcom Review*, Gamespot, https://www.gamespot.com/reviews/marvel-vs-capcom-review/1900-2540395/ (retrieved 11.02.2019)

Gremlin, 1976, *Blockade*, Arcade, Gremlin

iD Software 1992, *Doom*, PC, id Software

Lucas Arts, 2009, *The Secret Of Monkey Island Special Edition*, Xbox 360, Lucas Arts

Midway Games, 1992, *Mortal Kombat*, Arcade, Acclaim

Mitchell Corporation, 1988, *Puzz Loop*, Arcade, Mitchell Corporation

Namco 1980, *Pac-Man*, Arcade, Namco

Newman James, 2012, *Best before : videogames, supersession and obsolescence*, Routledge 2012

Nintendo 1987, *Yume Koujou: Doki Doki Panic*, Famicom Disk System, Nintendo

Nintendo 1988, *Super Mario Bros 2*, Nintendo Entertainment System, Nintendo

Nintendo, 2017, *Splatoon 2*, Nintendo Switch, Nintendo

Nintendo 2018, *Super Smash Bros Ultimate*, Nintendo Switch, Nintendo

Nokia, 1997, *Snake*, Nokia 6110, Nokia

Oberon Media, 2003, *Zuma*, PC, PopCap Games

Pajitnov, A. 1984, *Tetris*, Elektronika 60, Pajitnov

Sega, 1987, *Thunder Blade*, Arcade, Sega

Stuckey, Helen, (2014), *Exhibiting The Hobbit: A tale of memories and microcomputers*, Kinephanos - History of Games International Conference Proceedings, 2014

Švelch, J. 2018. *Gaming the Iron Curtain. How Teenagers and Amateurs in Communist Czechoslovakia Claimed the Medium of Computer Games*, MIT Press.

Tanenbaum, A.S., Klint P., Bohm, W. 1978, *Guidelines for Software Portability*, Software: Practice and Experience 8, no. 6

Vid Kidz, 1982, *Robotron 2049*, Arcade, Williams Electronics