BIRMINGHAM CITY
University

CMP6200/DIG6200

Individual Undergraduate Project
2024–2025

A2: Literature Review and Methods

File Reverse engineering for the purpose of Game Preservation and Restoration to create an abstract tool for future Reverse engineering.

Course: Digital Games Technology
Student Name: Khai Ailyan
Student Number: 22130235
Supervisor Name: Jan Kraśniewicz

1. Report Introduction

This report is a review of the literature of games preservation, more specifically the decoding and recovering of old obsolete file formats, as a means of accessing their contents and recovering artifacts such as Textures, 3d Models and audio files etc.

The purpose of which is to avoid obsolete game files / games from becoming inaccessible/unusable, therefore these games and the syntax that underpins them will be forgotten and inaccessible.

1.1 Aims and Objectives

The aim of the project is to create software artifacts that decode and convert two file types to more contemporary file types and to then use these artifacts to create a proof of concept, a graphical learning tool to then help other programmers and reverse engineers reverse other file types.

The artifacts should take the old files, read its contents, then convert them into a newer format such as zip, txt, obj etc, and the graphical learning tool ought to include the step by step process in which a person can convert an old file type to a newer file type via software e.g reading the header file, understanding it, finding the offsets in the file etc.

Objectives include:

Conduct Primary And Secondary research - Research already existing syntax and documentation of .Str and .Arc files (Secondary Research) then research and understand not already publicly available syntax of .Str and .Arc files (Secondary Research)

Create File readers - Create two pieces of software that read in information from .Str and .Arc files without syntactical failure.

Create File writers - Create two pieces of software that writes the now read information into a contemporary format e.g txt, zip file etc

Create Digraphic Tool - Create a diagram that illustrates the exact features of game file reverse engineering as well as a chronological list of steps to take to reverse a file.

1.2 Literature Search Methodology

The literature search comprises the processes of decoding files, files preservation techniques, ADT structures and syntax as well as Games preservation as a whole.

Search Terms:	Rationale
File syntax File structure	Each file that is to be decoded to then reverse engineer has as specific structure and syntax, so understanding how they work is crucial
Reverse Engineering Software Reverse Engineering	Understanding the techniques that underpin reversing software and files.
Games Preservation Files Preservation	Understanding and techniques used in preserving games and files
.Rws Models .Rws Object Formats .Rw Object syntax Renderware Object Files	Specifically relating to the.Str file artifact. EA were using these formats to create their models and objects
str file format .str file syntax Ps3 stream file syntax Ea stream file syntax Ea object file syntax	Specifically relating to .Str file artifact, EA were using these formats as storage for all of their games file.

2. Literature Review

2.1 Themes

File Syntax - denotes the rules and structures that data is stored as in a file and is a fundamental to reverse engineering files, an example of which would be .Obj where all of the vertices are denoted with “v” and the vertex normals “vn”, the relevance to the project being file syntax is needed because the .Str and .Arc files structure is unknown, and needs to be known to then create a file converter.

Context drives what type of abstract data structures and structures the file uses in its syntax, for instance 32 bit architectures such as operating windows operating systems that are 32 bit, they will have smaller integer sizes compared to 64 bit operating systems.

Reverse Engineering - Reproducing a proprietary product (i.e files, code and data formats) via the understanding of syntax, this involves understanding the **File syntax** of the specific file to then reverse engineer OR to decompiler the software code in order to convert it to another language or for a newer system.

The ethics and legality is purely contextual, due to many aspects needing to be present for the company of the proprietary software to care such as, the age of the software, commercial impact, whether its used commercially, whether it infringes on copyright or not (this is because if a file reader is made, are said files that are read then used for future use, that would cause the company to actually care).

Games preservation - Preserving and maintaining games either through physical maintenance and or protection OR through the preservation of the games software, code and or files.

In this context it is converting their respective contents to more contemporary file formats to make it so future operating systems and systems in general can use / play those games.

ADT (Abstract Data Type) Syntax - Bytes in code with specific structures and rules created for specific purposes, e.g List, contiguous block of memory containing a list of items of a specific type, does not need to be redefined to be bigger than what it already is.

Compression Algorithms - Algorithms created to make data and files smaller than they actually are for transportation between devices or storage capacity optimisation, compressed items can then be uncompressed to their original size for further use,

2.2 Review of Literature

2.2.1 Review -

Games Preservation:

Games preservation has existed as long as games have needed to be stored, as time progressed the games software separate to its hardware needed to be maintained (Harkai Istvan 2022)(Haydock Christopher 2018) as well as the hardware (Dany Guay-belanger, may 2021)(Widget, Megan A, 2009) , moreover the importance of games preservation cannot be understated due to its cultural impact (Todd Benjamin C 2019)(Henry Lowood 2009) as games and games systems are being threatened with obsolescence such as the discontinuation of ps3.

Moreover the importance is shown with people's desire to preserve games, but can't due to legal restraints (Bachel, Alasdair and matthew barr 2014). Most games preservation today is hardware based and comes in the form of “Archiving” (Camila johansson 2023).

There lacks research on a software approach to Games preservation, most of the research refers to hardware and the minimal research that refers to software rarely accounts for the legal issues, especially when it comes to international law, not just america.

Reverse Engineering:

Reverse Engineering's existence dates to the first thing ever made. However in the context of software, software reverse engineering began in 1975 with the first text editor WYLBUR (Kathi Hogshead Davis, Peter H Aiken, 2000), features of reverse engineering include disassembler, debugger, etc (Abigail A 2021), whilst in contrast (Alessandro Mantovani , Simone Aonzo, Yanick Fratanatonio), refer to the techniques of reverse engineering through their behavior and performance in time.

(Anand Gadwal 2011) talks in detail on the step by step process of reversing software from “context parsing” through to “Design Reconstructing Phase”, (Ramandeep singh 2013) refers to the types of reverse engineering such as “data” and “code” reverse engineering, detailing the different types depending on what is being reversed.

There needs to be a greater focus on breaking down the individual processes, with example of high level reverse engineering in software, most of the research refers to very low level or do break down the steps enough, as well as on **Data Reverse Engineering** specifically.

Abstract Data Types:

Abstract Data types or ADT’s have existed since the 70’s created by (Barbara Liskov, 1974), a standard example of an adt being a “Stack” (John V Guttag 1977) which contains items of a certain data type and allows for the storage of multiple items in a first in last out order of execution.

In Contrast to ADO’s (Abstract Data objects) which are more raw and have little functionality (Jean Francois, Rainer Koschke 2000), Vectors in games would be an ADO, another ADT a Queue is commonly used for AI behavior in games (John v Guttag June 1977).

Limitations are that the general research goes back to the 60’s and 70’s and the majority of the ADT’s that are common place are incredibly old, whilst the research does show the process of them being made and so could be replicated, they are dated.

Compression:

Data Compression’s origins come from (Robert Fano, Claude Shannon 1949), then (Huffman Coding 1952) with “Huffman Coding” and (Lempel-ziz-welch) with LZW, Different compression algorithms are used for specific files and data structures (Tito Waluyo Puroboyo 20,17). This in turn leads to the most common use of compression algorithms, compressing text (Amandeep Singh Sidhy 2014).

Custom Compression algorithms are created for specific data structures and object types like model files (Mustafa, Ammar 2017; JingLiang Ping 2005; siddeeq mohammed m, rodrigues marcos 2016), said algorithms are specialized to be better optimized for 3d objects.

There lacks research into more proprietary compression algorithms, companies often make their own, as a result these compression algorithms that have been documented and research are often non-applicable for reverse engineering and thus for games preservation.

File Syntax:

File syntax originated in (Russel Kirsch 1957) via a picture of his son, and from that the first file syntax was created alongside it having its own rule set. File syntax varies between different types of files due to each type of file having a different purpose (Kauthar Abdulazeez, sohit agarwal 2021; Samiya Khan;Mansaf Allam 2019), coupled with different structures inside different files (Mike Folk 2003) each type of file is unique in its rules, structure and therefore purpose.

The limitations are that there are little research documents/journals that go into proprietary formats and more look into general syntax and older simpler formats like text or image files.

2.2.2 Theory -

Games Preservation:

Games Preservation includes the physical storage of games and their associated hardware (Haydock, Christopher april 2018) (Dany Guay-belanger may 2021), it includes the maintenance of games/games systems themselves through maintaining hardware or Copying contents of games to newer external storage due to bit rot (Haydock, Christopher april 2018), bit rot being when data is stored in a harddrive for too long, the data deteriorates.

Also includes the maintenance of software via reverse engineering the games/games systems software to accommodate newer hardware. (Camilla Johansson, spring semester 2023), so that it can be played on contemporary systems. This is the approach used in the project, software maintenance is the method of which the game/game files will be preserved, not hardware maintenances.

Reverse Engineering:

Reverse Engineering in a general sense is when a product/system has been understood mostly in its entirety and recreated.

However in this context, Reverse Engineering consists of understanding the syntax of code, a file, hardware architecture/structure (Aremo, Adeyinka Abigail, 2021) etc and trying to recreate it, it's about understanding and recreation.

In this context its about understanding the header file of a file, reading the values inside of the header to ascertain data about the file, locating offsets in the file to determine where to go to in the file and where to stop, reading the header file to determine key features of the file such as, the amount of files, of what memory policy/allocation, what structures exist within the file so forth.

From this, creating code (an artifact) to then read the file and write it or its contents into more understandable/contemporary sub-files, if applicable.

Abstract Data Types:

Abstract Data types allow for data to be stored with specific functionality depending on what ADT it is, a list allows for continuous addition of items of a type, and once fully allocates more memory to the list for further use, arrays are finite and need redeclaring if the array is to full, sets are immutable and so the values inside cannot be modified directly.

For this project ADT's are necessary due to a file effectively being a contiguous block of memory that functions like an array/list, therefore reading line by line and assigning to an array or list variable ADT is crucial in creating an artifact to read/write files.

Compression:

Compression is the practice of taking in data and making it smaller for the sake of storage and transport.

Compression is either lossy, data is lost in compression , first lossy algorithm (Nasir ahmed 1974), or is lossless and so data is not lost on compression, the first and commonly used LZW (Lempel-ziv-welch).

Depending on the type of data certain compression algorithms are more ideal e.g audio, model files, textures etc (Tito waluyo Purboyo, anggunmeka luhur prasasti 2017; Amandeep Singh Sidhu 2014), this is especially true with model files that have specific compression algorithms ideal for triangles (Mustafa, Ammar, 2017; Mohammed m Rodrigues 2016; Jingliang Ping 2005).

File syntax:

File syntax is the structures and rules of the file itself normally shown through the use of a "header file", which is at the top of the file, file syntax is important as by understanding its structures and rules a programmer can read its data correctly without mishap.

Understanding file syntax is crucial for extracting models, audio files, textures etc from games in order to reverse engineer them as the programmer is then able to determine where they are in the file, there size etc.

2.3 Summary

Games Preservation:

Games preservation in the form of hardware preservation is seemingly a problem that has already been fixed, in terms of there is a legal viable solution, however from a software maintenance approach there seems to still be a lack of information and problems with legality, making the project more important due to a lack of emphasis on both software maintenance as well as a more legal avenue.

Reverse Engineering:

The research shows the methodologies and types of reverse engineering, as a result there is a detailed blueprint as to what type of reverse engineering is to be used and the steps to which software can be reversed.

Furthermore for the project, it is data reverse engineering, using text editing tools, reading offsets in memory to memory and identify key structures and repetition in the files.

Abstract Data Types:

Abstract data types are shown to be useful containers for data with additional functionality, artifacts will have to be read as arrays to process line by line characters for binary analysis and data reverse engineering

File syntax:

File syntax research has shown the importance of header files due to their relation to the rest of the file, moreover file syntax is different for each file type, given this information the project going forward requires better understanding of custom files due to their custom syntax as shown by their respective header files, this will be crucial in decoding the files.

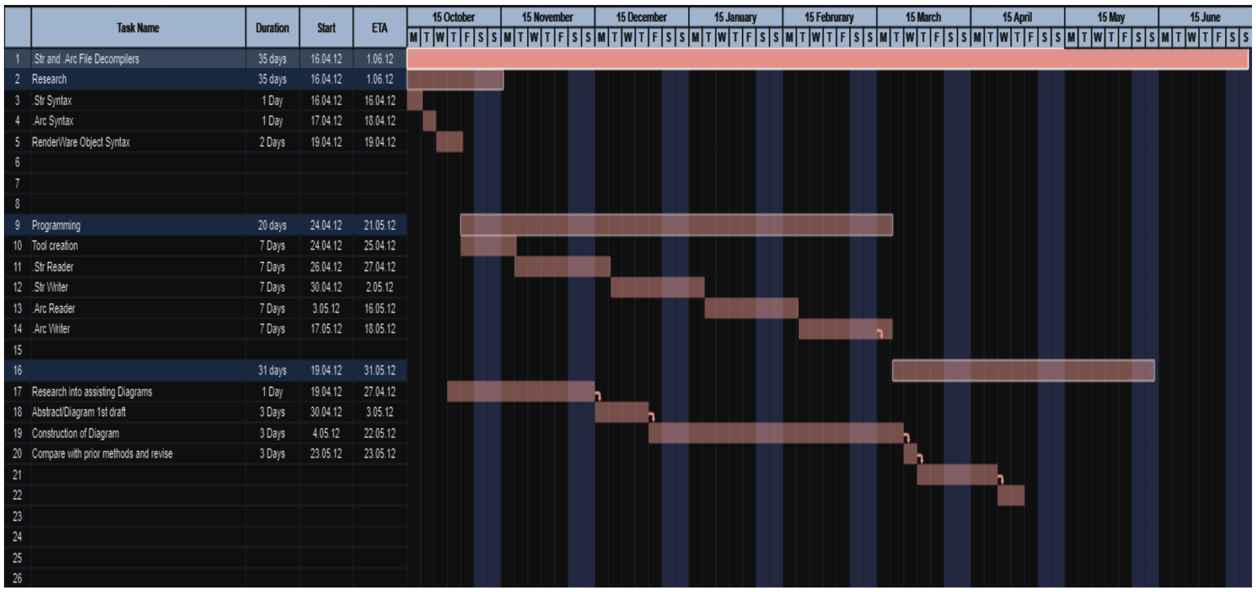
Compression:

The research can be summarised as each data type and or file having its own ideal compression algorithm designed to deal with that specific file, this is important because companies tend to design their own file systems and as a result have their own compression algorithms to better optimise space efficiency.

Said algorithm needs to be understood before reverse engineering.

3. Appendix

3.1 Gantt Chart



	Task Name	Duration	Start	ETA
1	.Str and .Arc File Decompilers	35 days	16.04.12	1.06.12
2	Research	35 days	16.04.12	1.06.12
3	.Str Syntax	1 Day	16.04.12	16.04.12
4	.Arc Syntax	1 Day	17.04.12	18.04.12
5	RenderWare Object Syntax	2 Days	19.04.12	19.04.12
6				
7				
8				
9	Programming	20 days	24.04.12	21.05.12
10	Tool creation	7 Days	24.04.12	25.04.12
11	.Str Reader	7 Days	26.04.12	27.04.12
12	.Str Writer	7 Days	30.04.12	2.05.12
13	.Arc Reader	7 Days	3.05.12	16.05.12
14	.Arc Writer	7 Days	17.05.12	18.05.12
15				
16		31 days	19.04.12	31.05.12
17	Research into assisting Diagrams	1 Day	19.04.12	27.04.12
18	Abstract/Diagram 1st draft	3 Days	30.04.12	3.05.12
19	Construction of Diagram	3 Days	4.05.12	22.05.12
20	Compare with prior methods and revise	3 Days	23.05.12	23.05.12
21				
22				
23				
24				
25				
26				

4. References

Tito waluyo purboyo, 2017, a review of data compression technqiques, international journal of applied engineering research, 8596 - 8963, (PDF) A review of data compression techniques.

SIDDEQ, mohammed M and Rodrigues, marcos, 2016, Novel 3D compression methods for geometry connectivity and texture, Sheffield hallam university research archive, 1 - 16, Microsoft Word - Research_No_7_MR_final.docx

Mustafa ORAL, Ammar abbas Elmas, 2017, A brief history of 3d mesh compression, Cukorva university, 1-6, ,
https://www.researchgate.net/publication/327905583_A_Brief_History_of_3D_Mesh_Compression

Amandeep Singh Sidhu, 2014, Research paper on text data compression algorithm using hybrid approach, International Journal of Computer Science and Mobile Computing,volume 3 issue 12, 1 - 10, https://ijcsmc.com/docs/papers/December2014/V3I12201404.pdf.

Sjöstrand, M. (2005). *A study in compression algorithms*. https://www.diva-portal.org/smash/get/diva2:830266/FULLTEXT01.pdf.

Peng, J., Kim, C.-S. and Jay Kuo, C.-C. . (2005). Technologies for 3D mesh compression: A survey. *Journal of Visual Communication and Image Representation*, 16(6), pp.688–733. doi:https://doi.org/10.1016/j.jvcir.2005.03.001.

Girard, J.-F. and Koschke, R. (2000). A comparison of abstract data types and objects recovery techniques. *Science of Computer Programming*, 36(2-3), pp.149–181. doi:https://doi.org/10.1016/s0167-6423(99)00035-0.

Guttag, J.V., Horowitz, E. and Musser, D.R. (1978). Abstract data types and software validation. *Communications of the ACM*, 21(12), pp.1048–1064. doi:https://doi.org/10.1145/359657.359666.

Guttag, J. (1977). Abstract data types and the development of data structures. *Communications of the ACM*, 20(6), pp.396–404. doi:https://doi.org/10.1145/359605.359618.

Liskov, B. and Zilles, S. (1974). Programming with abstract data types. *ACM SIGPLAN Notices*, 9(4), pp.50–59. doi:https://doi.org/10.1145/942572.807045.

Musser, D.R. (1980). On proving inductive properties of abstract data types. doi:https://doi.org/10.1145/567446.567461.

Bertoni, A., Mauri, G. and P. Miglioli (2006). Towards a theory of abstract data types: A discussion on problems and tools. *Lecture Notes in Computer Science*, [online] pp.44–58. doi:https://doi.org/10.1007/3-540-09981-6_4.

A, A. (2021). *Reverse Engineering Research*. [online] doi:https://doi.org/10.13140/RG.2.2.28030.51520.

Mantovani, A., Aonzo, S., Fratantonio, Y., Talos, C. and Balzarotti, D. (n.d.). *RE-Mind: a First Look Inside the Mind of a Reverse Engineer*. [online] Available at: https://www.usenix.org/system/files/sec22summer_mantovani.pdf.

Jain, A., Swapnil Soner and Anand Gadwal (2011). Reverse engineering: Journey from code to design. doi:https://doi.org/10.1109/icectech.2011.5941966.

Singh, R. (2013). A Review of Reverse Engineering Theories and Tools. [online] 2, pp.35–38. Available at: https://www.ijesi.org/papers/Vol(2)1/G213538.pdf.

Cipresso, T. and Stamp, M. (2010). Software Reverse Engineering. *Handbook of Information and Communication Security*, pp.659–696. doi:https://doi.org/10.1007/978-3-642-04117-4_31.

Müller, H.A., Jahnke, J.H., Smith, D.B., Storey, M.-A., Tilley, S.R. and Wong, K. (2000). Reverse engineering. *Proceedings of the conference on The future of Software engineering - ICSE '00*. doi:https://doi.org/10.1145/336512.336526.

Google Books. (2024). *Reversing*. [online] Available at: https://books.google.co.uk/books?hl=en&lr=&id=_78HnPPRU_oC&oi=fnd&pg=PR23&dq=reverse+engineering+software&ots=EP1MLkmRVn&sig=stA7p1aP8xPSkUcFJi6jOCmA_Is&redir_esc=y#v=onepage&q=reverse%20engineering%20software&f=false [Accessed 25 Nov. 2024].

Davis, K.L. and Alken, P.H. (2002). Data reverse engineering: a historical survey. doi:https://doi.org/10.1109/wcre.2000.891454.

Bachell, A. and Barr, M. (2014). Video Game Preservation in the UK: A Survey of Records Management Practices. *International Journal of Digital Curation*, 9(2), pp.139–170. doi:https://doi.org/10.2218/ijdc.v9i2.294.

Davide V (2024). *GTA-Modding.com - Download Area» GTA San Andreas» Tools» RW Analyze*. [online] Gta-modding.com. Available at: https://www.gta-modding.com/area/file-33-rw-analyze.html [Accessed 25 Nov. 2024].

Winget, M.A. and Murray, C. (2008). Collecting and preserving videogames and their related materials: A review of current practice, game-related archives and research projects. *Proceedings of the American Society for Information Science and Technology*, 45(1), pp.1–9. doi:https://doi.org/10.1002/meet.2008.1450450250.

Todd, B. and Hopkins, J. (2019). *Running head: PRESERVING VIDEO GAME SIGNIFICANCE Preserving Video Game Significance: A Practical Guide for Video Game Preservation, Exhibition, and their Significant Properties*. [online] Available at: https://jscholarship.library.jhu.edu/server/api/core/bitstreams/49d19d02-a439-4a7d-8243-a685429cbcac/content.

Guay-Bélanger, D. (2021). Assembling Auras: Towards a Methodology for the Preservation and Study of Video Games as Cultural Heritage Artefacts. *Games and Culture*, p.155541202110203. doi:https://doi.org/10.1177/15554120211020381.

Digra.org. (2024). *View of Before It's Too Late: Preserving Games across the Industry / Academia divide*. [online] Available at: https://dl.digra.org/index.php/dl/article/view/468/468 [Accessed 25 Nov. 2024].

Johansson, C. and Koenitz, H. (n.d.). *Video Game Preservation and Emulation from Three Perspectives: Developers, Archivists and Gamers Video Game Preservation and Emulation from Three Perspectives: Developers, Archivists and Gamers*. [online] Available at: http://sh.diva-portal.org/smash/get/diva2:1807915/FULLTEXT02.pdf.

Haydock, C. (2018). *Challenges in Preserving Video Games*. [online] Carolina Digital Repository. Available at: https://cdr.lib.unc.edu/concern/masters_papers/fn107276t [Accessed 25 Nov. 2024].

Folk, M. and Barkstrom, B.R. (2003). *Attributes of file formats for long-term preservation of scientific and engineering data in digital libraries*. [online] Available at: https://www.researchgate.net/publication/228726593_Attributes_of_file_formats_for_long-term_preservation_of_scientific_and_engineering_data_in_digital_libraries.

THE DEFINITIVE GUIDE TO EXPLORING FILE FORMATS Mr.Mouse and WATTO. (n.d.). Available at: https://www.gamedevs.org/uploads/the-definitive-guide-to-exploring-file-formats.pdf [Accessed 25 Nov. 2024].

File Formats for Big Data Storage Systems. (2019). *International Journal of Engineering and Advanced Technology*, 9(1), pp.2906–2912. doi:https://doi.org/10.35940/ijeat.a1196.109119.

Zhenhua, W. (2018). Design and Research of an Image File Format with Rich Information. *Journal of Electrical and Electronic Engineering*, 6(2), p.71. doi:https://doi.org/10.11648/j.jeee.20180602.16.

Ontology of Heterogeneous Image File Formats and their Disparate Applications. (2021). *International Journal of Advanced Trends in Computer Science and Engineering*, 10(6), pp.3138–3143. doi:https://doi.org/10.30534/ijatcse/2021/091062021.

Dinneen, J.D. and Julien, C.-A. (2021). *The ubiquitous digital file: A review of file management research*. [online] doi:https://doi.org/10.48550/arXiv.2109.09668.

Underwood, W. (2012). Grammar-Based Specification and Parsing of Binary File Formats. *International Journal of Digital Curation*, 7(1), pp.95–106. doi:https://doi.org/10.2218/ijdc.v7i1.217.

Fred brooks, Data definition and file syntax for ISO/TS 14048 data exchange with data storage format based on ISO/TS 14048, RAUL CARLSON JOHAN TIVANDER, CHALMERS UNIVERSITY OF TECHNOLOGY, 2001, https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=676dc77da1217c8cdd1b61d76c83cad530bc3159

5. Bibliography

Tito waluyo purboyo, 2017, a review of data compression technqiques, international journal of applied engineering research, 8596 - 8963, (PDF) A review of data compression techniques.

SIDDEQ, mohammed M and Rodrigues, marcos, 2016, Novel 3D compression methods for geometry connectivity and texture, Sheffield hallam university research archive, 1 - 16, Microsoft Word - Research_No_7_MR_final.docx

Mustafa ORAL, Ammar abbas Elmas, 2017, A brief history of 3d mesh compression, Cukorva university, 1-6, , https://www.researchgate.net/publication/327905583_A_Brief_History_of_3D_Mesh_Compression

Amandeep Singh Sidhu, 2014, Research paper on text data compression algorithm using hybrid approach, International Journal of Computer Science and Mobile Computing,volume 3 issue 12, 1 - 10, https://ijcsmc.com/docs/papers/December2014/V3I12201404.pdf.

Sjöstrand, M. (2005). *A study in compression algorithms*. https://www.diva-portal.org/smash/get/diva2:830266/FULLTEXT01.pdf.

Peng, J., Kim, C.-S. and Jay Kuo, C.-C. . (2005). Technologies for 3D mesh compression: A survey. *Journal of Visual Communication and Image Representation*, 16(6), pp.688–733. doi:https://doi.org/10.1016/j.jvcir.2005.03.001.

Peng, J., Kim, C.-S. and Jay Kuo, C.-C. . (2005). Technologies for 3D mesh compression: A survey. *Journal of Visual Communication and Image Representation*, 16(6), pp.688–733. doi:https://doi.org/10.1016/j.jvcir.2005.03.001.

Girard, J.-F. and Koschke, R. (2000). A comparison of abstract data types and objects recovery techniques. *Science of Computer Programming*, 36(2-3), pp.149–181. doi:https://doi.org/10.1016/s0167-6423(99)00035-0.

Guttag, J.V., Horowitz, E. and Musser, D.R. (1978). Abstract data types and software validation. *Communications of the ACM*, 21(12), pp.1048–1064. doi:https://doi.org/10.1145/359657.359666.

Guttag, J. (1977). Abstract data types and the development of data structures. *Communications of the ACM*, 20(6), pp.396–404. doi:https://doi.org/10.1145/359605.359618.

Liskov, B. and Zilles, S. (1974). Programming with abstract data types. *ACM SIGPLAN Notices*, 9(4), pp.50–59. doi:https://doi.org/10.1145/942572.807045.

Musser, D.R. (1980). On proving inductive properties of abstract data types. doi:https://doi.org/10.1145/567446.567461.

Bertoni, A., Mauri, G. and P. Miglioli (2006). Towards a theory of abstract data types: A discussion on problems and tools. *Lecture Notes in Computer Science*, [online] pp.44–58. doi:https://doi.org/10.1007/3-540-09981-6_4.

A, A. (2021). *Reverse Engineering Research*. [online] doi:https://doi.org/10.13140/RG.2.2.28030.51520.

Mantovani, A., Aonzo, S., Fratantonio, Y., Talos, C. and Balzarotti, D. (n.d.). *RE-Mind: a First Look Inside the Mind of a Reverse Engineer*. [online] Available at: https://www.usenix.org/system/files/sec22summer_mantovani.pdf.

Jain, A., Swapnil Soner and Anand Gadwal (2011). Reverse engineering: Journey from code to design. doi:https://doi.org/10.1109/icectech.2011.5941966.

Singh, R. (2013). A Review of Reverse Engineering Theories and Tools. [online] 2, pp.35–38. Available at: https://www.ijesi.org/papers/Vol(2)1/G213538.pdf.

Cipresso, T. and Stamp, M. (2010). Software Reverse Engineering. *Handbook of Information and Communication Security*, pp.659–696. doi:https://doi.org/10.1007/978-3-642-04117-4_31.

Müller, H.A., Jahnke, J.H., Smith, D.B., Storey, M.-A., Tilley, S.R. and Wong, K. (2000). Reverse engineering. *Proceedings of the conference on The future of Software engineering - ICSE '00*. doi:https://doi.org/10.1145/336512.336526.

Google Books. (2024). *Reversing*. [online] Available at: https://books.google.co.uk/books?hl=en&lr=&id=_78HnPPRU_oC&oi=fnd&pg=PR23&dq=reverse+engineering+software&ots=EP1MLkmRVn&sig=stA7p1aP8xPSkUcFJi6jOCmA_Is&redir_esc=y#v=onepage&q=reverse%20engineering%20software&f=false [Accessed 25 Nov. 2024].

Davis, K.L. and Alken, P.H. (2002). Data reverse engineering: a historical survey. doi:https://doi.org/10.1109/wcre.2000.891454.

Bachell, A. and Barr, M. (2014). Video Game Preservation in the UK: A Survey of Records Management Practices. *International Journal of Digital Curation*, 9(2), pp.139–170. doi:https://doi.org/10.2218/ijdc.v9i2.294.

Davide V (2024). *GTA-Modding.com - Download Area» GTA San Andreas» Tools» RW Analyze*. [online] Gta-modding.com. Available at: https://www.gta-modding.com/area/file-33-rw-analyze.html [Accessed 25 Nov. 2024].

Winget, M.A. and Murray, C. (2008). Collecting and preserving videogames and their related materials: A review of current practice, game-related archives and research projects. *Proceedings of the American Society for Information Science and Technology*, 45(1), pp.1–9. doi:https://doi.org/10.1002/meet.2008.1450450250.

Todd, B. and Hopkins, J. (2019). *Running head: PRESERVING VIDEO GAME SIGNIFICANCE Preserving Video Game Significance: A Practical Guide for Video Game Preservation, Exhibition, and their Significant Properties*. [online] Available at: https://jscholarship.library.jhu.edu/server/api/core/bitstreams/49d19d02-a439-4a7d-8243-a685429cbcac/content.

Guay-Bélanger, D. (2021). Assembling Auras: Towards a Methodology for the Preservation and Study of Video Games as Cultural Heritage Artefacts. *Games and Culture*, p.155541202110203. doi:https://doi.org/10.1177/15554120211020381.

Digra.org. (2024). *View of Before It's Too Late: Preserving Games across the Industry / Academia divide*. [online] Available at: https://dl.digra.org/index.php/dl/article/view/468/468 [Accessed 25 Nov. 2024].

Johansson, C. and Koenitz, H. (n.d.). *Video Game Preservation and Emulation from Three Perspectives: Developers, Archivists and Gamers Video Game Preservation and Emulation from Three Perspectives:*

Developers, Archivists and Gamers. [online] Available at: http://sh.diva-portal.org/smash/get/diva2:1807915/FULLTEXT02.pdf.

Haydock, C. (2018). *Challenges in Preserving Video Games*. [online] Carolina Digital Repository. Available at: https://cdr.lib.unc.edu/concern/masters_papers/fn107276t [Accessed 25 Nov. 2024].

Folk, M. and Barkstrom, B.R. (2003). *Attributes of file formats for long-term preservation of scientific and engineering data in digital libraries*. [online] Available at: https://www.researchgate.net/publication/228726593_Attributes_of_file_formats_for_long-term_preservation_of_scientific_and_engineering_data_in_digital_libraries.

THE DEFINITIVE GUIDE TO EXPLORING FILE FORMATS Mr.Mouse and WATTO. (n.d.). Available at: https://www.gamedevs.org/uploads/the-definitive-guide-to-exploring-file-formats.pdf [Accessed 25 Nov. 2024].

File Formats for Big Data Storage Systems. (2019). *International Journal of Engineering and Advanced Technology*, 9(1), pp.2906–2912. doi:https://doi.org/10.35940/ijeat.a1196.109119.

Zhenhua, W. (2018). Design and Research of an Image File Format with Rich Information. *Journal of Electrical and Electronic Engineering*, 6(2), p.71. doi:https://doi.org/10.11648/j.jeee.20180602.16.

Ontology of Heterogeneous Image File Formats and their Disparate Applications. (2021). *International Journal of Advanced Trends in Computer Science and Engineering*, 10(6), pp.3138–3143. doi:https://doi.org/10.30534/ijatcse/2021/091062021.

Dinneen, J.D. and Julien, C.-A. (2021). *The ubiquitous digital file: A review of file management research*. [online] doi:https://doi.org/10.48550/arXiv.2109.09668.

Underwood, W. (2012). Grammar-Based Specification and Parsing of Binary File Formats. *International Journal of Digital Curation*, 7(1), pp.95–106. doi:https://doi.org/10.2218/ijdc.v7i1.217.

Fred brooks, Data definition and file syntax for ISO/TS 14048 data exchange with data storage format based on ISO/TS 14048, RAUL CARLSON JOHAN TIVANDER, CHALMERS UNIVERSITY OF TECHNOLOGY, 2001, https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=676dc77da1217c8cdd1b61d76c83cad530bc3159

Greavesy1899 (2023). *Resource Types*. [online] GitHub. Available at: https://github.com/Greavesy1899/VisceralToolkit/wiki/Resource-Types [Accessed 25 Nov. 2024].

Greavesy1899 (2024). *Releases · Greavesy1899/VisceralToolkit*. [online] GitHub. Available at: https://github.com/Greavesy1899/VisceralToolkit/releases/ [Accessed 25 Nov. 2024].