**Packaging and streaming assets to decrease the memory footprint in video games**

Jarryd Hall, Taher Odeh

**ABSTRACT**

Memory related issues such as memory management, limited memory and memory related application crashes, etc. often affect video games. Private game development companies have created the previous solutions to these kinds of issues and there are no open source solutions available for developers to utilize. This paper presents a constructive research approach to solving these memory issues. Operating systems were studied in order to properly understand these memory related issues and determine a viable solution. The solution needed to be implemented in such a way that multiple platforms could be targeted, although for the scope of this paper the focus was on iOS and Mac OSX. The solution

**General Terms**

video games, memory management, virtual memory, cross-platform, game performance, optimization

**Keywords**

virtual memory, mmap, Packaging tool, API, compression, caching, RAM, HDD, memory enhancement, file-system

**1. INTRODUCTION**

Almost all video games are bound by memory constraints. The system running the game can only allocate a certain amount of usable memory for the game’s process and this is usually less than advanced games need. The system’s kernel running the game will allocate memory for the game on RAM as well as on the HDD, which is where the virtual memory is located, and often the assets and game resources are very large, consuming a lot of the available memory.  
  
These memory constraints effects game developers and designers working within the development process of a project.

There are techniques used to solve this memory issue, but they are privately owned and with little documentation surrounding how the memory problems are solved. This study set out to solve the issue and implement an open source solution so that indie game developers who are not aided by proprietary techniques can create better games. Implementing a solution for this problem and licensing it as open source will allow anyone to be able to increase game performance as well as improve the overall game design and development.  
  
The problem above was pointed out during an initial interview with Johan Knutzen, the founder of Senri and Phobic-Games, mobile application and game development companies based in Gothenburg, Sweden, which are strongly involved in this research paper and it’s requirements and solutions.

The solution that was developed included a number of elements. The packaging tool compresses all the assets for a game into a single pak file. The pak file format is a file format that contains archived data, which is either compressed or decompressed. The API allows developers to use the generated pak file, as well as handles the memory optimization.

In order to test the developed solution, a sample game was run numerous times with and without the solution present. The game was profiled and monitored for memory and performance variables and the data was compared.

The question researched in this paper was:

* How can the memory footprint of video games be decreased?

**Constructive Research Method**

[What is a constructive research method]

The method chosen to be followed in this research is the Constructive Research Method, [ref1] describes this method as a way to turn existing knowledge into novelty or innovation by implementing a solution, to an existing problem, whether it is theoretical or practical, as long as it involves the usage of preexisting knowledge and thinking to produce artifact design solutions, for example, plans, diagrams, charts or software implementation; it may be considered constructive.

In addition to the description above, Kasanen et al. (1991) summarizes this method as a solution oriented method where innovation step-by-step solutions are taken in account, followed by testing of the solution and using the data within the testing phrase for analysis purposes.

[ref3] provides 3 category examples of knowledge gaps to be filled using the constructive research method: feasibility, where a solution to a common problem has not been done yet; Novelty, where a unique and new solution is provided to an already solved problem; or an improvement, where the goal of the research focuses on a preexisting solution and aims to produce better results than the ones available.

[why constructive]

The method mentioned has been found the most suitable for this research, basing on the fact that the latter aims to cover a feasible constructivism in terms of finance and freedom by having the solution free of charge and also freely licensed. Additionally, [ref4] argues that the constructive research method befits the computer science and the IT related problems in a usual manner.

The constructive research methodology allows for qualitative and quantitative approaches to be mixed or used as needed according to the kind of needed to be gathered during the development process of the solution.

Steps ….

“According to Kasanen et al. (1993), Lukka (2000), Labro and Tuomela (2003), there are seven crucial steps in the constructive research approach:

* (1)  to find a practically relevant problem, which also has research potential;
* (2)  to examine the potential for long-term research co-operation with the target organisation;
* (3)  to obtain a general and comprehensive understanding of the topic;
* (4)  to innovate and construct a theoretically grounded solution idea;
* (5)  to implement the solution and test whether it works in practice;
* (6)  to examine the scope of the solution’s applicability; and
* (7)  to show the theoretical connections and the research contribution of the solution. “

NOTE: This needs to be referenced properly from the original source.

[framework]

**References:**

[ref1 = **Constructivist Research and Info-Computational Knowledge Generation** Gordana DODIG CRNKOVIC]

[ref2 = Kasanen et al. (1991)]

[ref3] Anna-Liisa Lindholm

[ref4]= Information systems research methodologies and models(Albertas Caplinskas, Olegas Vasilecas)