**Memory Management Assignment**

**Tammy Hartline**

**Week 6 CS-230**

**04/08/2023**

This paper’s purpose is to show that I understand memory management and optimization concepts, as they relate to software and application development run-time and performance. In this instance, the example scenario assigned is for a client, *The Gaming Room*, who is seeking to expand their current game application, “*Draw It or Lose It!”*, to be available through multiple platforms, such as Android, IOS, Mac OS, Windows, etc. It was created originally for use with Android systems only. For this assignment, I was given the assumption that *The Gaming Room* will have 200 high-definition image files, each of which is approximately eight megabytes in size, that need to be considered during development in order to create the most optimized parameters for the game’s memory management.

When considering how to optimize an application’s performance, understanding and developing a plan that involves ways to best implement memory management is one way to do that. There are several considerations that need to be made when choosing the best approach for storage management, a key part in the process of memory management. One consideration in this scenario, is the amount of memory each of the 200 image files will take up, which totals currently at 1600 megabytes/1.6 gigabytes of used storage space.

To ensure the game runs as effectively and efficiently on all the desired operating systems, reducing image file size through compression or resolution reduction (slight), without impacting image quality, could have a significant impact, that offers improvements to the games performance and speed. Another approach is ensuring that I, as the developer, choose the most efficient data structure, based on each section of the games program requirements and needs. To do this, I would need to consider each code block, and whether it would benefit most from an array, linked list, hash table, and so on. Choosing the correct structure will help minimize the amount of memory that is needed to not only store the images, but also to process and render the images to the screen. I could also use memory management techniques, such as memory pooling or garbage collection, both of which operate in a similar manner, with slight differences in the way it redistributes the memory. Using memory pooling, allows reuse of already allocated memory, that is no longer being accessed or used regularly. Whereas garbage collection allows the system to automatically reclaim memory that is no longer needed, making the process of adding additional programming that handles the allocation and release of memory unnecessary, as it automates the process. Usage of techniques similar to these also helps prevent or eliminate common concerns and problems, such as memory leaks. Lastly, rendering images at a fixed rapid rate, using efficient rendering techniques, is an important consideration and needed step to optimize memory management and processing. To do this, I would consider using cached images or even to preload them, which will minimize the memory needed for rendering the images.

The next consideration is that of storage management. While I must admit, I previously considered memory and storage to be the exact same thing, versus having considered them to be a subset of one another. Given this new understanding, an important consideration is estimation of storage, considering we are already given 1.6 gigabytes of storage that we know for certain will be taken and already set aside for the image files. To optimize storage space needed/required for this multi-platform application, using efficient storage methods like compression to reduce the size of the image files will ultimately reduce the needed amount of storage that is required for allocating the images to memory. Using cloud storage is likely the most cost efficient, yet adjustable solution that is currently available for this program. It would make the process of managing storage scalable, secure, and easily manageable. Lastly, making certain to archive/move outdated, unused and/or least used data files will maximize efficiency of storage needed and memory allocation. A needed addition to this, however, is ensuring that archives are sent/moved to a secure and long-term data storage environment, that can still be accessed on an as needed basis.

In conclusion, how is memory different from storage? Memory management focuses mostly on the optimization and processing of the stored information. Whereas storage management is focused more-so on managing the location and relocation of data. While similar, but not identical, they are both essential in ensuring a program is operating optimally and effectively. Therefore, both are critical components that allow the game to run rapidly and effectively.