**Jude Mensah Nana Baffoe**

**SNHU**

**August 07, 2025**

**CS 230 Module Six: Memory and Storage Management for *Draw It or Lose It***

**Overview**  
The Gaming Room’s *Draw It or Lose It* is a multi-user, multi-instance game played in timed rounds where high-definition images are shown as clues. Right now, it runs on Android, but the goal is to make it work across different platforms. Because the game will utilize 200 HD images (approximately 8 MB each), we require a strategic plan for both memory (the device's current usage) and storage (the data retained over time).

Memory Management  
Memory is all about short-term use — what’s loaded and actively being worked on while the game is running. For *Draw It or Lose It*, the challenge is to pull and show images quickly, animate the drawing process smoothly, and keep things running fast even when multiple games happen at once.

**Key things to consider:**

* Each image is big, so loading too many at once could slow everything down or even crash the app.
* The game only has seconds to render and display images, so speed matters.
* Different devices and platforms have different memory limits and ways of managing active data.

**How to handle it:**

* Load images only when needed (lazy loading) and clear them from memory as soon as the round ends.
* Use caching to keep recently used images ready without holding too much in memory.
* Resize or compress images in memory for display when full resolution isn’t necessary.
* Be careful with background tasks and free up memory properly to avoid leaks.
* Preprocess images in the background so the player never sees a lag.
* Test memory use on all target platforms and fix issues before release.

**Storage Management**  
Storage is the long-term home for the game’s files, where the 200 images, game state, settings, and other assets live on the device.

Key things to consider:

* 200 images at 8MB each is about 1.6GB, so installing everything at once may eat up too much space.
* App size and updates matter. Large downloads can turn users away or cause mobile storage issues.
* Assets need to be organized so the game can pull the right image instantly.

How to handle it:

* Consider downloading image packs on demand instead of bundling everything in the initial install.
* Compress images (e.g., JPEG, WebP) to reduce storage size.
* Organize assets so they can be accessed quickly without scanning through all files.
* Allow the game to clean up old or rarely used files automatically.
* Follow platform-specific rules for where and how to store files on disk.

**Memory vs. Storage**  
Memory and storage are connected, but they serve different parts of the game’s life cycle:

* Memory is temporary. It’s the workspace the game uses while it’s running for images being drawn, for animations on screen, and for live game logic.
* Storage is permanent. It’s where all the image files, game saves, and other data sit between play sessions.

An example: when a new round starts, the game grabs an image from storage and loads it into memory to display it. Once the round ends, the image leaves memory but stays in storage.