Adding a new subtype of MemoryObject:

1. Make a MonoBehaviour extend MemoryMonoBehaviour
   1. Give it a getMemoryObject() method
2. Define a class below it in the same file
   1. Give the new class the same name but with “Memory” at the end
   2. Make the new class extend MemoryObject
   3. Override its loadState and saveState methods
   4. Give it 2 constructors
      1. One with no parameters
      2. One that takes a GameObject
         1. Call saveState() in this constructor, with the correct MemoryMonoBehaviour you need
3. Go to Assets->Easy Save 2->Manage Types
   1. Type in the name of the new class that extends MemoryObject and select it from the list
   2. Check all 3:
      1. “found”
      2. ”objectName”
      3. “sceneName”
   3. Click “Add Type”
4. In the Projects tab, go to Easy Save 2->Types
   1. Find the file that begins with “ES2UserType\_” and ends with your class that extends MemoryObject
   2. Open it.
5. Edit the “ES2UserType\_[MemoryObject subtype]” class
   1. In the Write(..) method, insert the following line of code before any other writer.Write(.) calls:
      1. writer.Write("[MemoryObject subtype]");
6. Edit the “ES2UserType\_MemoryObject” class
   1. In the Write(..) method, insert the following line of code in the if structure before the else clause, where [MOST] is the name of the MemoryObject SubType:
      1. else if (obj.GetType() == typeof([MOST]))
      2. {
      3. [MOST] cpcm = ([MOST])obj;
      4. writer.Write("[MOST]");
      5. [any write calls unique to this subtype go here]
      6. }
   2. In the Read(.) method, do the same:
      1. else if (objType == "[MOST]")
      2. {
      3. data = new [MOST] ();
      4. [any read calls unique to this subtype go here, and you may have to cast]
      5. }
7. Also, somewhere in the MemoryMonoBehaviour’s code, where it changes state, it should inform the GameManager by calling:
   1. GameManager.saveMemory(this);