**gameObject**

* Users create an instance of this in their game for each of the primary 2D game objects, such as players, enemies, power-ups, projectiles, effects, etc.

**objectContainer**

* This stores all instances of the *gameObject* instance, for example, all of the enemies on a given level/board.
* So if there are 3 types of enemies in your game enemy A, enemy B, and enemy C, this object holds multiple of each enemy type. The enemy object data could look like A, A, B, A, C, B, B, C, A, C, C, A. This would be 12 total enemies in that level, 5 enemy A’s, 3 enemy B’s, and 4 enemy C’s.
* This also stores the separate attributes of each object, such as the x and y position, the current frame of animation, timers, sound effects, sound effects channel, imgContainer reference, etc.
* Performance is optimal when each *objectContainer* instance references only 1 *imgContainer*, however multiple *imgContainers* (images) can be associated with one *objectContainer*. This is done automatically by setting the ImgLookupFlag to true, however there is a performance hit.

**imgContainer**

* This stores multiple image files (sprite sheets) to be used/referenced by instances of *objectsContainer* for drawing purposes.
* Each *objectsContainer* instance stores a reference (set by the user) to the *imgContainer* instance. Most *gameObjects* will only have one image file (sprite sheet), however some users may, for example, split each enemy into separate image files (sprite sheets).
* This structure is also useful if one large image file is too large for the video driver on the machine and must be spilt into multiple files (today’s standard size is 8194x8194 pixels per image).
* Also stores hurt and death status.

**clips**

* Stores a grid of rows/cols for each *imgContainer* instance where each row represents the animation status (the rows in the image file) and each animation status’ frames (the columns in the image file).
* The underlying structure is a vector of SDL\_Rect’s.

**statusAttributes**

* Stores the animation status (each row in the image file) attributes for each *imgContainer*.
* These are attributes that are identical for each objectsContainer instance, such as the speed of the animation, the start and end frames numbers, attack frames, collision detection rectangles and points, etc.