# PowerGrid caching structure

To speed up the AI and to provide access to (relatively) static information, PowerGrid will have to cache some information from the Runescape world to allow the AI module to automate certain tasks that otherwise could not be performed. Most notably the path finding, which relies on a worldwide collision map to compute its path, will need the caches for accurate path finding. This is because Runescape only loads a small part of the world at a time, preventing worldwide path finding without a suitable caching structure to remember the locations of collisions and shortcuts.

## Objects suited to be cached

Scene objects and the ground information are by far the most cacheable. No-interact objects as well as collision info also belong to this category.

Widgets in Runescape 2 are only partially cacheable due to continuously changing content. It is more efficient to map only the id’s of the widgets to their object in the Java environment and link the id’s to a semantic meaning by using either the #define pre-processor instruction or by using an enumeration that maps the id’s to semantic names.

It is also possible to cache the contents of the player’s bank, since this does not change unless the player has the bank widget open. This allows for the AI module to make more accurate decisions based on the items available to it. It also prevents it from having to go to an actual bank just to see if a certain item is in the player’s possession.

To sum up, the elements from the Runescape world to be cached are:

* Information relating to the world itself
  + Collision info
  + Objects featuring a certain type of interaction
  + The interaction options belonging to the objects can also be stored along the object if it is known. However, since the same type of objects usually have the same actions, it is more useful to store a reference to a definition type object and load the information from there.
* Widgets and in some cases their content
  + Using #defines or an enumeration to identify the widget
  + Also cache semantic content of some widgets like skills, equipment and bank content

Note that in some cases the exact information that is cached depends on the content of the object itself, most notably the widgets, which are cached depending on whether they could be interesting for the AI module, and the objects. Wall objects and other objects that only serve as a collision are not cached in the objects list, but only as a collision on the collision map.

## Caching method for different object types

The ground information as well as objects can be cached by running a dedicated thread that loads and maps the Runescape environment as the player moves through it. Widgets can be cached when they are loaded and visible. Non-visible widgets are usually not important and should not be cached at all. Also widgets that do not have content at all should not be cached to prevent caching redundant widgets.

It is decided that PowerGrid will map the environment by splitting the world in 64x64-sized clusters. Each cluster can contain multiple levels (planes) on which collision information can be stored. Such a cluster also contains a list of the objects contained in that cluster. This allows for a quick method of looking up objects near the current player by first searching the cluster the player is in, then the clusters surrounding that, and so on.

Widgets will be cached by id in a hash structure. This allows for constant time store and retrieve functions as well as a relatively compact way of storing the widget references by using linear probing to optimize the usage of the available memory space. The widget cache itself should have a maximum size, and when a widget gets cached that causes the cache to grow past the maximum size, that widget should replace the widget being accessed last. *However, this poses the problem that the retrieve and store functions are no longer performed in constant time when the cache is full.*

## Using the cache structure

The caches will be used mostly in the AI module, but this does not mean similar caches are not used in other modules. However, the focus of this article lies with the caches used by the AI.

The pathfinder can use the caches to compute paths across the entire Runescape world. This allows for other tasks to be executed in a more dynamic fashion, and also speeds up any required operations for the path finding itself.

Also the collection of various data from Runescape widgets by the AI module speeds up by using caches to store information. This is because retrieving the same information through JNI over and over again is pointless and should be avoided. The AI module should be able to indicate a refresh of the content is absolutely required in cases where getting the latest information is more important than speed.