**PackCode Structure**

Inbuilt AR image recognising database

|  |  |
| --- | --- |
| Index | Images |
| 1 | β |
| 2 | ☺ |

HashMap/T-Map that compares (this is hardcoded) the index of the above database as a key that provides a 3-digit number followed by a hyphen and a further number of varying numbers of digits, e.g. 100-3

|  |  |
| --- | --- |
| Integer | String |
| 1 | 100-1 |
| 2 | 100-2 |

The associated String is deconstructed into an integer made of the first three digits which becomes a “Pack Number” all digits after the third (the hyphen) are also stored as an integer to become an “asset number”

The Pack Number is put into another HashMap or array to find the associated pack array (an array containing all the assets in the specified pack)

|  |  |
| --- | --- |
| Pack Number | Pack Array |
| 100 | Basic Pack |
| 200 | Monster Pack |

The associated pack array is then searched using the Asset Number (which refers to the index within the array) to find the specific asset the initial image was associated with

|  |  |
| --- | --- |
| Index | Asset |
| 1 | ShrineRoom |
| 2 | BarracksRoom |

The found room asset is then spawned and anchored to the scanned location