|  |  |
| --- | --- |
| Beacon has a light color (x,y) position and name |  |
|  |  |
| Int Beacon\_count() | Int place\_count() |
| Void clear\_beacon() | Void clear\_all() |
| all\_beacon  std::vector<BeaconID>  all\_beacon() | all\_places  std::vector<PlaceID>  all\_places() |
| All\_beacon ID Name (x.y) (r,g,b)  Bool add\_beacon(BeaconID id, string const& name, Coord xy, Color color) | Add\_place ( PlaceID id, Name const& name, Coord xy, PlaceType type |
| Std::string get\_name(BeaconID id) | Place\_name\_type ID  Std::pair <Name, PlaceType>  Get\_place\_name\_type ( PlaceID id) |
| Coord get\_coordinate(BeaconID id) | Coord get\_place\_coordinate(PlaceID id) |
|  |  |
| std::vector<BeaconID> beacon\_alphabetically() | Std::vector<PlaceID>places\_alphabetically() |
| Std::vector<BeaconID>beacon\_brightness\_increasi() | Std::vector<PlaceID>places\_coord\_order() |
| Std::vector<BeaconID>find\_beacon(string const& name) | Std::vector<PlaceID>find\_place\_name (Name const& type) |
|  | Std::vector<PlaceID>find\_place\_type(PlaceType type) |
| Bool change\_beacon\_name ( BeaconID id, string newname | Bool change\_place\_name(PlaceID place, Name const& new\_name) |
| Bool change\_beacon\_color(BeaconID id, Color newcolor) | Bool change\_place\_coord(PlaceID id, Coord newcoord) |
|  |  |
| Bool add\_lightbeam(BcID sourceid, BcID targetid) | Bool add\_area(AreaID id, Name name, std:vector<Coord>cords, |
| Std::vector<BcID>Path\_outbeam(BcID id) | Vct<AreaID>Sub\_area\_to\_area(AreaID id,AreaID parentid) |
| Path\_inbeam\_longest(BeaconID id) | Places\_closest\_to(Coord xy, PlaceType type) |
| Bool remove\_beacon(BeaconID id) | Remove\_place(PlaceID id) |
|  |  |

Beacon = placeCsst

Optional func : remove\_beacon(), path\_inbeam\_longest(), total\_color(). Combining for received light beam done simply by calculating the average of each color component (r,g,b). Brightness 3\*r + 6\*g +b

Sorting beacon in alphabetic order.Sort do with comparison < in C++ string class.