**DataContainerSingleton.h**

(Alla structs finns längst ner i dokumentet om ni är för lata att kolla i koden (: )

Init:

För att skapa/hämta singletonen anropar man den statiska funtionen **getInstance();**

DataContainerSingleton& data = DataContainerSingleton::getInstance();

Datan laddas in i början av OnInit() i game.cpp så ni behöver inte anropa denna metod för att hämta datan, det ska redan finnas i singletonen.

Hämta data:

För att hämta data används en av de 4 get funktionerna. Datan är sparad i Oryol::map container som fungerar som en vanlig map.

Oryol::Map<Oryol::StringAtom, UnitInfo> unitMap = data.getUnitMap();

Oryol::Map<Oryol::StringAtom, BuildingInfo> buildingMap = data.getBuildingMap();

Oryol::Map<Oryol::StringAtom, MaterialInfo> materialMap = data.getMaterialMap();

Oryol::Map<Oryol::StringAtom, TileInfo> tileInfoMap = data.getTileInfoMap();

I dessa maps är olika structs lagrade. Dessa structs är baserade på informationen som finns i json filen. Det finns en unit, building, material och en tileinfo struct. För att t.ex. hämta hälsan för worker skriver man såhär.

int health = unitMap["worker"].health;

där nyckeln är namnet på entiteten du vill hämta info om.

För att ge ett till exempel så hämtar man hur mycket träd som går åt för att bygga en smed såhär.

int woodReq = buildingMap["smithy"].materialsRequired.wood;

struct MaterialsRequired {

int wood;

int charcoal;

int ironOre;

int ironIngot;

int sword;

};

struct UnitInfo {

Oryol::StringAtom name;

int health;

int movementSpeed;

int upgradeTime;

Oryol::StringAtom upgradeLocation;

MaterialsRequired materialsRequired;

};

struct BuildingInfo {

Oryol::StringAtom name;

MaterialsRequired materialsRequired;

int buildtime;

int size;

};

struct MaterialInfo {

Oryol::StringAtom name;

int timeToCreate;

Oryol::StringAtom refineLocation;

MaterialsRequired materialsRequired;

};

struct TileInfo {

Oryol::StringAtom name;

int materialCount;

float movementSpeed;

bool walkable;

bool buildable;

};