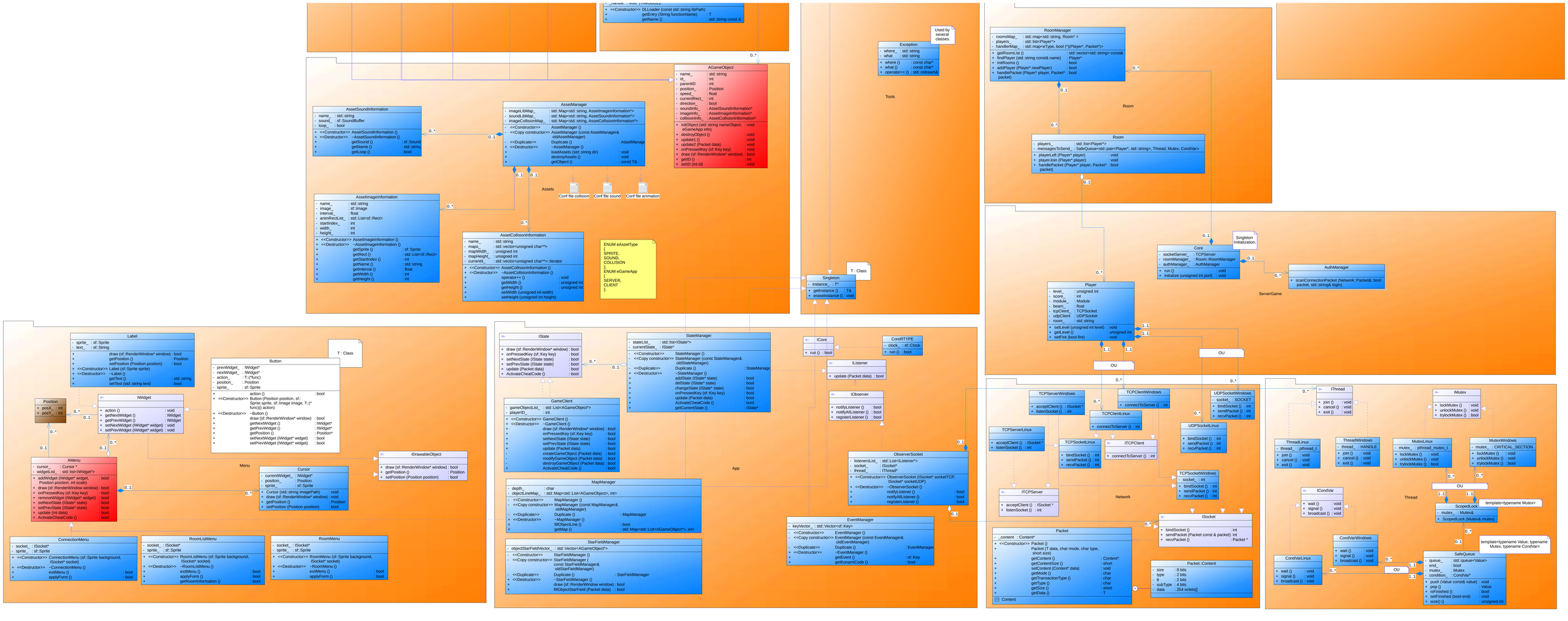
Fighter.lib
Monster1Lib
Monster2Lib
Monster3Lib
Bullet1Lib
Bullet2Lib
Bullet3Lib
Wait1Lib
Wait2Lib
Wait3Lib
Weapon1Lib
Weapon2Lib
Weapon3Lib
Weapon4Lib
Weapon5Lib
Weapon6Lib
Weapon7Lib
Object
Loader
DLLoader
- name : String
- handle : void *
loadersymbol(pname T*)
0..*
1..1
0..1
LoaderManager
- dcSpecMap_ : std::map<std::string, DLLoader<AlarmObject>*>
+ LoadSelect(String folder) : void
+ getAlarmObject(String filename) : AlarmObjectPtr
Timer
- time_ : float
Database
- tableVector_ : std::vector<Table*>
+ fillFromFile (std::string const& file) : bool
+ operator[](std::string const& name) : Table*
0..1
RTDB
Table
- rows_ : Row
- name_ : std::string
+ addRow () : bool
+ getRows () : Rows const&
+ getColumns () : std::string const&
0..*
0..1
0..*
DatabaseParser
- currentTable_ : std::string
- currentRow_ : std::vector<std::string>*
+ readFile(std::istream& &in) : std::istream&
+ newRow () : bool
+ newTable () : bool
+ getTableName () : std::string const&
+ getRow () : std::vector<std::string> const&
ConsumerParser
- buffer_ : std::string
- tags_ : Map
+ parseLine () : bool
+ readFile () : bool
+ readLine() : bool
+ readBOF () : bool
+ readEOF () : bool
+ readBytes () : bool
+ readUntil () : bool
+ readUntilEOF () : bool
+ readline () : bool
+ readMessage () : bool
+ readMessageBegin () : bool
+ readMessageEnd () : bool
+ readStructBegin () : bool
+ readStructEnd () : bool
+ readFieldBegin () : bool
+ readFieldEnd () : bool
+ readMapBegin () : bool
+ readMapEnd () : bool
+ readListBegin () : bool
+ readListEnd () : bool
+ readSetBegin () : bool
+ readSetEnd () : bool
+ skip(int) : void
+ reset() : void
Parser


DLLoader
+ <<Constructor>> DLLoader (const std::string libPath)
+ getEntry (String functionName) : T
+ getName () : std::string const &
Exception
- where_ : std::string
- what_ : std::string
+ where () : const char*
+ what () : const char*
+ operator<< () : std::ostream&
Used by several classes.
Tools
AGameObject
- name_ : std::string
- id_ : int
- parentID : int
- position_ : Position
- speed_ : float
- currentRect_ : int
- direction_ : bool
- soundInfo_ : AssetSoundInformation*
- imageInfo_ : AssetImageInformation*
- collisionInfo_ : AssetCollisionInformation*
+ initObject (std::string nameObject, eGameApp info) : void
+ destroyObject () : void
+ update1 () : void
+ update2 (Packet data) : void
+ onPressedKey (sf::Key key) : void
+ draw (sf::RenderWindow* window) : bool
+ getID () : int
+ setID (int id) : void
AssetSoundInformation
- name_ : std::string
- sound_ : sf::SoundBuffer
- loop_ : bool
+ <<Constructor>> AssetSoundInformation ()
+ <<Destructor>> ~AssetSoundInformation ()
+ getSound () : sf::Sound
+ getName () : std::string
+ getLoop () : bool
AssetManager
- imageLibMap_ : std::Map<std::string, AssetImageInformation*>
- soundLibMap_ : std::Map<std::string, AssetSoundInformation*>
- imageCollisionMap_ : std::Map<std::string, AssetCollisionInformation*>
- <<Constructor>> AssetManager ()
+ <<Copy constructor>> AssetManager (const AssetManager& oldAssetManager)
- <<Duplicate>> Duplicate () : AssetManager
+ <<Destructor>> ~AssetManager ()
+ loadAssets (std::string dir) : void
+ destroyAssets () : void
+ getObject () : const T&
Assets
Conf file collision
Conf file sound
Conf file animation
AssetImageInformation
- name_ : std::string
- image_ : sf::Image
- interval_ : float
- animRectList_ : std::List<sf::Rect>
- startIndex_ : int
- width_ : int
- height_ : int
+ <<Constructor>> AssetImageInformation ()
+ <<Destructor>> ~AssetImageInformation ()
+ getSprite () : sf::Sprite
+ getRect () : std::List<sf::Rect>
+ getStartIndex () : int
+ getName () : std::string
+ getInterval () : float
+ getWidth () : int
+ getHeight () : int
AssetCollisionInformation
- name_ : std::string
- maps_ : std::vector<unsigned char**>
- mapWidth_ : unsigned int
- mapHeight_ : unsigned int
- current_ : std::vector<unsigned char**>::iterator
+ <<Constructor>> AssetCollisionInformation ()
+ <<Destructor>> ~AssetCollisionInformation ()
+ operator++ () : void
+ getWidth () : unsigned int
+ getHeight () : unsigned int
+ setWidth (unsigned int width)
+ setHeight (unsigned int height)
ENUM eAssetType
{
SPRITE,
SOUND,
COLLISION
}
ENUM eGameApp
{
SERVER,
CLIENT
}
T : Class
Singleton
- instance_ : T*
+ getInstance () : T&
+ eraseInstance () : void
RoomManager
- roomsMap_ : std::map<std::string, Room* >
- players_ : std::list<Player*>
- handlerMap_ : std::map<eType, bool (*)(Player*, Packet*)>
+ getRoomList () : std::vector<std::string> const&
+ findPlayer (std::string const& name) : Player*
+ initRooms () : bool
+ addPlayer (Player* newPlayer) : bool
+ handlePacket (Player* player, Packet* packet) : bool
Room
Room
- players_ : std::list<Player*>
- messagesToSend_ : SafeQueue<std::pair<Player*, std::string>, Thread, Mutex, CondVar>
+ playerLeft (Player* player) : void
+ playerJoin (Player* player) : void
+ handlePacket (Player* player, Packet* packet) : bool
Core
- socketServer_ : TCPServer
- roomManager_ : Room::RoomManager
- authManager_ : AuthManager
+ run () : void
+ initialize (unsigned int port) : void
Singleton initialization
AuthManager
+ scanConnectionPacket (Network::Packet&, bool packet, std::string& login)
ServerGame
Player
- level_ : unsigned int
- score_ : int
- module_ : Module
- beam_ : float
- tcpClient_ : TCPSocket
- udpClient_ : UDPSocket
- room_ : std::string
+ setLevel (unsigned int level) : void
+ getLevel () : unsigned int
+ setFire (bool fire) : void
OU
OU
Label
- sprite_ : sf::Sprite
- text_ : sf::String
+ draw (sf::RenderWindow* window) : bool
+ getPosition () : Position
+ setPosition (Position position) : bool
+ <<Constructor>> Label (sf::Sprite sprite)
+ <<Destructor>> ~Label ()
+ getText () : std::string
+ setText (std::string text) : bool
Position
+ posX_ : int
+ posY_ : int
IWidget
+ action () : void
+ getNextWidget () : IWidget
+ getPrevWidget () : IWidget
+ setNextWidget (IWidget* widget) : void
+ setPrevWidget (IWidget* widget) : void
Button
- prevWidget_ : IWidget*
- nextWidget_ : IWidget*
- action_ : T::(*func)
- position_ : Position
- sprite_ : sf::Sprite
+ action () : bool
+ <<Constructor>> Button (Position position, sf::Sprite sprite, sf::Image image, T::(* func)() action)
+ <<Destructor>> ~Button ()
+ draw (sf::RenderWindow* window) : bool
+ getNextWidget () : IWidget*
+ getPrevWidget () : IWidget*
+ getPosition () : Position*
+ setNextWidget (IWidget* widget) : bool
+ setPrevWidget (IWidget* widget) : bool
T : Class
Menu
AMenu
- cursor_ : Cursor *
- widgetList_ : std::list<IWidget*>
+ addWidget (IWidget* widget, Position position, int scale) : bool
+ draw (sf::RenderWindow window) : bool
+ onPressedKey (sf::Key key) : bool
+ removeWidget (IWidget* widget) : bool
+ setNextState (IState* state) : bool
+ setPrevState (IState* state) : bool
+ update (int data) : bool
+ ActivateCheatCode () : bool
Cursor
- currentWidget_ : IWidget*
- position_ : Position
- sprite_ : sf::Sprite
+ Cursor (std::string imagePath) : void
+ draw (sf::RenderWindow* window) : void
+ getPosition () : Position
+ setPosition (Position position) : bool
IDrawableObject
+ draw (sf::RenderWindow* window) : bool
+ getPosition () : Position
+ setPosition (Position position) : bool
ConnectionMenu
- socket_ : ISocket*
- sprite_ : sf::Sprite
+ <<Constructor>> ConnectionMenu (sf::Sprite background, ISocket* socket)
+ <<Destructor>> ~ConnectionMenu ()
+ exitMenu () : bool
+ applyForm () : bool
RoomListMenu
- socket_ : ISocket*
- sprite_ : sf::Sprite
+ <<Constructor>> RoomListMenu (sf::Sprite background, ISocket* socket)
+ <<Destructor>> ~RoomListMenu ()
+ exitMenu () : bool
+ applyForm () : bool
+ getRoomInformation () : bool
RoomMenu
- socket_ : ISocket*
- sprite_ : sf::Sprite
+ <<Constructor>> RoomMenu (sf::Sprite background, ISocket* socket)
+ <<Destructor>> ~RoomMenu ()
+ exitMenu () : bool
+ applyForm () : bool
IState
+ draw (sf::RenderWindow* window) : bool
+ onPressedKey (sf::Key key) : bool
+ setNextState (IState state) : bool
+ setPrevState (IState state) : bool
+ update (Packet data) : bool
+ ActivateCheatCode () : bool
StateManager
- stateList_ : std::list<IState*>
- currentState_ : IState*
- <<Constructor>> StateManager ()
- <<Copy constructor>> StateManager (const StateManager& oldStateManager)
- <<Duplicate>> Duplicate () : StateManager
+ <<Destructor>> ~StateManager ()
+ addState (IState* state) : bool
+ delState (IState* state) : bool
+ changeState (IState* state) : bool
+ onPressedKey (sf::Key key) : bool
+ update (Packet data) : bool
+ ActivateCheatCode () : bool
+ getCurrentState () : IState*
App
GameClient
- gameObjectList_ : std::List<AGameObject*>
- playerID_ : int
+ <<Constructor>> GameClient ()
+ <<Destructor>> ~GameClient ()
+ draw (sf::RenderWindow* window) : bool
+ onPressedKey (sf::Key key) : bool
+ setNextState (IState state) : bool
+ setPrevState (IState state) : bool
+ update (Packet data) : bool
- createGameObject (Packet data) : bool
- modifyGameObject (Packet data) : bool
- destroyGameObject (Packet data) : bool
+ ActivateCheatCode () : bool
MapManager
- depth_ : char
- objectLineMap_ : std::Map<std::List<AGameObject>, int>
- <<Constructor>> MapManager ()
- <<Copy constructor>> MapManager (const MapManager& oldMapManager)
- <<Duplicate>> Duplicate () : MapManager
+ <<Destructor>> ~MapManager ()
- fillObjectLine () : bool
+ getMap () : std::Map<std::List<AGameObject*>, int>
StarFieldManager
- objectStarFieldVector_ : std::Vector<AGameObject*>
- <<Constructor>> StarFieldManager ()
- <<Copy constructor>> StarFieldManager (const StarFieldManager& oldStarFieldManager)
- <<Duplicate>> Duplicate () : StarFieldManager
+ <<Destructor>> ~StarFieldManager ()
+ draw (sf::RenderWindow window) : bool
- fillObjectStarField (Packet data) : bool
ICore
+ run () : bool
CoreRTYPE
- clock_ : sf::Clock
+ run () : bool
IListener
+ update (Packet data) : bool
IObserver
+ notifyListener () : bool
+ notifyAllListener () : bool
+ registerListener () : bool
ObserverSocket
- listenersList_ : std::List<IListener*>
- socket_ : ISocket*
- thread_ : IThread*
+ <<Constructor>> ObserverSocket (ISocket* socketTCP, ISocket* socketUDP)
+ <<Destructor>> ~ObserverSocket ()
+ notifyListener () : bool
+ notifyAllListener () : bool
+ registerListener () : bool
EventManager
- keyVector_ : std::Vector<sf::Key>
- <<Constructor>> EventManager ()
- <<Copy constructor>> EventManager (const EventManager& oldEventManager)
- <<Duplicate>> Duplicate () : EventManager
+ <<Destructor>> ~EventManager ()
+ getEvent () : sf::Key
+ getKonamiCode () : bool
Network
TCPServerWindows
+ acceptClient () : ISocket *
+ listenSocket () : int
TCPClientWindows
+ connectToServer () : int
UDPSocketWindows
- socket_ : SOCKET
+ bindSocket () : int
+ sendPacket () : int
+ recvPacket () : int
TCPClientLinux
+ connectToServer () : int
TCPServerLinux
+ acceptClient () : ISocket *
+ listenSocket () : int
TCPSocketLinux
+ bindSocket () : int
+ sendPacket () : int
+ recvPacket () : int
ITCPClient
+ connectToServer () : int
UDPSocketLinux
+ bindSocket () : int
+ sendPacket () : int
+ recvPacket () : int
TCPSocketWindows
- socket_ : int
+ bindSocket () : int
+ sendPacket () : int
+ recvPacket () : int
ITCPServer
+ acceptClient () : ISocket *
+ listenSocket () : int
ISocket
+ bindSocket () : int
+ sendPacket (Packet const & packet) : int
+ recvPacket () : Packet *
Packet
- _content : Content*
+ <<Constructor>> Packet ()
+ Packet (T data, char mode, char type, short size)
+ getContent () : Content*
+ getContentSize () : short
+ setContent (Content* data) : void
+ getMode () : char
+ getTransactionType () : char
+ getType () : char
+ getSize () : short
+ getData () : T
Content
Packet::Content
- size : 8 bits
- type : 2 bits
- ti : 2 bits
- subType : 4 bits
- data : 254 octet[s]
Thread
IThread
+ join () : void
+ cancel () : void
+ exit () : void
IMutex
+ lockMutex () : void
+ unlockMutex () : void
+ trylockMutex () : bool
ThreadLinux
- thread_ : pthread_t
+ join () : void
+ cancel () : void
+ exit () : void
ThreadWindows
- thread_ : HANDLE
+ join () : void
+ cancel () : void
+ exit () : void
MutexLinux
- mutex_ : pthread_mutex_t
+ lockMutex () : void
+ unlockMutex () : void
+ trylockMutex () : bool
MutexWindows
- mutex_ : CRITICAL_SECTION
+ lockMutex () : void
+ unlockMutex () : void
+ trylockMutex () : bool
ICondVar
+ wait () : void
+ signal () : void
+ broadcast () : void
OU
ScopedLock
- mutex_ : Mutex&
+ ScopedLock (Mutex& mutex)
template<typename Mutex>
CondVarWindows
+ wait () : void
+ signal () : void
+ broadcast () : void
CondVarLinux
+ wait () : void
+ signal () : void
+ broadcast () : void
OU
SafeQueue
- queue_ : std::queue<Value>
- end_ : bool
- mutex_ : Mutex
- condition_ : CondVar*
+ push (Value const& value) : void
+ pop () : Value
+ isFinished () : bool
+ setFinished (bool end) : void
+ size() () : unsigned int
template<typename Value, typename Mutex, typename CondVar>