UML Class Diagram

GraphicEngine
- +display(const Entities &) : void
- +isWindowOpen() : bool
- +clearWindow() : void
- +getUserEvent() : const std::vector<std::shared_ptr<IUserEvent>> &
- +getIUserPos() : Position

SFMLGraphic
- + SFMLGraphic()
- +display(const Entities &) : void
- +displayEntity(*) : void
- +displayFrame(*) : void
- +displaySquare(*) : void
- +displayCircle(*) : void
- +displayText(*) : void
- +displayASprite(*) : void
- +displaySound(*) : void
- +isWindowOpen() : bool
- +clearWindow() : void
- +getIUserPos() : Position
- +getUserEvent() : const std::vector<std::shared_ptr<IUserEvent>> &
- +createIUdd(Event::KeyEvent) : KeyboardEvent::Add
- +getColor(const RGBAColor &) : sf::Color
- _window : sf::RenderWindow
- _vsl : sf::RectangleShape
- _scale : sf::CircleShape
- _text : sf::Text
- _sprites : std::map<std::string, std::pair<sf::Texture, sf::Sprite>>
- _font : std::map<std::string, sf::Font>
- _sounds : std::map<std::string, std::pair<sf::SoundBuffer, sf::Sound>>

GameEngine
- GameEngine(IGraphics *, InterfaceManager *)
- getData(Time) : double
- read() : int
- _graphicEngine : std::unique_ptr<IGraphics>
- _interfaceManager : std::unique_ptr<InterfaceManager>
- _lastLoop : std::chrono::time_point<std::chrono::system_clock>

Client
- socket

Server
- games : vector<game>
- thread : Thread

Game
- player : vector<Player>
- thread : Thread
- socket

Player
- player_name : string
- thread : Thread

UDP
- +UDP
- +UDP
- +writestring() : void
- +readbool() : string
- socket : int

Thread
- Thread(string, int)

ClientUtils

IUserEvent
- +getType() : EventType

IUserEvent
- IUserEvent(const EventType &)
- +getEventType() : EventType
- _type : const EventType

UserEvent
- StateEvent(EventState)
- +getState() : EventState
- _state : EventState

KeyboardEvent
- KeyboardEvent(EventState, KeyboardInput, Add)
- +getKey() : KeyboardInput
- +get_Add(keys) : Add
- _key : const KeyboardInput
- _Add : const Add

MouseEvent
- MouseEvent(MouseActionType, EventState)
- +getEventType() : MouseActionType
- _buttonType : const MouseActionType

ScreenEvent
- ScreenEvent(ScreenEventType)
- +getEventType() : ScreenEventType
- _screenEvent : const ScreenEventType

TextEvent
- TextEvent(wchar_t code)
- +getCode() : wchar_t
- _code : const wchar_t

Entity
- +getType() : EntityType
- +getPosition() : const Position &
- +setPosition(const Position &) : void
- +getSize() : const Size &
- +setSize(const Size &) : void

AEntity
- AEntity(EntityType, const Position &, const Size &)
- +getType() : EntityType
- +getPosition() : const Position &
- +setPosition(const Position &) : void
- +getSize() : const Size &
- +setSize(const Size &) : void
- _type : EntityType
- _pos : Position
- _size : Size

Color
- Color(const RGBAColor &)
- +getColor() : const RGBAColor &
- +setColor(const RGBAColor &) : void
- _color : RGBAColor

Form
- Form(const Position &, const RGBAColor &, FormType)
- +getFormType() : FormType
- _type : const FormType

Square
- Square(const Position &, const Size &, const RGBAColor &)
- +getSize() : const Size &
- +setSize(const Size &) : void
- _size : Size

Circle
- Circle(const Position &, double, const RGBAColor &)
- +getRad() : double
- +setRad(double) : void
- _rad : double

Text
- Text(const std::string &, const std::string &, const Position &, const Size &, const RGBAColor &)
- +getMsgstr() : const std::string &
- +getText() : const std::string &
- +getSize() : const Size &
- +setPos(const std::string &) : void
- +setText(const std::string &) : void
- +operator=(wchar_t_) : void
- +operator=() : void
- _filepath : const std::string
- _text : std::string
- _size : Size

Sound
- Sound(const std::string &, bool)
- +getMsgstr() : const std::string &
- +operator=(double) : void
- _filepath : const std::string
- _isLooping : const bool

Sprite
- +getMsgstr() : const std::string &
- +getSize() : bool
- +isAnimated() : bool

ASprite
- ASprite(const std::string &, const Position &, bool, const Size &)
- +getMsgstr() : const std::string &
- +isLooping() : bool
- +operator=(double) : void
- _filepath : const std::string
- _isLooping : const bool
- _cumulDeltaTime : double

NoAnimatedSprite
- NoAnimatedSprite(const std::string &, const Size &, const Position &, bool, const Size &)
- +getRect() : Rect
- _imageSize : const Size
- _rect : Rect

AnimatedSprite
- AnimatedSprite(const std::string &, const Position &, const Size &, bool, const SpriteSheetRects &, double timeBetweenAnim, const Size &, const Size &)
- +getRect() : Rect
- +newAnims() : void
- +setActionIndex(const size_t &) : void
- +setAnimIndex(const size_t &) : void
- +getAnimIndex() : const size_t &
- +getCumul() : double
- +operator=(double) : void
- _rect : const SpriteSheetRects &
- _rowIndex : size_t
- _timeBetweenAnim : double
- _cumulDeltaTime : double

Directional
- +setDirection(const Direction &) : void
- +getDirection() : const Direction &
- _direction : Direction

Projectile
- +Projectile(const std::string &, const Position &, const SpriteSheetRects &, double timeBetweenAnim, const Direction &, const Size &, const size_t &, const size_t &)
- +newAnim() : void

Player
- Player(const Position &, const size_t, playerType)
- +newAnim() : void
- +setPosition(const Position &) : void
- +setDirection(const Direction &) : void
- _spriteSheet : const SpriteSheetRects

PlayerFire
- PlayerFire(const Position &, const size_t, playerType)
- +newAnim() : void
- _spriteSheet : const SpriteSheetRects

AEnemy
- AEnemy(const std::string &, const SpriteSheetRects &, double, const Size &)
- +getLife() : double
- +operator=(double) : void
- +newAnim() : void
- _life : double

IEnemy
- +getLife() : double
- +operator=(double) : void
- +fire(const Direction &) : std::shared_ptr<Projectile>

PlayerProjectile
- PlayerProjectile(const Position &, double)
- +newAnim() : void
- +getBeamPower() : double
- _spriteSheet : const SpriteSheetRects
- _beamPower : double

IPlayerUtils
- +getPlayerPath(char **, const std::string &) : std::string
- +isTimePassed(std::chrono::time_point<std::chrono::system_clock>, double) : bool
- +removeFromList(std::vector<T *>, const std::shared_ptr<T> &) : void

ServerUtils

ClientUtils
- +PointRect(Rect, Position) : bool
- +PointRect(Square *, Position) : bool

100(client) - Client connexion

110() - disconnect from server

120(game) - Create game

140(game) - join game

150() - leave game

170() - Client ready

180() - Client not ready

210(ID) - loadingBeam

220(ID) - beamShooted

230(ID, Direction) - playerDirection

101(Game list) - Login Success

102(error) - Login Failure

121(Game) - Game created

122(error) - Game couldn't be created

130(Games) - Update Game list

141(Game) - Game joined

142(error) - Game couldn't be joined

151() - Game left

152(error) - Couldn't left game

160(Clients, state) - Update player list

171() - Client is ready

172(error) - Client couldn't be set as ready

181() - Client isn't ready

182(error) - Client couldn't be set as not ready

190() - Game start - Start sending map

211(ID) - send Player beam loading

221(ID) - send Player beam shooted

231(ID, Direction) - send Player direction

240(Projectile) - send enemy shoot

250(Entities) - update entites position

- CLIENT_NAME : 16
- GAME_NAME : 16
- ERROR : 128
- MAP
- PATH : ---
- RECT :
- LIFE : 2
- SCALE :
- TBA :
- LI :
- AI :
- DIRECT :
- SIZE :
- COLOR :
- SPEED :
- POS :

Enemy;path;rect;pos;life;scale;
timeBetweenAnim;lineIndex;animIndex

Player;pos;lineIndex

Projectile;path;pos;rect;timeBetweenAnim;
direction;projectileSpeed;size;lineIndex;animIndex

Text;filepath;text;pos;size;color

Example:
 XYZ(args) - Request description

Description:
 X : Connection type (1 = TCP, 2 = UDP)
 Y : Request id
 Z : Request type (0 = ping, 1 = ping back success, 2 = ping back error)