

The Class Model is a logical model of the software system under construction. Classes generally have a direct relationship to source code or other software artifacts that can be grouped together into executable components.

The System package contains the classes and artifacts which are being built or designed as part of the current model. The Frameworks package generally contains classes and components that have been designed and built earlier and are being reused as part of the current project.

 [Read about Class Modeling](#)

 [View Further Examples](#)

System

- + Class1
- + Class2
- + Class3
- + Interface1

Classes, interfaces and logical components for the new system

Frameworks

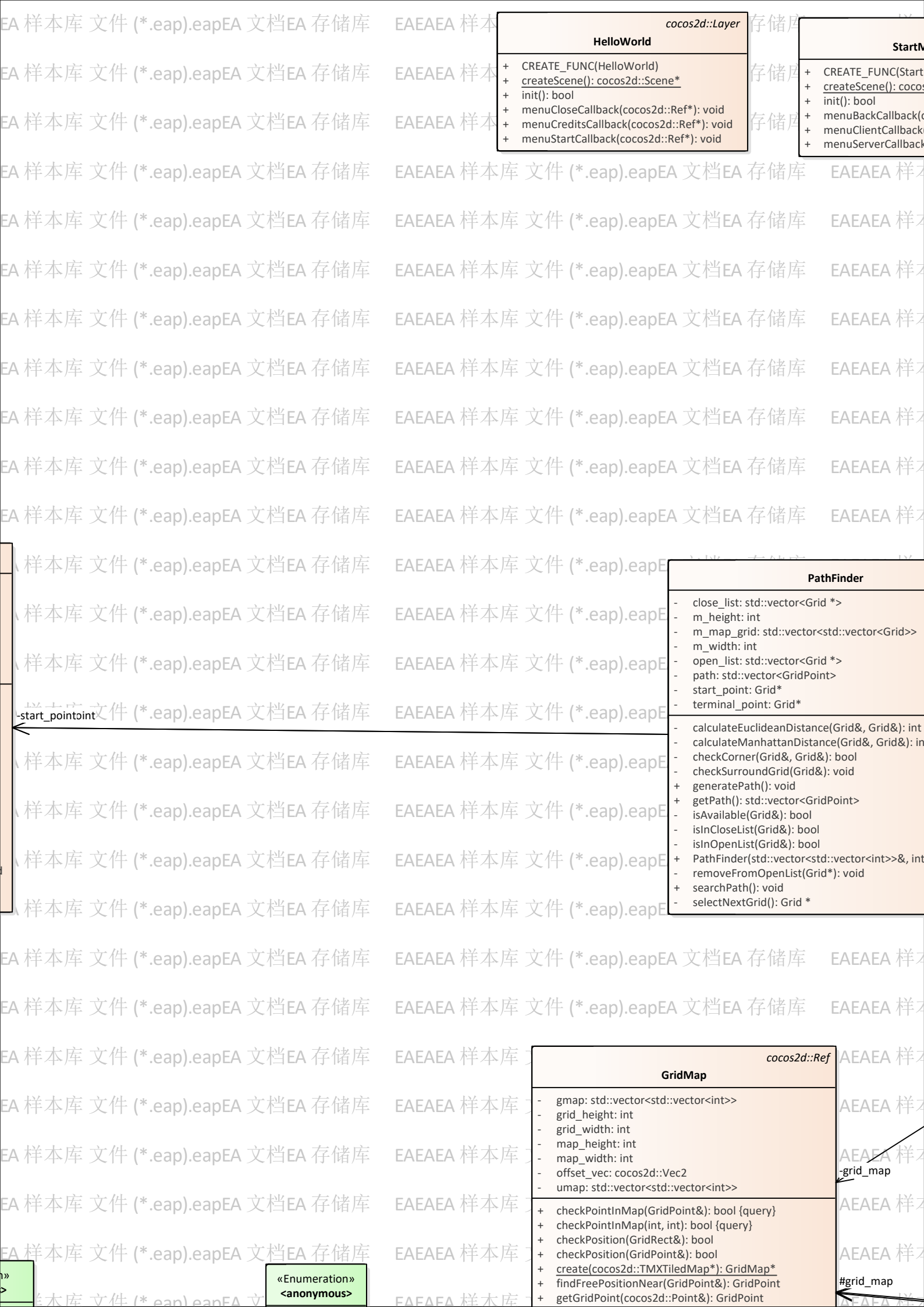
Class libraries, API's and other re-usable components

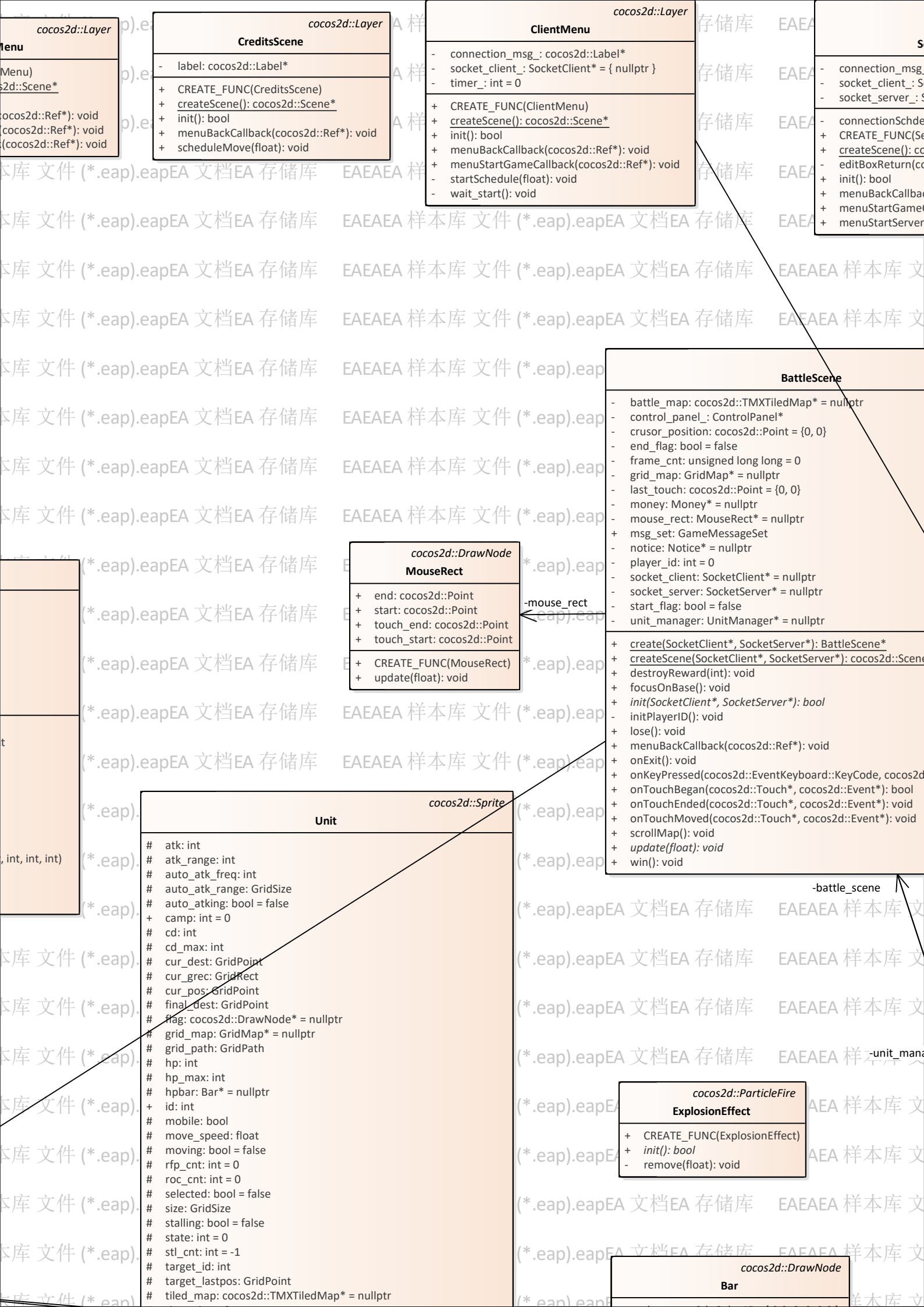
Grid

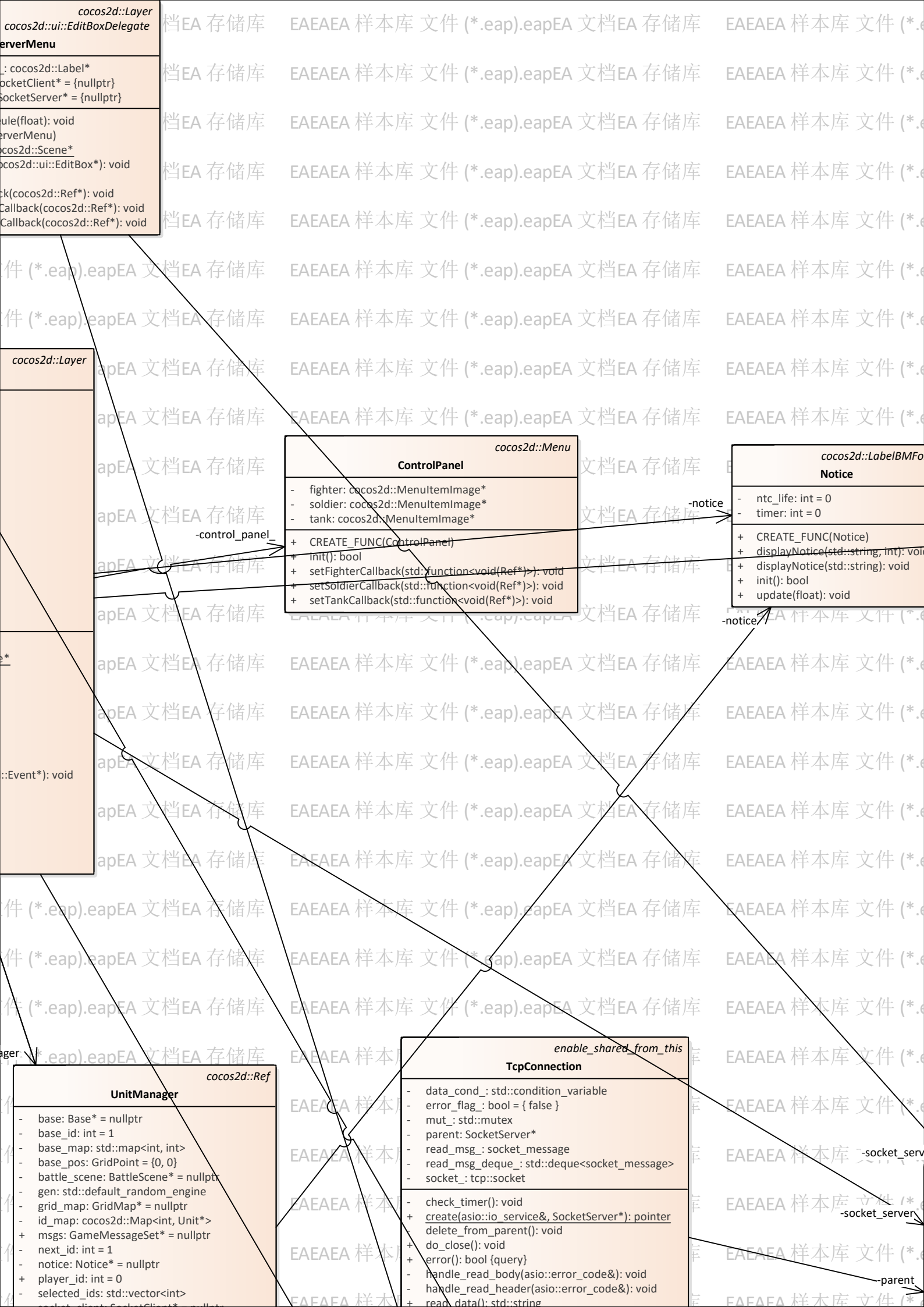
- m_f: int
- m_flag: int
- m_g: int
- m_h: int
- m_p_parent: Grid*
- m_x: int
- m_y: int

- + getF(): int
- + getFlag(): int
- + getG(): int
- + getH(): int
- + getParent(): Grid *
- + getX(): int
- + getY(): int
- + Grid()
- + setF(int): void
- + setFlag(int): void
- + setG(int): void
- + setH(int): void
- + setParent(Grid*): void
- + setX(int): void
- + setY(int): void

«Enumeration
<anonymous







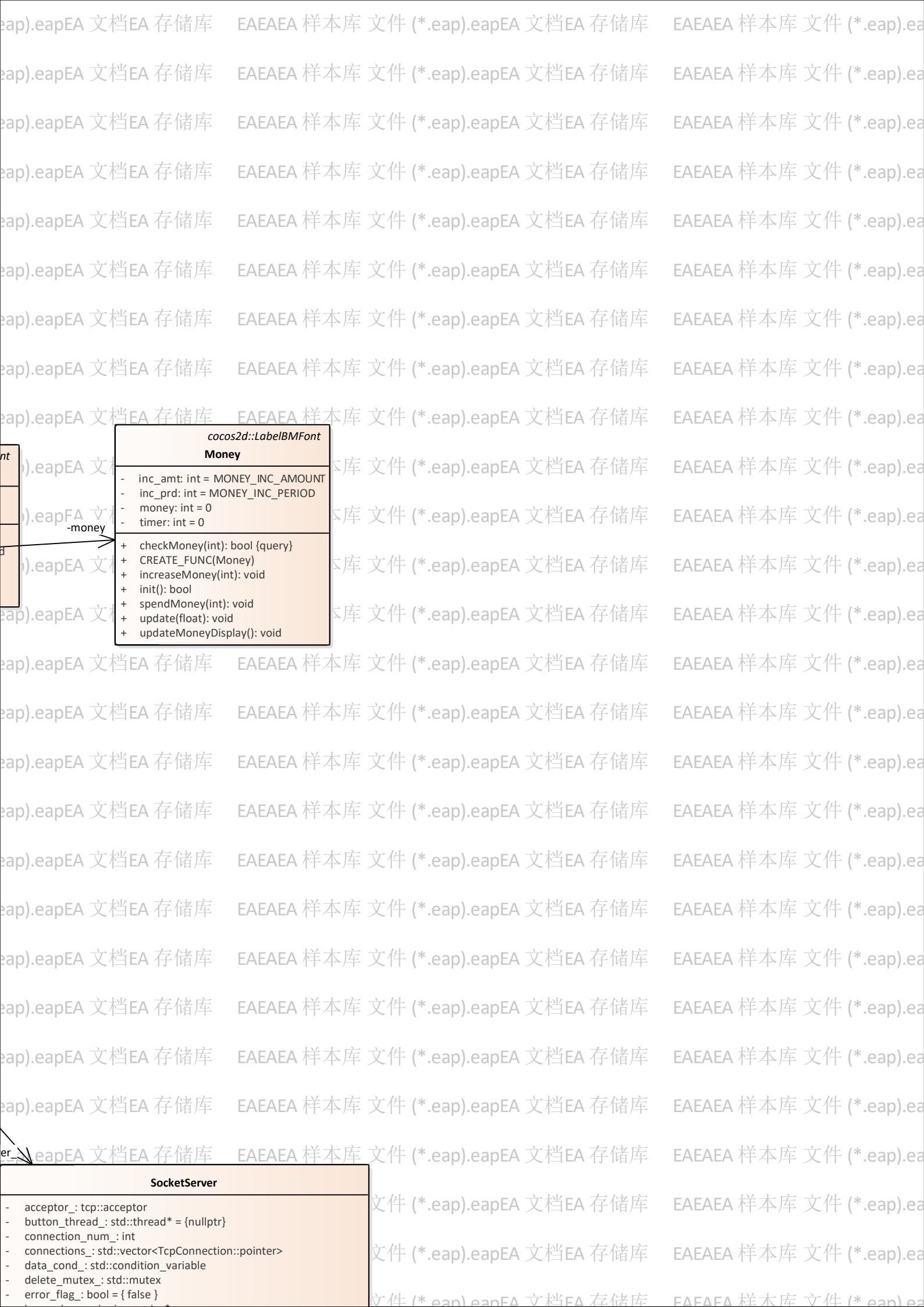


Diagram illustrating the structure and relationships of various components in a game engine, likely related to a grid-based system.

Enums:

- `enum class State { UNOCCUPIED, OCCUPIED, START, TERMINAL, INOPEN, INCLOSE }`

Structs and Typedefs:

- `shared_ptr<socket_ptr>`
- `std::shared_ptr<TcpConnection>`
- `GridRect` (struct):
 - Attributes: `gp: GridPoint`, `size: GridSize`
 - Method: `GridRect(GridPoint, GridSize)`
- `GridSize` (struct):
 - Attributes: `height: int`, `width: int`
 - Method: `GridSize(int, int)`
 - Operator: `operator/(GridSize&, int)`
- `GridPoint` (struct):
 - Attributes: `x: int`, `y: int`
 - Methods: `getDirectionVector(): GridPoint`, `GridPoint(int, int)`, `operator==(GridPoint&): bool {query}`
 - Operator: `operator-(GridPoint&, GridPoint&): GridPoint`, `operator-(GridPoint&, GridSize&): GridPoint`, `operator+(GridPoint&, GridPoint&): GridPoint`
- `GridPath` (typedef): `vector<GridPoint>`
- `GridVec` (typedef): `vector<GridPoint>`

Methods:

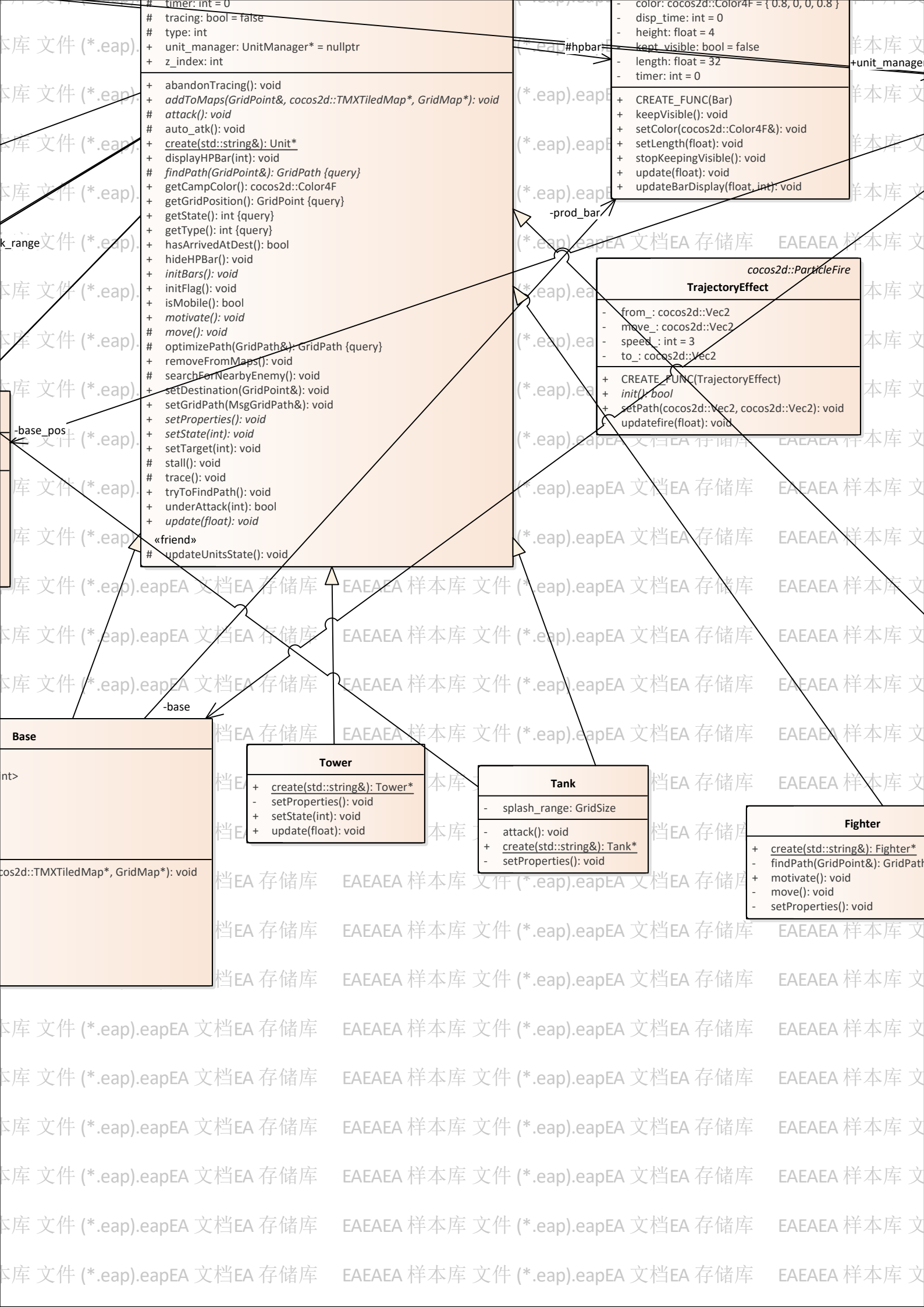
- `getGridPointWithOffset(cocos2d::Point&): GridPoint`
- `getLogicalGridMap(): std::vector<std::vector<int>>&`
- `getPoint(GridPoint&): cocos2d::Point`
- `getPointWithOffset(GridPoint&): cocos2d::Point`
- `getUnitIDat(GridRect&): std::vector<int> {query}`
- `getUnitIDat(GridPoint&): int {query}`
- `hasApproached(cocos2d::Point&, GridPoint&): bool`
- `initWithTiledMap(cocos2d::TMXTiledMap*): bool`
- `leavePosition(GridPoint&, bool): void`
- `leavePosition(GridRect&, bool): void`
- `occupyPosition(int, GridPoint&, bool): bool`
- `occupyPosition(int, cocos2d::Point&, bool): bool`
- `occupyPosition(int, GridRect&, bool): bool`

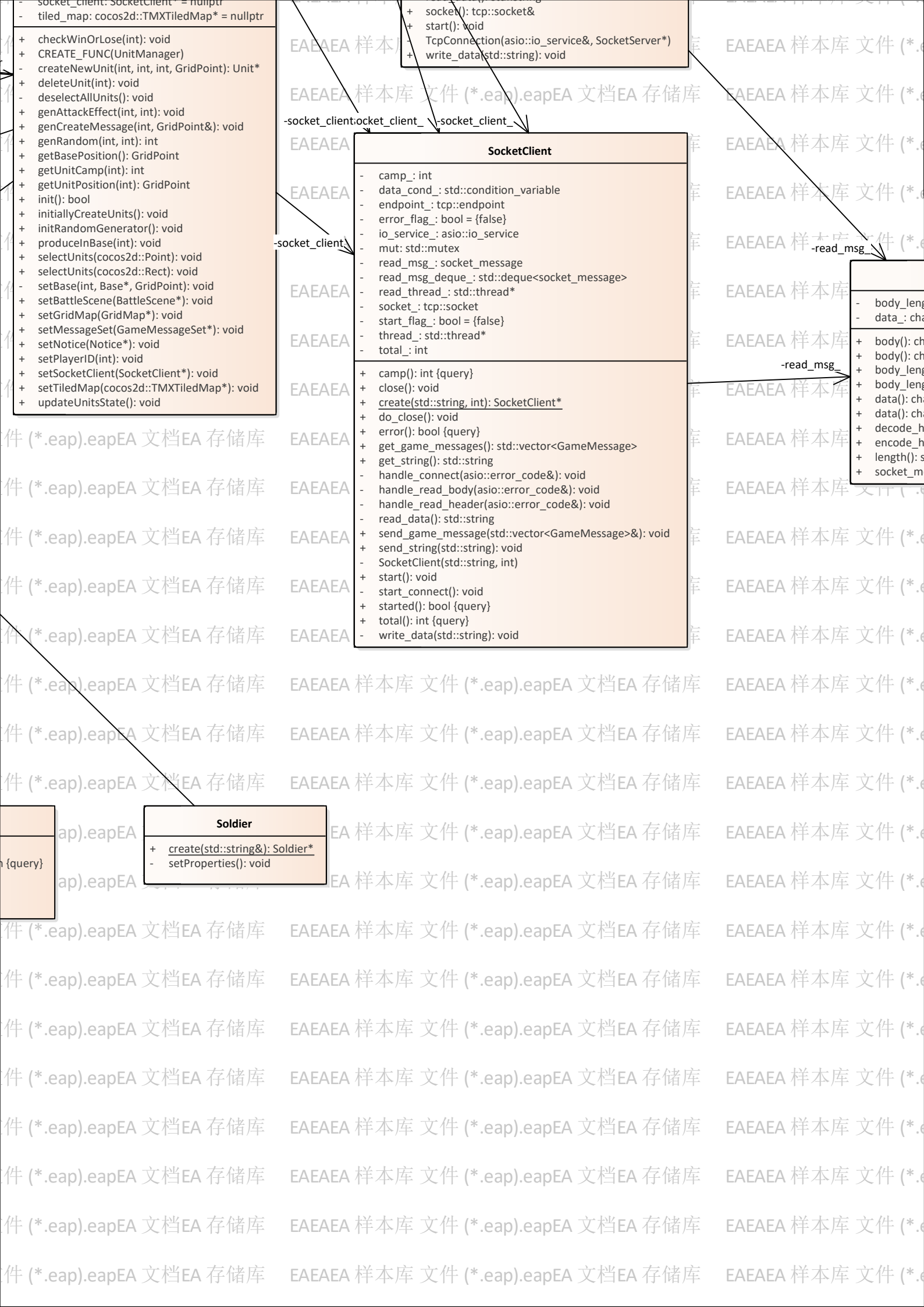
Other Components:

- `cur_prod: int`
- `period_map: std::map<int, Bar*>`
- `prod_bar: Bar* = nullptr`
- `prod_list: std::vector<int>`
- `prod_period: int`
- `prod_process: int`
- `producing: bool = false`
- `addToMaps(GridPoint&, cocos2d::Point&): void`
- `create(std::string&): Base*`
- `displayProdBar(): void`
- `initBars(): void`
- `setProperties(): void`
- `setState(int): void`
- `startProduce(int): void`
- `update(float): void`

Relationships:

- `GridRect` contains `GridPoint` and `GridSize`.
- `GridSize` is a friend of `GridPoint`.
- `GridPoint` is a friend of `GridRect`.
- `GridPath` is a vector of `GridPoint`.
- `GridVec` is a vector of `GridPoint`.





```
io_service : asio::io_service*
thread_ : std::thread*

+ button_start(): void
+ close(): void
+ connection_num(): int {query}
+ create(int): SocketServer*
+ error(): bool {query}
+ get_connection(): std::vector<TcpConnection::pointer> {query}
+ handle_accept(TcpConnection::pointer, asio::error_code&): void
- loop_process(): void
+ remove_connection(TcpConnection::pointer): void
- SocketServer(int)
- start_accept(): void
```

eap).eapEA 文档EA 存储库

```
socket_message

length : size_t
header ([header_length + max_body_length])

char* {query}
char* {query}
length(): size_t {query}
length(size_t): void
char* {query}
char* {query}
header(): bool
header(): void
size_t {query}
message()
```

eap).eapEA 文档EA 存储库

eap).eapEA 文档EA 存储库

eap).eapEA 文档EA 存储库

eap).eapEA 文档EA 存储库

eap).eapEA 文档EA 存储库

eap).eapEA 文档EA 存储库

eap).eapEA 文档EA 存储库

eap).eapEA 文档EA 存储库

eap).eapEA 文档EA 存储库

eap).eapEA 文档EA 存储库

eap).eapEA 文档EA 存储库

eap).eapEA 文档EA 存储库

eap).eapEA 文档EA 存储库

eap).eapEA 文档EA 存储库

eap).eapEA 文档EA 存储库

eap).eapEA 文档EA 存储库

eap).eapEA 文档EA 存储库

文件 (*.eap).eapEA 文档EA 存储库

文件 (*.eap).eapEA 文档EA 存储库

文件 (*.eap).eapEA 文档EA 存储库

文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库

EAEAEA 样本库 文件 (*.eap).eapEA 文档EA 存储库