



**Bullet**

**Play**



```
<<I
I
+const getLi:
+const getPos:
+const getPos:
```

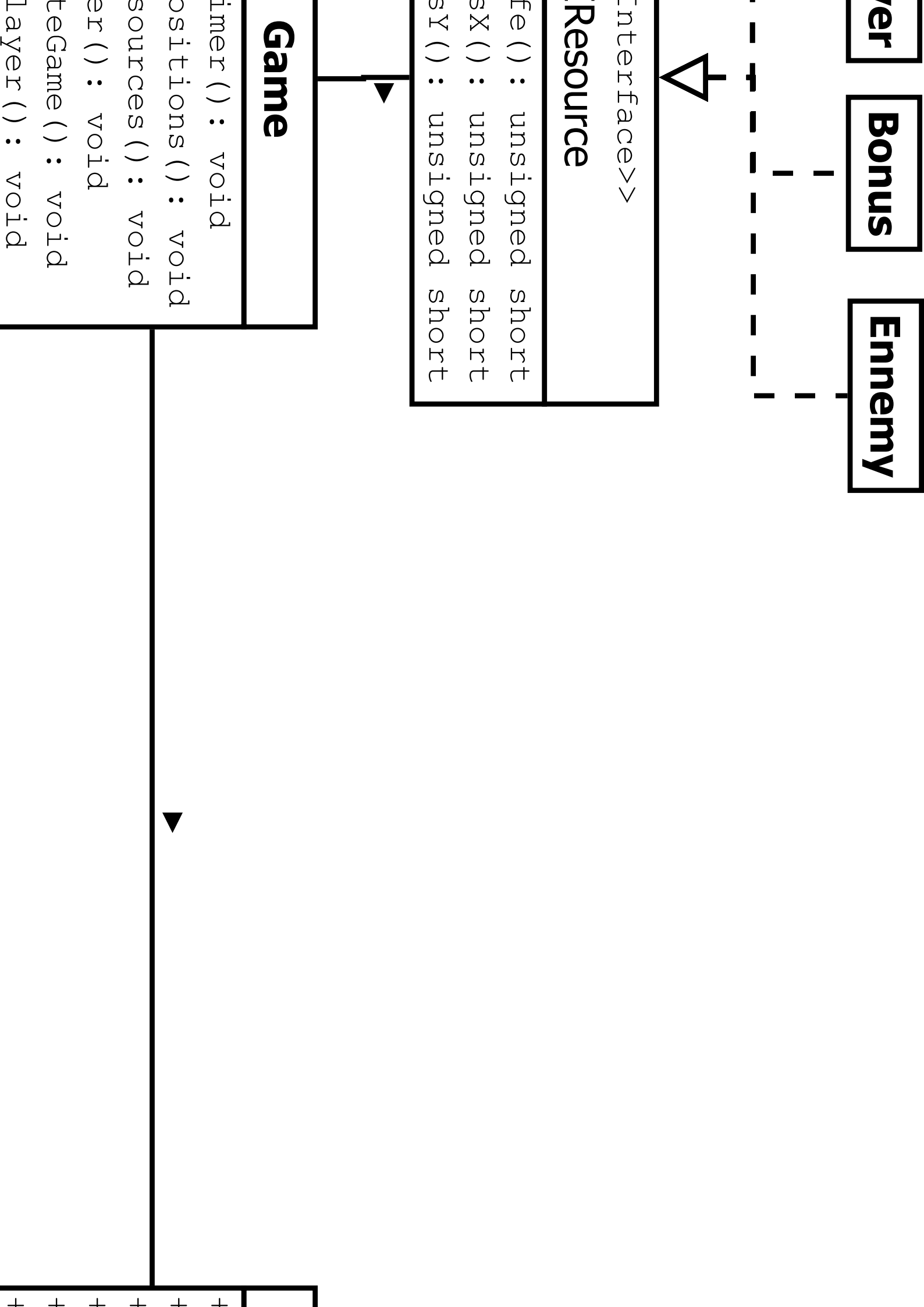
**Timer**

+timer: std::chrono

+restart(): void  
+refresh(): void  
+currentTime(): const std::chrono &

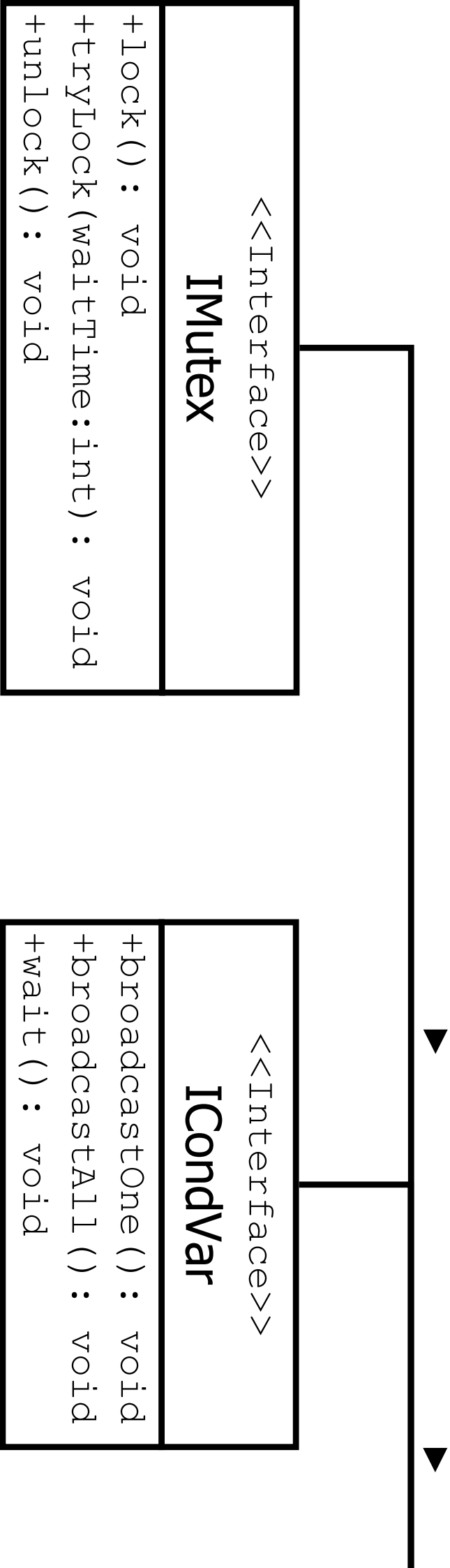
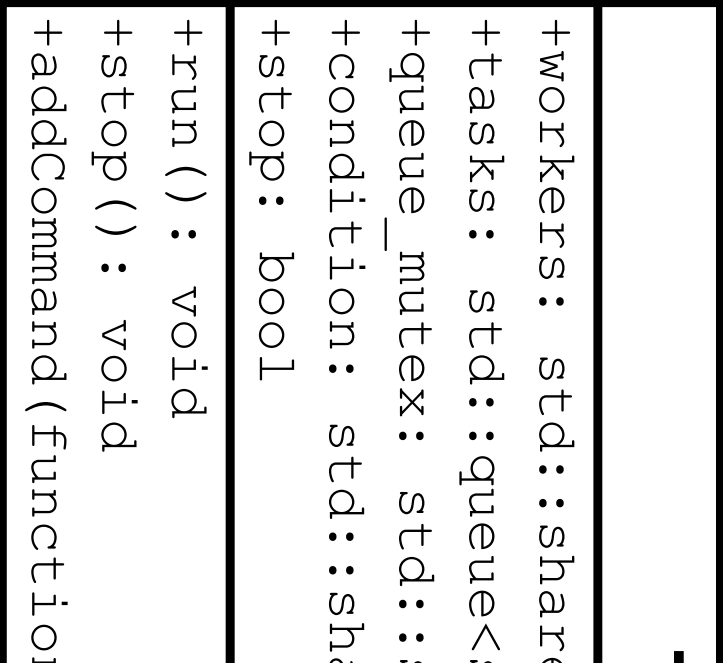
An arrow points from the bottom of the 'Timer' box to the top of the code block, indicating a relationship or flow.

```
+updateT:
+updateP:
+checkRe:
+addPlaye
+termina
+removeP:
```



## Script

```
-currentAction(): const Script::Action &
-goToNextAction(): void
-goToPrevAction(): void
-restart(): void
-addAction(action:const Script::Action &): void
-removeAction(actionId:int): void
```



# ThreadPool

```
ed_ptr<IThread>  
std::function<void()>>  
shared_ptr<IMutex>  
ared_ptr<ICondVar>
```

```
n:const std::function &): void
```



<<Interface>>

## IThread

```
+create(functor:T,param:U): void  
+wait(): void  
+exit(): void
```

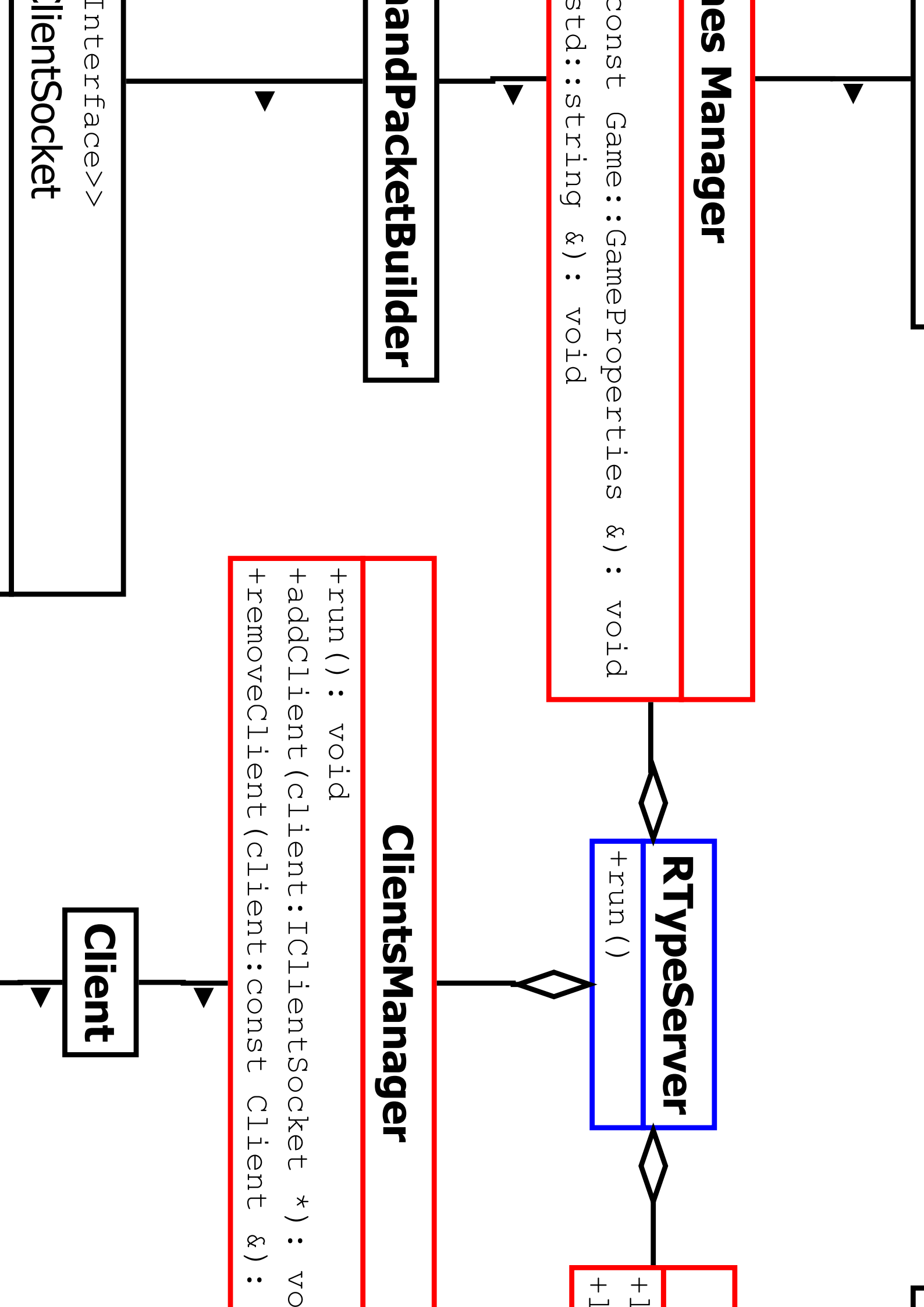


## Game

```
+run(): void  
+createGame(gameProperties:object)  
+removeGame(gameName:const string)
```

## UdpComm

<<...  
IC



## ScriptLoader

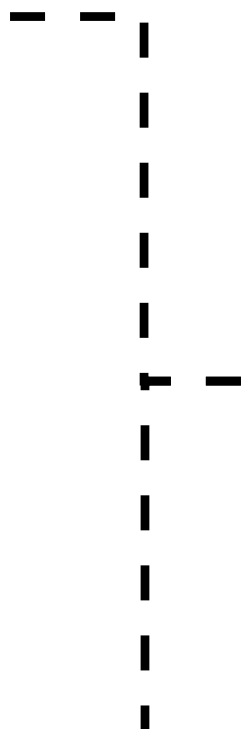
```
loadScript (scriptName:const std::string &): void  
loadAll(): void
```

ScriptParser

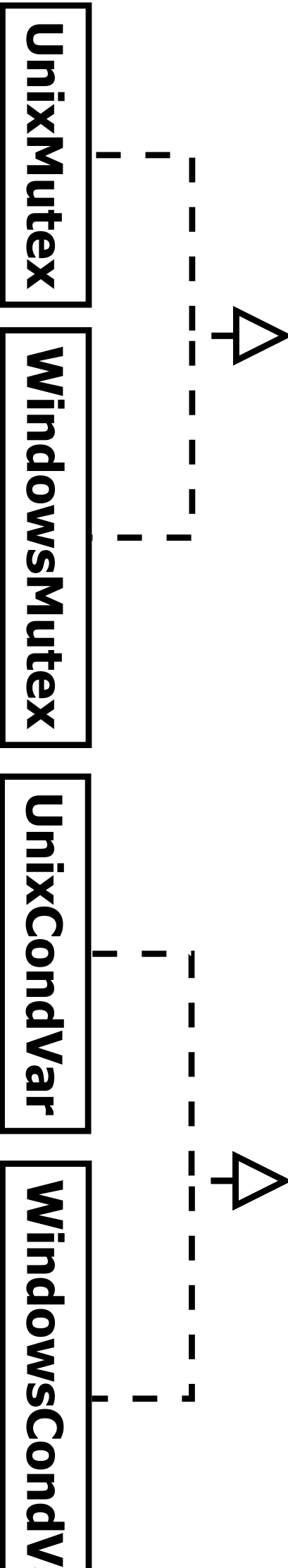
## <<Interface>> IServerSocket

```
+createServer(port:int,queueSize:int): void  
+getNewClient(): std::shared_ptr<IClientSocket>  
+closeServer(): void
```

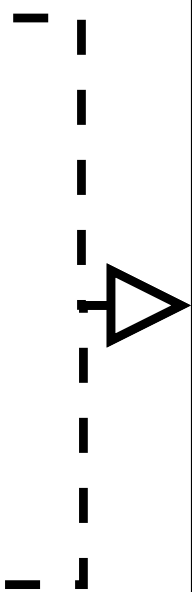
```
id  
void
```







```
+start(): void
```



ar

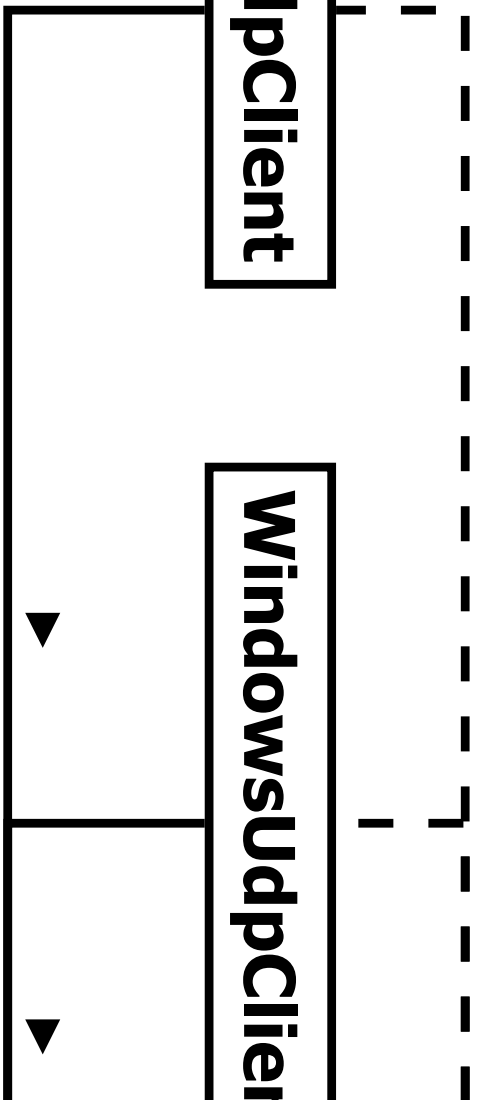
UnixThread

WindowsThread

```
+connect(addr:const st  
+initFromSocket(socket  
+receive(sizeToRead:in  
+send(message:const IC  
+closeClient(): void  
+nbBytesToRead(): int
```

UnixUdpClient

WindowsUdpClient

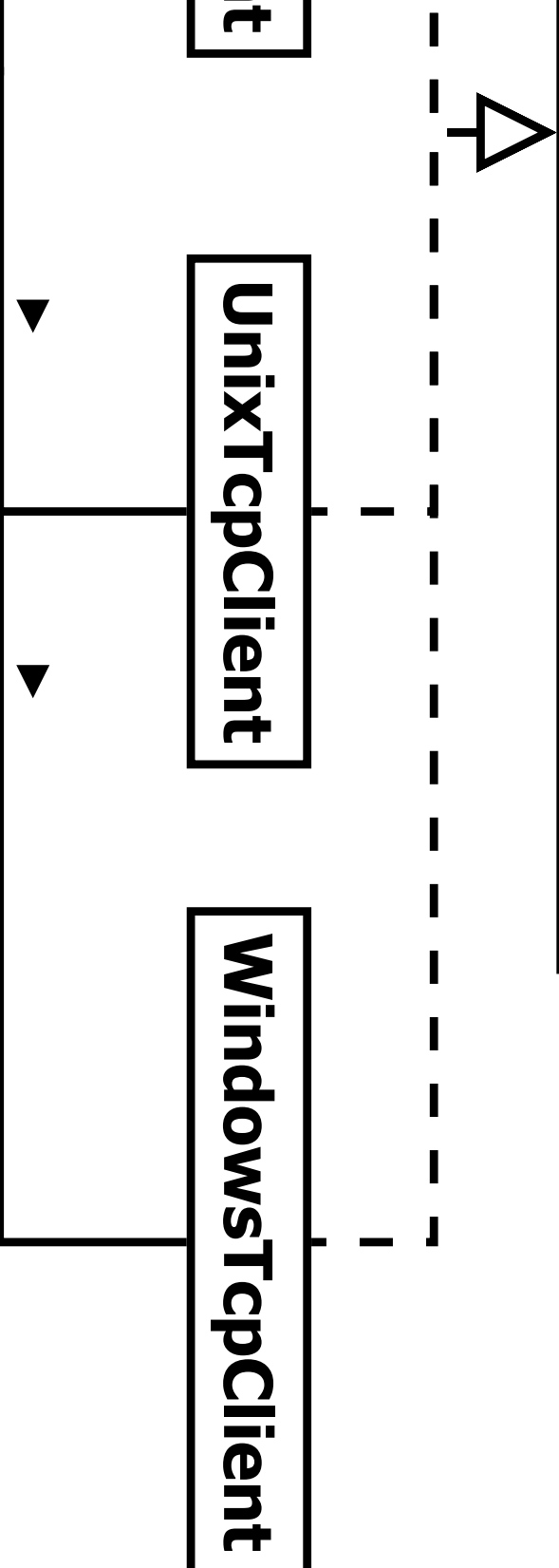


Netwo

```
+addSocket(s  
+removeSocke  
+operator()
```

```
ad::string &,port:int): void  
:void *): void  
nt): IClientSocket::Message  
ClientSocket::Message &): void
```

**TcpCommandPacketBuilder**



**rkManager**

```
socketFd:int)  
et(socketFd:int)  
( ): void
```

|

**UnixTcpServer**

|

**WindowsTcpServer**