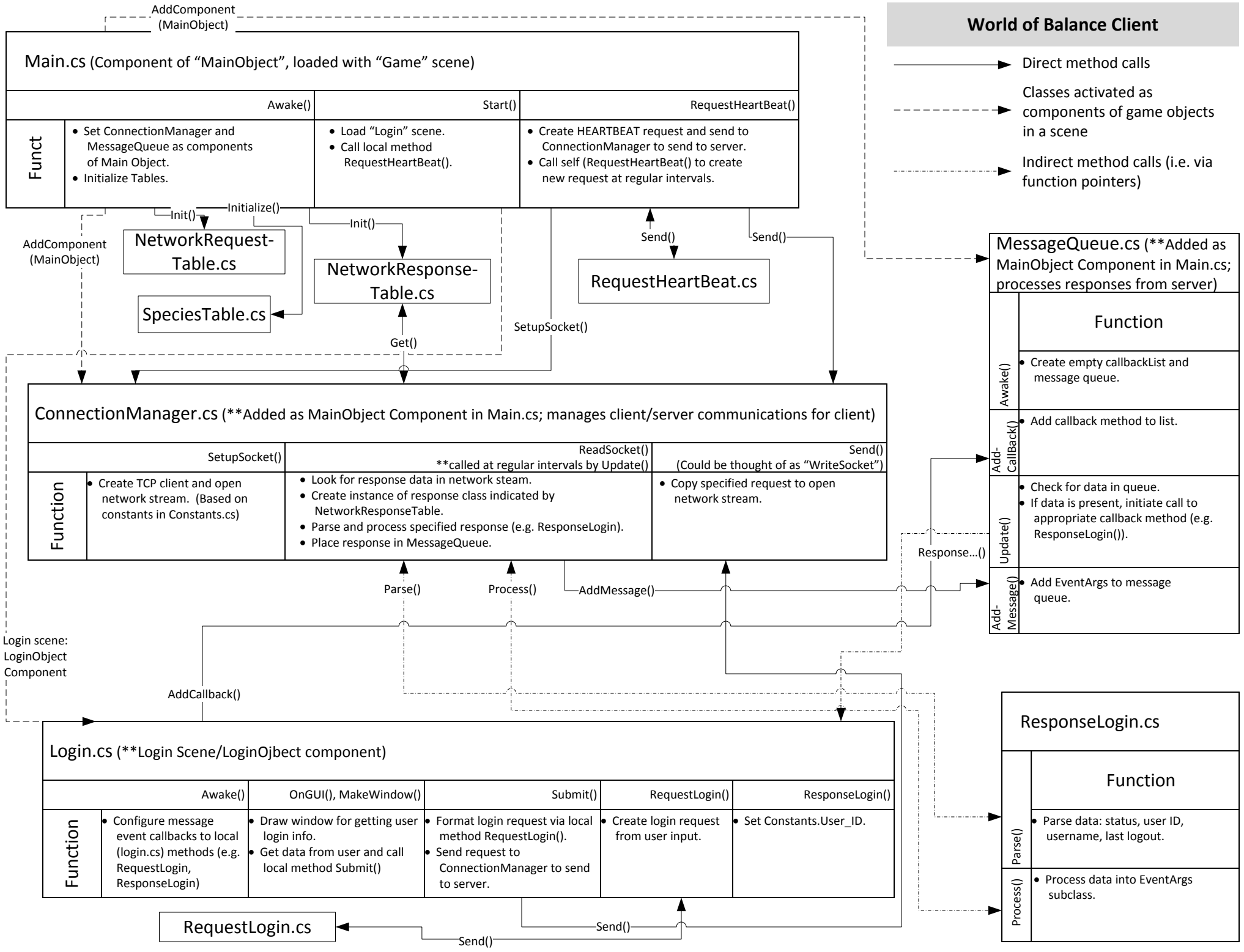
World of Balance Client
Direct method calls
Classes activated as components of game objects in a scene
Indirect method calls (i.e. via function pointers)
AddComponent (MainObject)
Main.cs (Component of "MainObject", loaded with "Game" scene)
Awake()
Start()
RequestHeartBeat()
Funct
• Set ConnectionManager and MessageQueue as components of Main Object.
• Initialize Tables.
• Load "Login" scene.
• Call local method RequestHeartBeat().
• Create HEARTBEAT request and send to ConnectionManager to send to server.
• Call self (RequestHeartBeat() to create new request at regular intervals.
Init()
Initialize()
Init()
Send()
Send()
AddComponent (MainObject)
NetworkRequest-Table.cs
NetworkResponse-Table.cs
RequestHeartBeat.cs
SpeciesTable.cs
Get()
SetupSocket()
MessageQueue.cs (**Added as MainObject Component in Main.cs; processes responses from server)
Function
Awake()
• Create empty callbackList and message queue.
Add-CallBack()
• Add callback method to list.
Update()
• Check for data in queue.
• If data is present, initiate call to appropriate callback method (e.g. ResponseLogin()).
Add-Message()
• Add EventArgs to message queue.
ConnectionManager.cs (**Added as MainObject Component in Main.cs; manages client/server communications for client)
SetupSocket()
ReadSocket() **called at regular intervals by Update()
Send() (Could be thought of as "WriteSocket")
Function
• Create TCP client and open network stream. (Based on constants in Constants.cs)
• Look for response data in network steam.
• Create instance of response class indicated by NetworkResponseTable.
• Parse and process specified response (e.g. ResponseLogin).
• Place response in MessageQueue.
• Copy specified request to open network stream.
Response...()
Parse()
Process()
AddMessage()
Login scene: LoginObject Component
AddCallback()
Login.cs (**Login Scene/LoginOjbect component)
Awake()
OnGUI(), MakeWindow()
Submit()
RequestLogin()
ResponseLogin()
Function
• Configure message event callbacks to local (login.cs) methods (e.g. RequestLogin, ResponseLogin)
• Draw window for getting user login info.
• Get data from user and call local method Submit()
• Format login request via local method RequestLogin().
• Send request to ConnectionManager to send to server.
• Create login request from user input.
• Set Constants.User_ID.
RequestLogin.cs
Send()
Send()
ResponseLogin.cs
Function
Parse()
• Parse data: status, user ID, username, last logout.
Process()
• Process data into EventArgs subclass.