1. Damit das funktionieren kann muss der server multi threaded sein => concurrent server
2. Concurrent collection -> wir brauchen eine lobby
3. Wenn der 1. Spieler reinkommt wird der dort abgelegt und wenn der 2. Reinkommt wird der abgelegt
4. Concurrent queue = meine lobby

-ist thread safe in sofern das 2 threads gleichzeitig etwas reingeben können

-ist eine möglichkeit in die ich die sachen reinschreibe  
-enque ist was reingeben   
-tryDequeue versuche etwas rauszunehmen ressut ist ein boolean konnte ich etwas rausbekommen

-thioretisch wenn 2 threads genau gleichzeitig reinkommen würden beide bei trydequeue no zurückbekommen

1. Wenn ich ins battle reinkomme für ich ein tryDequeue aus wenn da true zurückkommt dann kann ich battle starten mit diesen beiden usern und der nun neu reingekommene bekommt dann das log und der andere dadurch das ich es in die dictionary queue seinen log schreibe
2. Wenn ich ins battle reinkomme und tryDequeue gibt false zurück dann mach ich ein enque und gebe mich selber in die Enque

Oder ich mach ein polling Prinzip sprich der thread wird mit enque in die concurrent queue reingegeben und schaut im halben sekunden takt ob jz schon wer 2. Drinen ist

Dictionary concurrent queue da kann ich den username und den log hinterlegen dadurch

-TryRemove = ein befehlt schau ob es was gibt zu diesen key und entferne es wenn du es gefunden hast

Dictionary concurrent queue nehme ich für die ergebnisse von dem anderen

Ich bleibe solange im battle bis ich den log-Message zurückbekomme

Siehe produceConsumer problem: diese queue brauche ich dann auch   
  
Andere variante wäre ich habe einen extra thread der das battle aufruft   
  
wenn dann 2 user in dieser collection drinnen sind ruft dieser get 2 mal auf und dann ruft er das battle mit u1 und u2 auf

Den daraus entstandenen log legt er wo ab

Mit dem battle request vom u1 geht macht er ein add und wartet auf ein ergebnis (battle log) von diesem battle thread

Macht nen eintrag ins dictionary mit u1, log und u2, log   
  
u1 und u2 thread probieren aus diesem dictionary ihre daten zu bekommen

WO DAS THREADING BEGINNEN:  
da wo ein HttpClient erstellt wird… dieses sollte in einen thread ausgelagert werden

Also in HttpServer.cs in der zeile: var client = new HttpClient();