

Playground

Referee - *startTrial*, *assertTrialDecision*, não são pontos de bloqueio, sendo apenas locais onde o “referee” actualiza variáveis que acordam “contestants”.

Contestants - *pullTheRope* é o único ponto não bloqueante, onde apenas existe um **sleep** de tempo aleatório, para simular o “contestant” a puxar a corda.

Existem apenas dois pontos de bloqueio:

- *waitForStartTrial*, onde o “contestant” aguarda que o “referee” o acorde quando fizer *startTrial*.
- *waitForAssertTrialDecision*, o “contestant” aguarda que o “referee” o acorde quando tomar uma decisão acerca da trial.

Referee Site

Referee - *annouceNewGame*, *declareMatchWinner*, *declareGameWinner* e *endOfMatch* não são pontos de bloqueio, servem apenas para atualização de variáveis.

Existem três pontos de bloqueio:

- *waitForInformReferee*, o “referee” aguarda que os “coaches” o informem que a sua equipa já está seleccionada, fica bloqueado enquanto os dois não o tiverem informado.
- *waitForAmDone*, o “referee” aguarda que o último “contestant” a terminar o “pullTheRope” o avise que terminaram todos e que pode tomar uma decisão acerca da trial. Saindo deste ponto de bloqueio, o contador que indica se o contestant é o último, ou seja igual a 6, é recolocado a 0.
- *waitAllPositioned*, o “referee” aguarda que o último “contestant” a estar em “Stand in Position”, o acorde para poder dar início à trial. O contador que indica se o contestant é o último, ou seja igual a 6, é recolocado a 0.

Contestants - Não possui pontos bloqueantes, sendo apenas métodos que atualizam variáveis que vão acordar o “referee”. Como por exemplo, *amDone*, onde o contador *amDoneCounter* é incrementado e quando chega a 6, que indica que é o último “contestant” a terminar, coloca a variável booleana que vai acordar o “referee” a **true** e faz *notifyAll*. Existe ainda o método *positioned*, que é idêntico ao *amDone*, em termos de funcionamento, sendo que neste caso a variável alterada que acordará o “referee” é relativa a todos os “contestants” estarem posicionados.

Coaches - Não possui pontos bloqueantes, sendo apenas um método utilizado para alterar a variável referente ao *informReferee*, que acordará o “referee” quando ambos os “coaches” tiverem as suas variáveis booleanas a **true**.

Bench

Referee - *assertTrialDecision* e *wakeUp*, são pontos não bloqueantes, servem apenas para alterar variáveis que vão acordar outras entidades.

A entidade “referee”, nesta zona partilhada possui apenas um ponto de bloqueio, que é quando pretende fazer *callTrial*, tem de aguardar que estejam 10 “contestants” no banco e os 2 “coaches” a aguardar pela *callTrial*.

Contestants - *seatDown* e *followCoachAdvice*, são pontos não bloqueantes, onde são alteradas variáveis que poderão acordar outras entidades.

Existe um ponto bloqueante:

- *waitForCallContestants*, os “contestants” aguardam que o “coach” os chame para irem a jogo, sendo as condições para sair do ponto bloqueante, saber se o “coach” já fez a sua seleção, e se o “contestant” é um dos selecionados. Saindo deste ponto de bloqueio o número de “contestants” no banco é decrementado, e é guardado o *id* do último “contestant” a levantar-se, para ser este a acordar o “coach” no método *followCoachAdvice*.

Coaches - *callContestants* é um ponto não bloqueante, onde o “coach” seleciona aleatoriamente os “contestants” que vão a jogo e diz que já efetuou a seleção, podendo assim os “contestants” selecionados acordar.

Pontos bloqueantes, existem os seguintes:

- *waitForCallTrial*, o “coach”, aguarda que o “referee”, faça *callTrial*, para poder acordar.
- *waitForAssertTrialDecision*, onde o “coach” aguarda que o “referee” o acorde quando tomar uma decisão referente à trial.
- *reviewNotes*, que é acordado quando estiverem 10 “contestants” no banco, ou seja todos sentados.
- *waitForFollowCoachAdvice*, aguarda que o último jogador da sua equipa a levantar-se para ir jogar a trial, o avise de forma a poder avisar o “referee” que a sua equipa está pronta.

General Information Repository

Esta zona partilhada, é constituída pelo Log, pelas classes Match, Game e Trial, sendo estas três classes responsáveis pela gestão de um Match.

Todas as entidades interagem com o Log, de forma a que as informações referentes a forças dos “contestants”, posições onde estão colocados numa trial, estado em que se encontram as entidades, se encontrem atualizadas.

É nesta zona, que é escrito tudo o que acontece no problema, num ficheiro “.log”.