

Battleship Socket Game

Документация

1. Увод.

„Battleship Socket Game“ е игра базирана на комуникация с уеб-сокети която пресъздава класическата идея за „игра на потоци кораба“.

В играта двама играчи се опитват да потопят корабите си като победителя е този който успее първи да потопи противниковият.

За да се реализира на комуникацията между играчите и сървъра на играта са използвани уеб сокети от библиотеката `java.net` а за да се реализира графичният интерфейс се използва библиотеката `javax.swing`.

Играта е реализирана с използване на `java` тредове за сървърната комуникация и симулираните клиенти (проху клиенти), работещи на сървъра и комуникиращи с клиентите.

2. Програмен поток.

Програмният поток на „Battleship Socket Game“ е следният.

1. Стартират се сървъра (`BattleShipServer`) като се задава порт на който той да очаква клиенти (слуша).
2. Всеки клиент стартира своята програмата (`BattleShipClient`) като подава като параметри името на `host`-а (при нас `localhost`) и името на `port`-а на който да се опита да се свърже клиентската апликация.
3. След стартирането на клиентската част `“BattleShipClient”` пита клиента си за своето име.
4. Клиентите се опитват да се свържат със сървъра като ако в даденият момент това не е възможно опитват в продължение на 50 секунди в интервал от 5 секунди.

5. Ако клиент успее да се свърже със сървъра на сървъра се създават комуникационните канали и графичният интерфейс за него.
6. Сървъра допуска до два клиента като след това влиза в изчакване (до приключване на играта).
7. Двата клиента сега са стартирали играта си и с помощта на графичният интерфейс получават информация за предстоящите им стъпки.
8. Играта се играе на редове като те са:
 - a. Поставяне на кораб на полето. Ако при тази стъпка има колизия (съответствие) при на мястото на поставяне играта се рестартира. Ако клиента не избере поле до насроченото му време (10 секунди) такова му бива поставено автоматично.
 - b. Стрелба по противника. Ако клиента не избере поле до насроченото му време (10 секунди) изстрела не се произвежда.
9. Край на играта настъпва когато един от корабите е свален. Тогава конзолата изписва името на победителя.

3. Описание на класовете.

Описанието на класовете в „Battleship Socket Game“ е както следва:

1. **BattleShipServer;**

Този клас е практическият сървър на апликацията. Той приема integer представляващ порта на който искаме сървъра да слуша за появата на клиенти. Сървъра наследява класа Thread и логиката му може лесно да се паралелизира за да се поддържат на много игри едновременно.

2. **BattleShipClient;**

Този клас е практическият клиент на апликацията. Той приема параметрите хоста и номер. Двата представляват съответно; името на хоста, порта на хоста. Този клас прави запитване с InetAddress към сървъра което ако успешно позволява на „BattleShipServer“ да създаде сокет за комуникация а „ServeOneClient“ да подготви In/OutputStream обекти за комуникация. При успешно свързване клиента също създава In/OutputStream обекти.

„BattleShipClient“ също наследява класа Thread и логиката му (специално метода „listenToServer“) се възползва от това.

3. ServeOneClient;

Този клас прави от сокета пратен му от сървъра, InputStreamReader и OutputStreamWriter. InputStreamReader-а този клас използва за да слуша за команди от клиентите който да прати на обекта „BattleShipGameLogic“. OutputStreamWriter-а също се праща на класа „BattleShipGameLogic“ за да може тя (логиката на играта) да праща команди на клиента. Класа „ServeOneClient“ наследява класа Thread и използва run() метода му за да навлезе в безкраен цикъл от очакване на команди от който излиза само при край на играта.

4. BattleshipGameLogic;

Този клас представлява същинската игрова логика в апликацията. Този клас не наследява класа Thread, а се използва като монитор за две нишкови обекта създадени от класа „BattleshipClientThread“. Този клас благодарение на двете нишки възможността му за комуникация с клиентите е основен за играта и в него се намират синхронизираните методи който „BattleshipClientThread“ обхожда.

5. BattleshipClientThread;

Този клас наследява от класа Thread и използва класа игровата логика за да мине през двата стадия на играта: Поставянето на Кораб, Стреляне по противника.

6. BattleShipGUI;

Този клас наследява от класа JFrame и представлява графичният клиентски интерфейс на апликацията. Този клас разполага с таймер който се рестартира при началото на всеки ход и приключва същият при изтичане на времето. Този клас използва също така и BorderLayout, BoxLayout, JButton, JLabel и JPanel от библиотеката javax.swing.