Como criar um 'High score' com Java ME

Cada jogo deve ter um sistema de *High score* para permitir aos jogadores desafiarem outros jogadores ou a eles próprios. Este artigo demonstra uma maneira de implementar essa funcionalidade em Java ME.

O Java ME usa o Record Management System (RMS), permitindo aos desenvolvedores salvar e restaurar dados em sessões de aplicações diferentes. Assim, para a criação rápida de um sistema *High score* com nome e pontos do usuário, você pode usar o seguinte código.

Código fonte

Adicionando um novo *Hiscore*

```
...
// Abrimos o recordstore
RecordStore highscore = RecordStore.openRecordStore("High Score", true);

// Para adicionar um novo ''hiscore'' usamos uma ''String'' separada por virgula
String str = new String("John,3400");
byte [] strb = str.getBytes();
int id = highscore.addRecord(strb, 0, strb.length);

highscore.closeRecordStore();
...
```

Recuperando as pontuações

```
...
RecordStore highscore = RecordStore.openRecordStore("High Score", true);

// Para ler todas as ''hiscores'' salvas no RMS
RecordEnumeration enum = highscore.enumerateRecords(null, null, false);
int id;
byte[] record;
String str;
while (enum.hasNextElement( )) {
     id = enum.nextRecordId( );
     record = highscore.getRecord(id);
     str = new String(record);
     // Opera com ''string'' (str) para extrair o nome e os pontos
}
highscore.close();
...
```

Como recuperar os dados classificados corretamente

Para classificar os resultados do maior para o menor, uma opção é implementar a *interface* **RecordComparator**, e usá-la ao chamar o método **enumerateRecords()** da classe *RecordStore*.

A implementação básica de um *RecordComparator* que classifica os resultados armazenados é a seguinte:

```
...
class ScoresComparator implements RecordComparator
{
 public int compare(byte[] arg0, byte[] arg1)
 {
  int score0 = 0, score1 = 0;

  for(int i = 0; i < arg0.length; i++)
  {
   if(arg0[i] == ',')
   {
    score0 = Integer.parseInt(new String(arg0, i + 1, arg0.length - i - 1));
   }
  }
  for(int i = 0; i < arg1.length; i++)
  {
   if(arg1[i] == ',')
   {
    score1 = Integer.parseInt(new String(arg1, i + 1, arg1.length - i - 1));
   }
  }

  if(score0 < score1)
   return 1;
  else if(score0 == score1)
   return 0;
  else
   return -1;
 }
}
...
```

Assim, a chamada anterior para **enumerateRecords()** vai mudar:

```
RecordEnumeration enumerador = highscore.enumerateRecords (null, ScoresComparator novo (), false);
```


(C) Copyright Nokia 2013. All rights reserved.

Download

Você pode baixar o exemplo mostrando no código apresentado neste artigo aqui: Media: HighScoreDatabaseMIDlet.zip

Notas de implementação

- Você deve usar *try-catch* para capturar exceções e mensagens sobre a operação de E/S ou lançar uma exceção (*throws*).
- Quando estiver escrevendo ou lendo dados do RMS, você deve usar *threads* separadas para evitar o bloqueio da *thread* principal do MIDlet.

Links relacionados

- "Data Handling" section on Java ME Developer's Library
- javax.microedition.rms package JavaDocs

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.