Charactar

```
# boolean _over
# boolean _fallen
# Stack<String> _status
# int _attraction
# int _pendingLikeChange
# ArrayList<TreeNode> _stage1
# ArrayList<TreeNode> _stage2
# ArrayList<TreeNode> _stage3
# ArrayList<TreeNode> _currentStage
# String _descrip
# String _name
# Player _player
# boolean _isRichard
# int x
# int y
```

```
+ boolean isOver()
+ boolean hasFallen()
+ String getStatus()
+ int getAttraction()
+ ArrayList<TreeNode> getStage()
+ String getDescrip()
+ String getName()
+ boolean getIR()
+ boolean setOver(boolean isOver)
+ boolean setFallen(boolean hasFallen)
+ String changeStatus()
+ void friendify()
+ String maintainStatus(int oldAttraction)
+ int changeAttraction(int change)
+ boolean updateTree(TreeNode newHead)
+ int probeTree()
+ int probeTreeHelper(TreeNode node)
```

*Note: Charactar has 3 subclasses: Jessica, Brad, and Richard. The only distinction between these subclasses is the differences in their constructors. We've set it up this way to avoid convoluted Charactar instantiations in Woo; this way, we can create Jessica, Brad, and Richard without having an overly complicated set of parameters in the constructor.

Woo

```
- Brad _brad
- Jessica _jessica
- Richard _richard
- Player _player
- boolean _gameOver
- boolean _firstTime;
```

```
+ void introduction()
+ void play()
+ static void type(String s)
+ static void delay (int milliseconds)
+ static String removePunctuation(String word)
```

Player

```
- ArrayList<Charactar> _rank
- String _name
- boolean _hasFriend
- boolan _dead
```

```
+ ArrayList<Charactar> getRank()
+ void sortRank()
+ void addToRank(Charactar character)
+ String getName()
+ String setName(String name)
+ boolean hasFriend()
+ void giveFriend()
+ boolean isDead()
+ boolean die()
```

abstract TreeNode

```
# ArrayList<TreeNode> _children
# int _likeChange
# Charactar _character
# Player _player
```

```
+ abstract void interact()
+ ArrayList<Integer> getChildrenLikeChanges()
+ ArayList<TreeNode> getChildren()
+ static void type(String s)
+ static void delay (int milliseconds)
```

*Note: Each node in the storyline tree will be coded individually (a class for each node). It will extend TreeNode, and each will have a specialized interact() method (this is essentially a chatbox)

Scanny

```
- _response
```

```
+ String toString()
+ static void type(String s)
+ static void delay (int milliseconds)
```

*Note: Much of Scanny's functionality is contained in its constructor. It works similarly to Scanner but also gives the player the opportunity to ask a friend for advice before providing a response.