

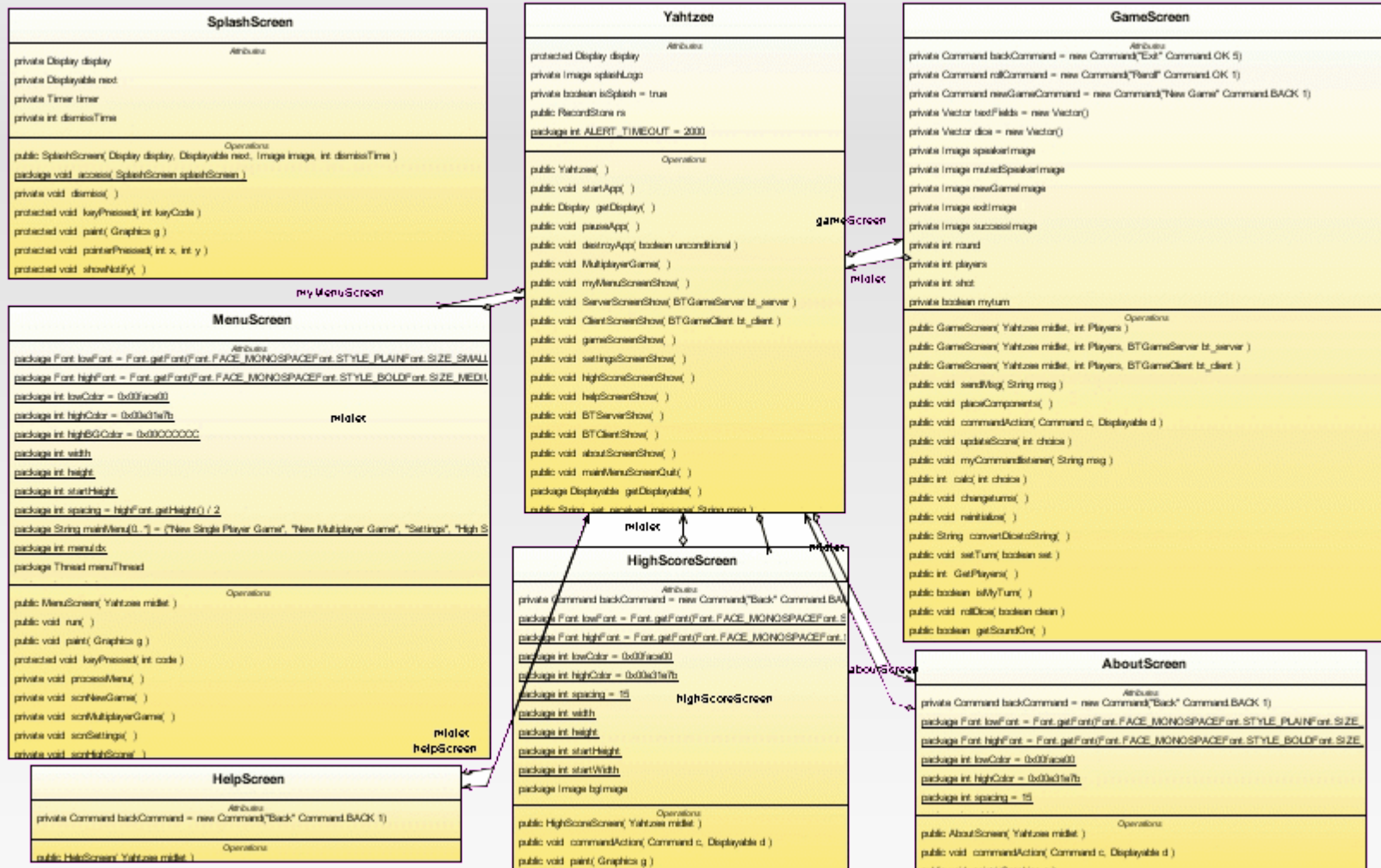
# Yet Another Yahtzee

A mobile adaptation of the classic Yahtzee game.

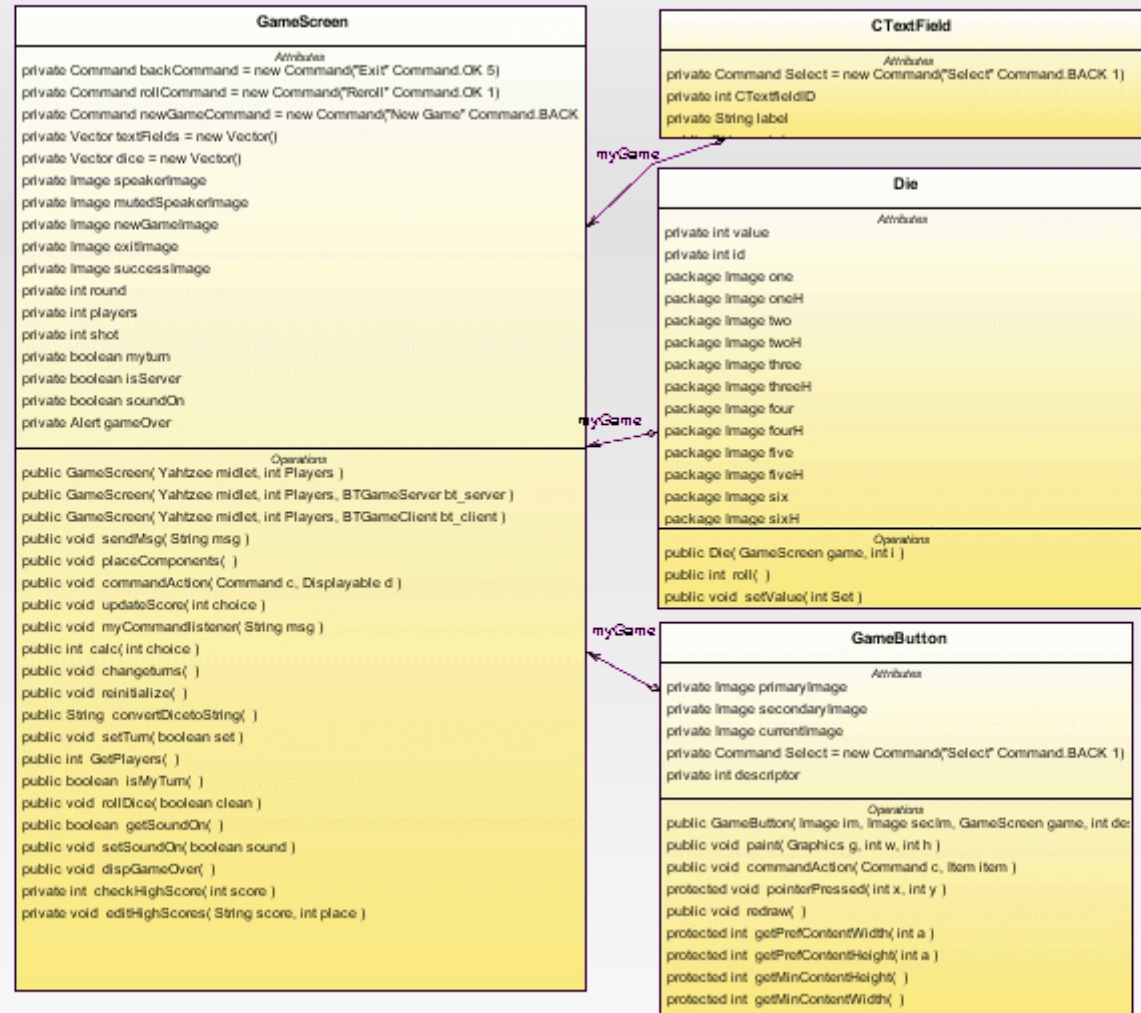
# The game in brief

- Based on the classic Yahtzee game.
- Single or peer to peer
- Having five dice at his disposal the player has to gain as many points as possible based on certain rules.
- Three shots in each turn
- Player can decide which dice to re-roll

# UML Diagram for Screen



# UML Diagram for Game Screen

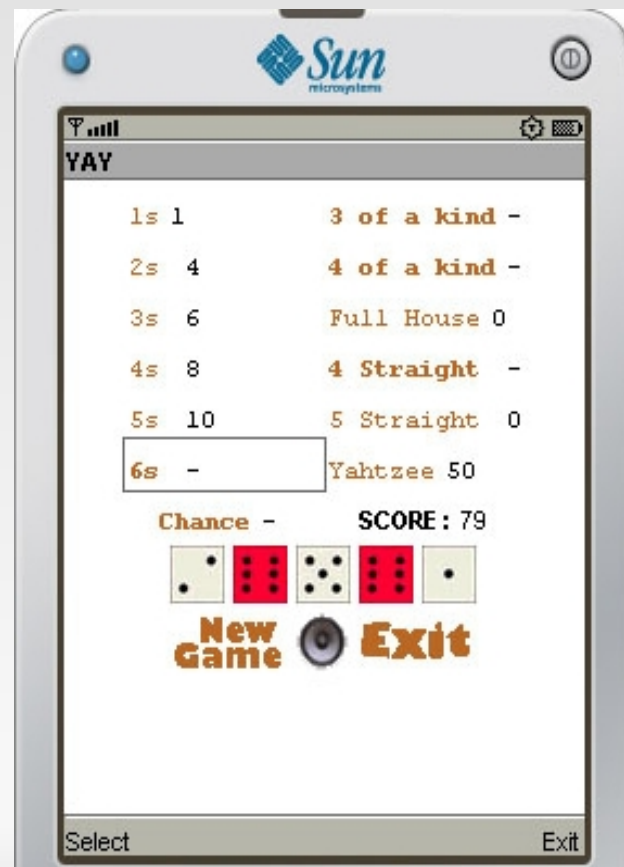


# Peer to Peer Yahtzee

- Two players can connect via Bluetooth link
- One acting as server and the other as client
- Players have to wait for their turn
- CommandListeners toggled on/off
- Players can see their opponent's moves when it's not their turn

# Main Game Screen

- ◆ Simple form, implemented using J2me High Level GUI API.



# Bluetooth peer to peer

- Two mobile devices connecting via a Bluetooth link.
- One acting as client, the other as server
- Implemented using Java's bluetooth API (JSR82)
- Server advertises service, client is searching for devices offering services
- After connection, game starts. By default, the server is the first to play
- Active player sends messages to opponent on every action
- Opponent is listening for incoming messages and updates screen accordingly

# Example



- After throwing the dice, the player decides to hold the pair of sixes.
- Sends message to opponent in a form of a string.
- `sndMsg(«h»,DiceVector.toString());`
- «h» for hold
- `DiceVector <(0,0),(1,0),(x,y)...>`.
- Where x index of die element, y boolean indicating if die is held or not
- Opponent who is listening for messages receives message and refreshes screen accordingly



# Game Over Conditions

- Game ends after all score boxes are filled (13 turns).
- In multiplayer sessions initiator waits for opponent to finish the last turn before proceeding to winner decision process
- After the game has ended, each device checks if the player has achieved a high score.

# High Score Records

- RMS managed by Yahtzee MIDlet
- 3 records in rms hold top 3 player names-scores on the device.
- High score check handled by GameScreen
- Player asked to enter name at the end of the game if a high score is achieved
- Name and score of the player stored in the highscore RMS, replacing the lowest high-score entry

# Additional Functionalities

SMS Push Registry

# SMS Push Registry

- During startApp() the application opens an connection for inbound SMS messages.
- A new thread is created to handle incoming SMSs by listening to a specified port.
- When a SMS is received the thread notifies the appication and the message is displayed
- The purpose of the SMS Push Registry is to inform user about new versions and patches for the game or updates and new products posted on the game's web page

# Yahtzee's Web Page

- Features:
- Flash Lite Demo
- OTA download and installation
- WURFL content adaptation

# Flash Lite Demo



- <http://people.dsv.su.se/~zarifis/yahtzeesite/demo/yahtzeedemo.swf>

# OTA download

- Add URL to Jar file in Yahtzee midlet's JAD file:

MIDlet-1: Yahtzee !,/res/icon.png,Yahtzee

MIDlet-Jar-Size: 446102

MIDlet-Jar-URL: <http://people.dsv.su.se/~zarifis/yahtzeesite/OTA/Yahtzee.jar>

MIDlet-Name: Yahtzee!

MIDlet-Vendor: KG Entertainment INC

MIDlet-Version: 1.0

MicroEdition-Configuration: CLDC-1.1

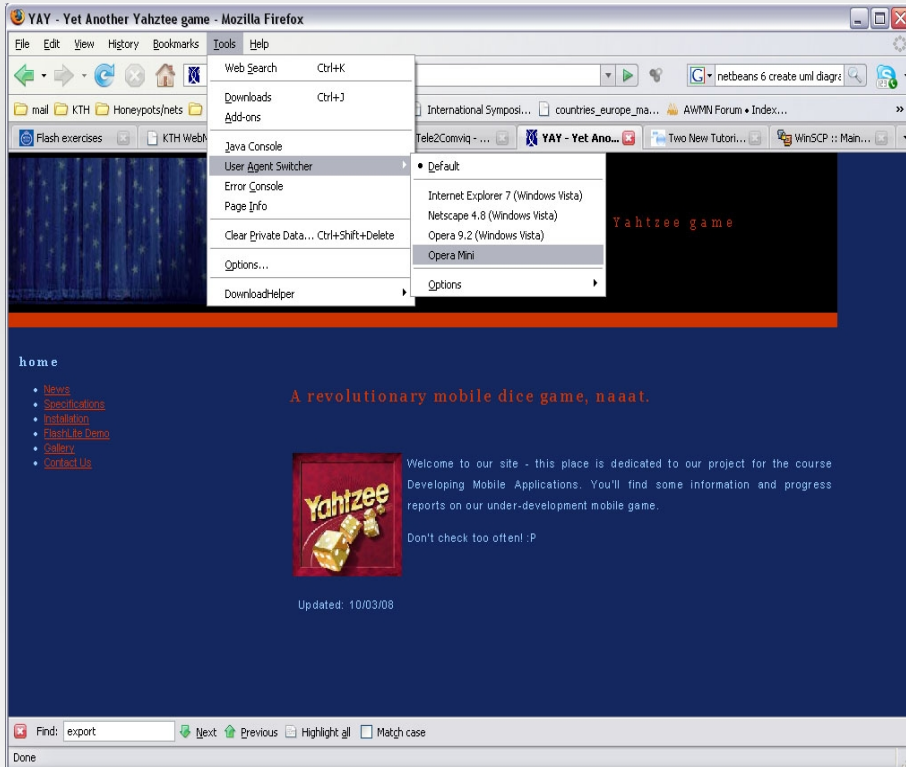
MicroEdition-Profile: MIDP-2.0

# WURFL Content Adaptation

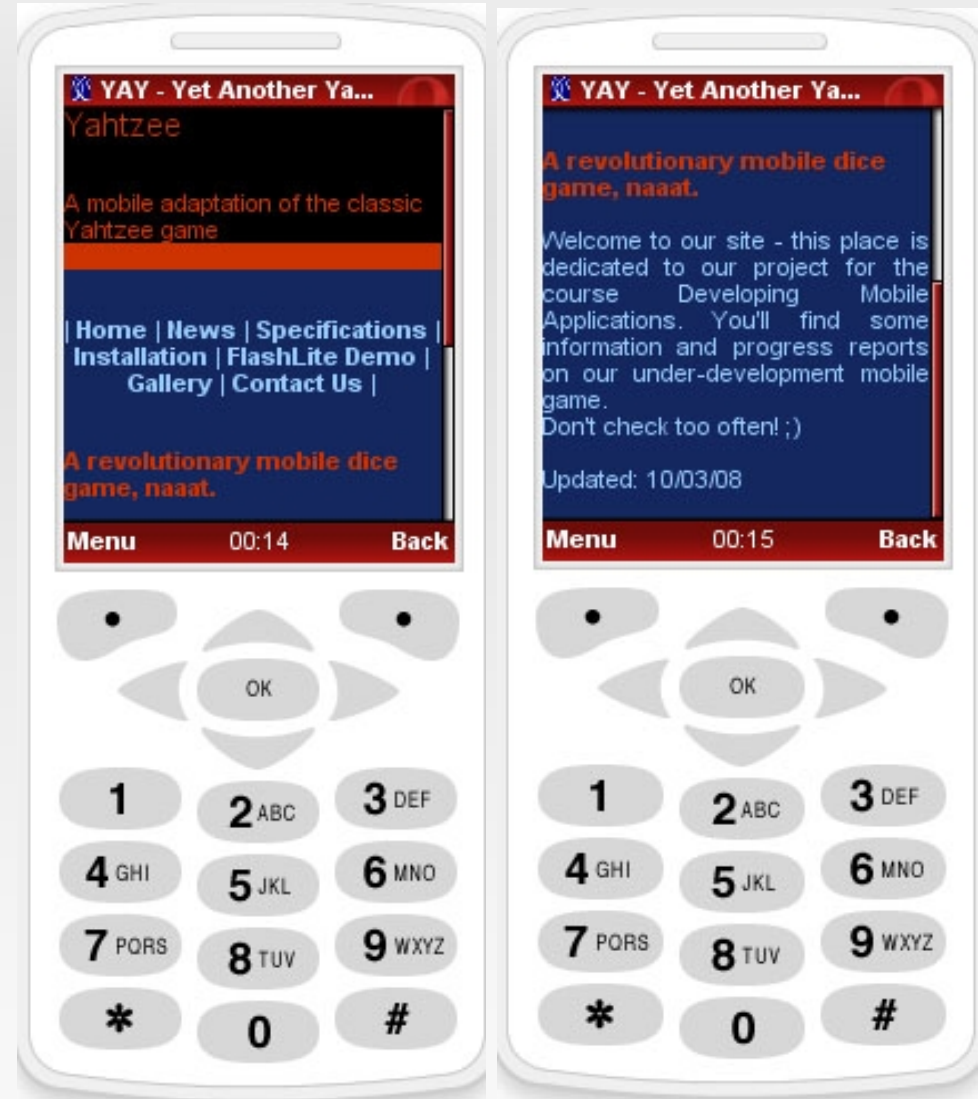
- WURFL class libraries provide User-Agent Detection
- Mobile version of Yahtzee web page, adjusted to mobile device restrictions
- Smaller font, image sizes
- Check if device supports capabilities (e.g., different image formats)



# Wurfl Example



- Browse site using Mozilla's plug in for User-Agent Switch- Get redirected to mobile version of site



- Browse mobile version of site with Opera mini