

Buzzword-Bingo

Manuel Gedack, Kristian Kraljic, Markus Opitz, Pascal Krause,
Timo Kordowski, Marcel Sinn

Hochschule Mannheim

8. April 2011

Agenda

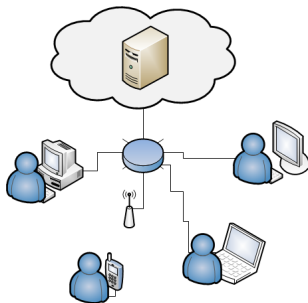
- Aufgabenstellung
- Architektur
- Verwendete Technologien
- Server
 - Implementierung
- Clients
 - Java Client
 - Implementierung
 - iPhone Client
- Probleme
 - Multithreading
 - JSON Bibliotheken

Aufgabenstellung

- Entwicklung einer Buzzword-Bingo Applikation
- Spielen soll möglich sein
- Wordlisten müssen hinzugefügt werden können

Architektur

- Client-Server-Architektur
- Ein zentraler Server stellt Clients Dienste über das Netzwerk bereit
- Clients können auf Wunsch einen Dienst über das Netzwerk beim Server anfordern



Verwendete Technologien

- PHP
- Java
- JavaScript Object Notation (JSON)
- Objective-C

- Apache Web Server
- PHP
- MySQL Datenbank
- Beinhaltet die komplette Logik und Verwaltung
- Stellt Services mit jeweils eigener URL per HTTP bereit
- Stellt den Clients Daten als JSON Objekte bereit

Server: Implementierung

Code Beispiel: Check ob alle Parameter übergeben wurden:

```
//check if all attributes are supplied
if(array_key_exists('playerid',$GET)&&!empty($GET['playerid'])
&&array_key_exists('player',$GET)&&!empty($GET['player'])) {
    $playerid = unsigned_crc32($GET['playerid']);
    $player = $GET['player'];
} else if(array_key_exists('player',$GET)&&!empty($GET['player'])) {
    $playerid = unsigned_crc32_ci($GET['player']);
    $player = $GET['player'];
} else if(array_key_exists('playerid',$GET)&&!empty($GET['playerid'])) {
    $playerid = $GET['playerid'];
    $player = database_single('SELECT `player` FROM `bingo`.`player` WHERE `playerid` = "%s" LIMIT 1;',mysql_real_escape_string($playerid));
    if(!$player)
        json_error('no player with this id was found [attribute \'playerid\']');
} else json_error('player or player id was missing [attribute \'player\', \'playerid\']');
if(!array_key_exists('gameid',$GET)||empty($GET['gameid']))
    json_error('game id was missing [attribute \'gameid\']');
```

Server: Implementierung

Code Beispiel: Check wer gewonnen hat bzw. wie der Status ist:

```
//check for status and/or winner (optional, only if player was not found for a strange reason), or game was cancelled
$json = array_type(array_part($array_merge($json, ($game = database_fetch('SELECT
'gameid', 'title', 'games`.`categoryid`, 'category', 'playerid_master', 'playerid_winner', 'rows', 'columns', 'status', 'mode', 'password', NOT 'password' IS NULL AS
'private' FROM 'bingo`.`games' LEFT OUTER JOIN 'bingo`.`categories' ON 'games`.`categoryid' = 'categories`.`categoryid' WHERE 'gameid' =
"%s":', mysql_real_escape_string($_GET['gameid']))))), array('gameid', 'status', 'title', 'mode', 'category', 'rows', 'columns', 'players', 'private'));
array('gameid'=>TYPE_INTEGER, 'rows'=>TYPE_INTEGER, 'columns'=>TYPE_INTEGER, 'private'=>TYPE_BOOLEAN));
    if(!$game) json_error('game could not be fetched'.mysql_error());
else
if($game['password']&&!array_key_exists('password', $_GET)||empty($_GET['password'])||!salted_md5($_GET['password']).$game['playerid_master']!=$game['password'])
    json_error('password did not match');
else if($game['status']=='wait') json_exit($json);
else if($game['status']=='end')
    if($game['playerid_winner'])
        json_exit(array('status'=>($playerid==$game['playerid_winner'])? 'won': 'lost'));
    else json_error('game has been cancelled');
```


Server: Implementierung

Code Beispiel: Check alle Attribute ob sie valide sind:

```
//check all attributes for validity
if(!preg_match('/\A[A-Za-z0-9][A-Za-z0-9.\- ]{0,63}\Z/',$_GET['player']))
    json_error('player name may only contain A-Z, a-z, 0-9, dots or dashes [attribute \'player\']');
if(!is_numeric($_GET['gameid']))
    json_error('game id must be numeric [attribute \'gameid\']');
if(array_key_exists('password',$_GET)&&!empty($_GET['password']))
    if(!preg_match('/\A[a-fA-F0-9]{32}\Z/',$_GET['password']))
        json_error('provide your password as an encrypted md5 hash [attribute \'password\']');
```

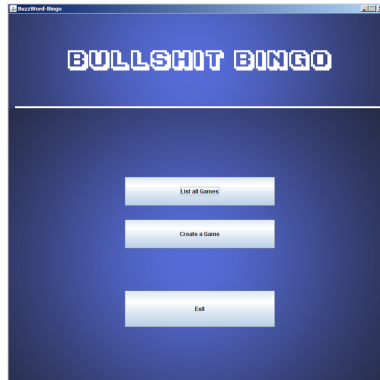
Code Beispiel: Check ob Spieler bereits existiert oder fügt:

```
//check if player exists, if not insert him
$count = database_single('SELECT COUNT(*) FROM `bingo`.`players` WHERE `playerid` = "%s";',mysql_real_escape_string($playerid));
if($count===false) json_error('cant query players database');
else if($count==0)
if(!database_query('INSERT INTO `bingo`.`players` (`playerid`,`player`,`timestamp`)
VALUES ("%s","%s",NOW());',mysql_real_escape_string($playerid),mysql_real_escape_string($player)))
    json_error('cant create player');
```

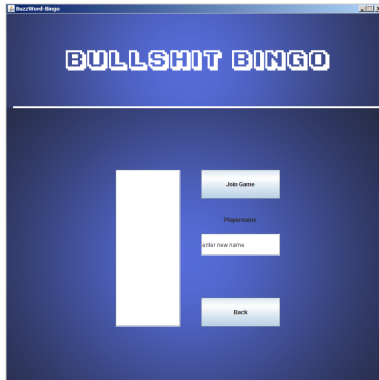
Clients: Java Client

- Swing UI
- Sind “Dumm”, können nur das was der Server als funktionalität bereitstellt
- Keine Datenbank oder ähnliches

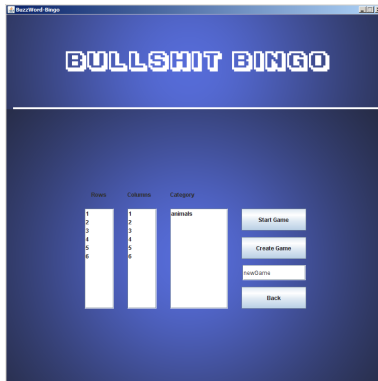
Clients: Java Client



Clients: Java Client



Clients: Java Client



Clients: Java Client



Clients: Java Client Implementierung

Code Beispiel: Client Array Cast

```
private List<String> castJSONArrayToList(JSONArray wordArray) throws JSONException {
    List<String> wordList = new ArrayList<String>();
    for(int i = 0; i < wordArray.length(); i++){
        wordList.add(wordArray.getString(i));
    }
    return wordList;
}

private List<String> getWordList(String objectString) throws JSONException {
    JSONObject wordObject = new JSONObject(objectString);
    JSONArray wordArray = wordObject.getJSONArray("words");
    return castJSONArrayToList(wordArray);
}

public List<String> getWords(String JSONWords) throws JSONException {
    return getWordList(JSONWords);
}
```


Clients: Java Client Implementierung

Code Beispiel: Client Game Start

```
/*
 * (non-Javadoc)
 *
 * @see buzzword.ClientFunctions#startGame()
 */
public boolean startGame() throws NoGameHostException, JSONException,
    ServerCommunicationException, GameEndedException {
    final String urlExtension = "start?gameid=" + gameID + "&player="
        + playerName;
    JSONObject starterMsg = new JSONObject(getJSON(urlExtension));
    if (starterMsg.getString("status").equals("error"))
        throw new NoGameHostException();
    else {
        this.get();
        return true;
    }
}
```

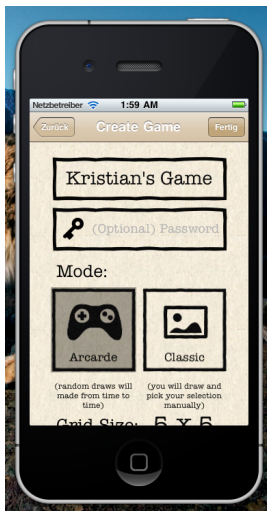
Clients: iPhone Client

- Objective-C
- Klassische iPhone App

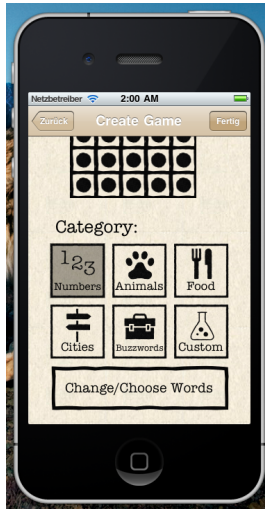
Clients: iPhone Client



Clients: iPhone Client



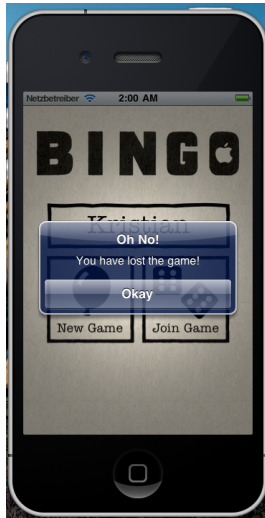
Clients: iPhone Client



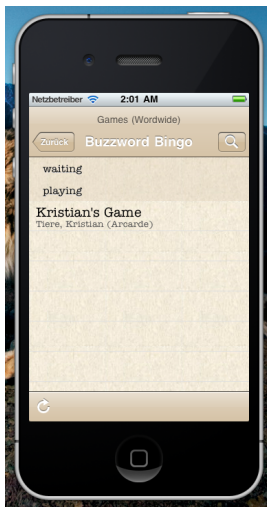
Clients: iPhone Client



Clients: iPhone Client



Clients: iPhone Client



Probleme

- Multithreading beim Java Client
- JSON
 - Schlechte Bibliotheken verfügbar
 - Musten erweitert werden

Ende

Danke für Ihre Aufmerksamkeit!!!

Für Fragen stehen wir gerne zur Verfügung.