Project Title: Heads up Game
Name: Gaury Nagaraju
Openshift URL: http://headsup-67328gauryn.rhcloud.com/

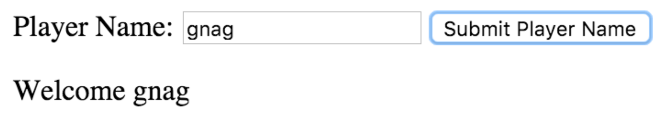
Game Details:

1. Single Player Mode:
    1. Enter Portal
    2. Enter name
    3. See list of categories available
    4. Enter name of category you want to play
    5. Click on Start Game Button.
    6. Game Over when all cards are guessed or 5 minute timer is exceeded
2. Multi-player Mode:
    1. Individual players enter portal and repeat till Steps 4.
    2. All players click Start Game button at the same time
    3. Game Over for all players when one player finishes game

Rubric Requirements:

1. User Interaction:

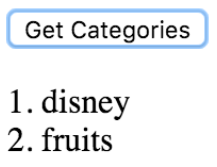
- Enter Player Name:



Player Name: gnag

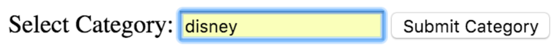
Welcome gnag

- See list of categories available to play:



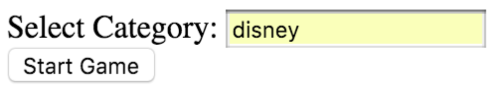
1. disney
2. fruits

- Submit name of category you wish to play:



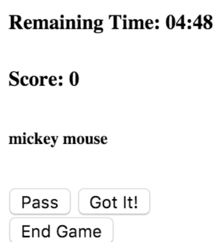
Select Category: disney

- Button to start game so that the timer starts only after they are ready:



Select Category: disney

- Button to end game at any time:



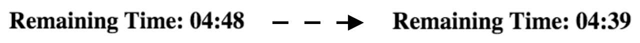
Remaining Time: 04:48

Score: 0

mickey mouse

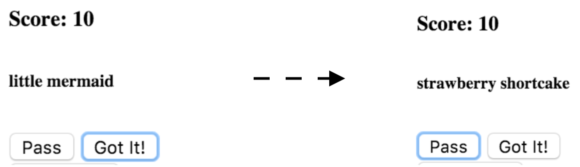
2. Update content displayed on client based on interaction with server:

- Updates to timer is constantly received from server:



Remaining Time: 04:48 → **Remaining Time: 04:39**

- Next card is loaded from server and displayed when player clicks pass/ got it:



Score: 10

little mermaid

Pass | Got It!

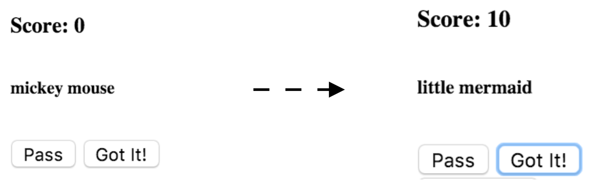
→

Score: 10

strawberry shortcake

Pass | Got It!

- Score is updated based on player's action (+10 if guesses card):



Score: 0

mickey mouse

Pass | Got It!

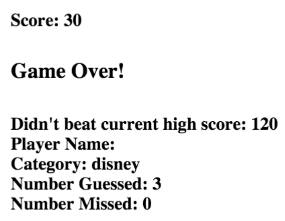
→

Score: 10

little mermaid

Pass | Got It!

- Game Details received from server are displayed at the end of the game:



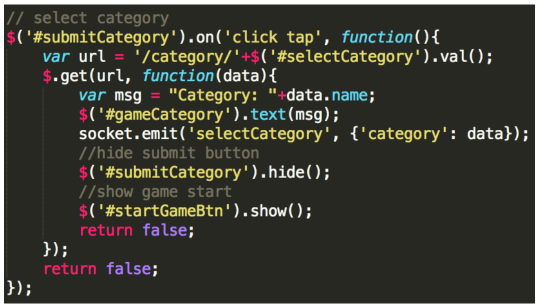
Score: 30

Game Over!

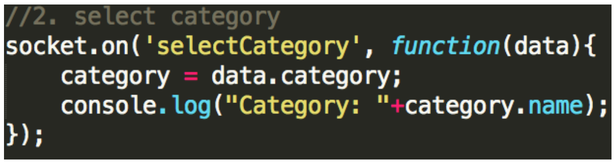
Didn't beat current high score: 120
Player Name:
Category: disney
Number Guessed: 3
Number Missed: 0

3. Use socket.io

Socket io is used for all communication between client and socket.
For e.g. the client emits an event to the server upon category selection:

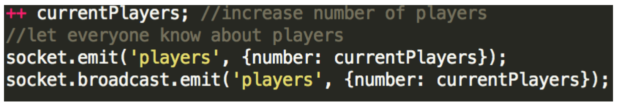


```
// select category
$('#submitCategory').on('click tap', function(){
    var url = '/category/'+$('#selectCategory').val();
    $.get(url, function(data){
        var msg = "Category: "+data.name;
        $('#gameCategory').text(msg);
        socket.emit('selectCategory', {'category': data});
        //hide submit button
        $('#submitCategory').hide();
        //show game start
        $('#startGameBtn').show();
        return false;
    });
    return false;
});
```

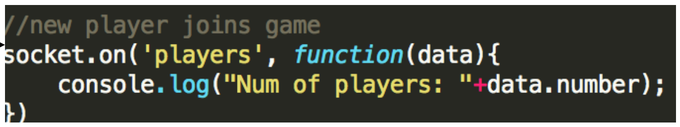


```
//2. select category
socket.on('selectCategory', function(data){
    category = data.category;
    console.log("Category: "+category.name);
});
```

For e.g. the server broadcasts new player arrival to all players:



```
++ currentPlayers; //increase number of players
//let everyone know about players
socket.emit('players', {number: currentPlayers});
socket.broadcast.emit('players', {number: currentPlayers});
```

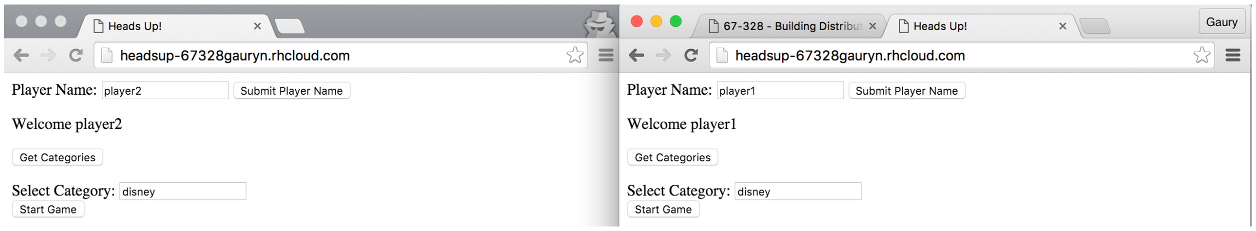


```
//new player joins game
socket.on('players', function(data){
    console.log("Num of players: "+data.number);
})
```

4. Live collaboration with other users:

For multi-player mode: Currently, there is no way to automatically start the game as soon as one player starts, so all players need to click start game at the same time. The game ends when any one player has reached the end. Thus, if player1 goes through all the cards before player2, the game ends for both player1 and player2.

Start Game:


Heads Up!

headsup-67328gauryn.rhcloud.com

Player Name: player2 Submit Player Name

Welcome player2

Get Categories

Select Category: disney

Start Game

67-328 - Building Distribut | Heads Up! | Gaury

headsup-67328gauryn.rhcloud.com

Player Name: player1 Submit Player Name

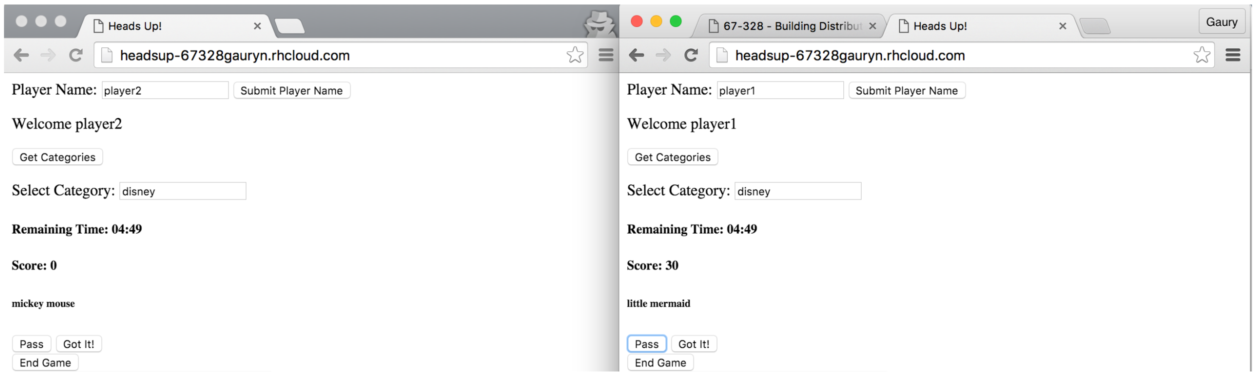
Welcome player1

Get Categories

Select Category: disney

Start Game

Different States of game for both players:


Heads Up!

headsup-67328gauryn.rhcloud.com

Player Name: player2 Submit Player Name

Welcome player2

Get Categories

Select Category: disney

Remaining Time: 04:49

Score: 0

mickey mouse

Pass Got It!

End Game

67-328 - Building Distribut | Heads Up! | Gaury

headsup-67328gauryn.rhcloud.com

Player Name: player1 Submit Player Name

Welcome player1

Get Categories

Select Category: disney

Remaining Time: 04:49

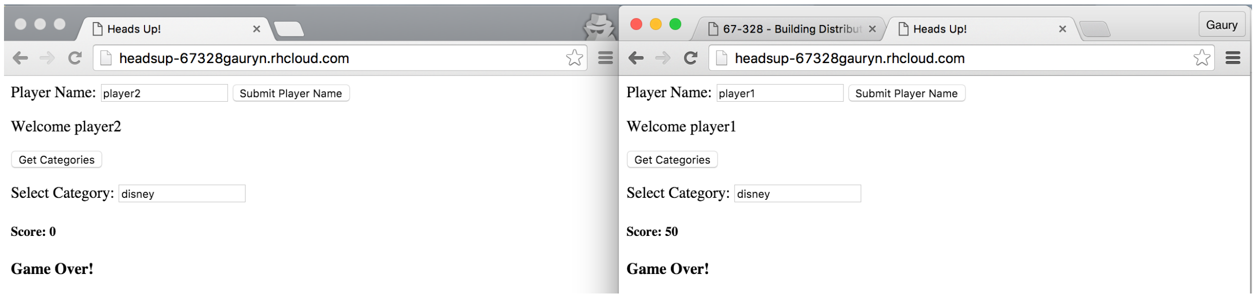
Score: 30

little mermaid

Pass Got It!

End Game

Game Over since player1 finished game:


Heads Up!

headsup-67328gauryn.rhcloud.com

Player Name: player2 Submit Player Name

Welcome player2

Get Categories

Select Category: disney

Score: 0

Game Over!

67-328 - Building Distribut | Heads Up! | Gaury

headsup-67328gauryn.rhcloud.com

Player Name: player1 Submit Player Name

Welcome player1

Get Categories

Select Category: disney

Score: 50

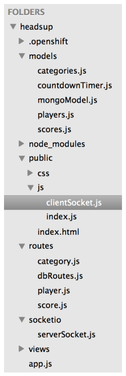
Game Over!

5. <u>Separation of concerns - MVC</u>

Models: category, player, score, mongoModel

Controller:

- Routes - category, player, score, dbRoutes
- Socket - ServerSocket, clientSocket


```
FOLDERS
▼ headsup
  ▶ .openshift
  ▼ models
      categories.js
      countdownTimer.js
      mongoModel.js
      players.js
      scores.js
  ▶ node_modules
  ▼ public
    ▶ css
    ▼ js
        clientSocket.js
        index.js
      index.html
  ▼ routes
      category.js
      dbRoutes.js
      player.js
      score.js
  ▼ socketio
      serverSocket.js
  ▶ views
    app.js
```

Views: Index.html

6. Use sessionStorage: Use session storage to show player's winning games in current session.

At start of game, no wins yet:

Show Winning Games History for current session

No wins yet

After one round:

Show Winning Games History for current session

Category: disney | Score: 20

7. Store data in database: High scores for each category is stored in database.

If there are no high scores for a particular category, it adds the player's score to database.
Browser:

Score: 10

Game Over!

Player Name:
Category: disney
Number Guessed: 1
Number Missed: 2

Console:

```
New High Score!                        clientSocket.js:155
```

If player's score is higher than old top score, update database with high score for that category.
Browser:

Select Category: disney

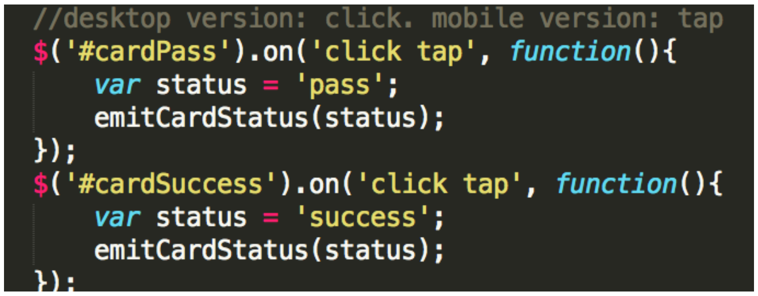
Score: 40

Game Over!

Console:

```
New High Score for Category:           clientSocket.js:169
disneyof 40
```

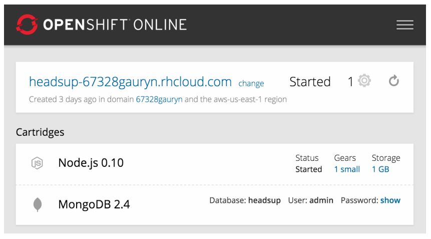
8. Desktop Version: For desktop, one needs to click all the buttons.
9. Mobile Version: For mobile, one needs to tap on the buttons.



```
//desktop version: click. mobile version: tap
$('#cardPass').on('click tap', function(){
    var status = 'pass';
    emitCardStatus(status);
});
$('#cardSuccess').on('click tap', function(){
    var status = 'success';
    emitCardStatus(status);
});
```

10. Deployment to Cloud:

Openshift URL: http://headsup-67328gauryn.rhcloud.com/



OPENSHIFT ONLINE

headsup-67328gauryn.rhcloud.com change   Started   1

Created 3 days ago in domain 67328gauryn and the aws-us-east-1 region

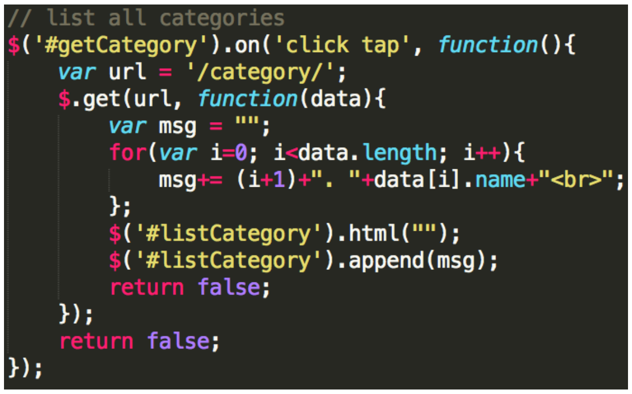
Cartridges

	Status	Gears	Storage
Node.js 0.10	Started	1 small	1 GB

MongoDB 2.4   Database: headsup   User: admin   Password: show

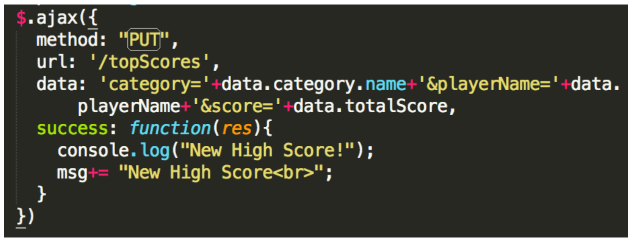
11. Good Separation of HTML, JS, CSS
Public folder has separate folders for html, js, css.
Javascript files are further divided to clientSocket.js and index.js. The former has code for client to communicate with server while the latter has methods that control display such as show(), hide() and sessionStorage.

12. Choice of HTTP methods
GET: get information such as categories available, top Score from database.
E.g.



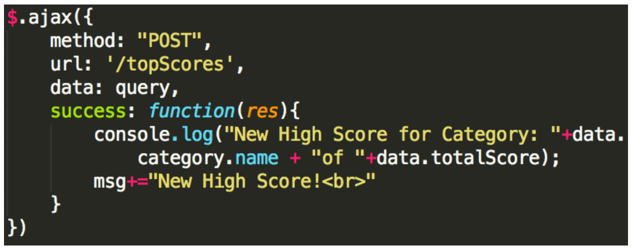
```
// list all categories
$('#getCategory').on('click tap', function(){
    var url = '/category/';
    $.get(url, function(data){
        var msg = "";
        for(var i=0; i<data.length; i++){
            msg+= (i+1)+". "+data[i].name+"<br>";
        };
        $('#listCategory').html("");
        $('#listCategory').append(msg);
        return false;
    });
    return false;
});
```

PUT: create new record for high score in database



```
$.ajax({
  method: "PUT",
  url: '/topScores',
  data: 'category='+data.category.name+'&playerName='+data.
      playerName+'&score='+data.totalScore,
  success: function(res){
    console.log("New High Score!");
    msg+= "New High Score<br>";
  }
})
```

POST: update high score in database, new player name
E.g.



```
$.ajax({
    method: "POST",
    url: '/topScores',
    data: query,
    success: function(res){
        console.log("New High Score for Category: "+data.
            category.name + "of "+data.totalScore);
        msg+="New High Score!<br>"
    }
})
```

13. Commenting Style
The code is commented to explain what each function does and the order in which the game works on both server and client side.

14. Publish to GitHub with license & Readme files:
GitHub Link: https://github.com/gauryn/headsup