



Design and Implementation of Online Experiments



nodeGame.org

Stefano Balietti

MZES and Heidelberg

**nodeGame:
Getting Started**

@balietti
@nodegameorg
stefanobalietti.com@gmail.com

Games/

Let's look inside the installation directory, and let's locate the ultimatum game



Game's Anatomy

Configuration and game files are separated in different folder

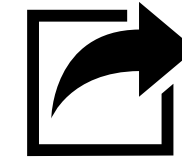
auth
channel
data
game
levels
public
requirements
views
waitroom
.gitignore
package.json

- Define treatments
- Define the sequence of stages and steps of the experiment
- Implement the rules of interactions among participants
- General setup

- Define the graphical contents of the game

- Define the rules about how to assign treatments to game rooms, and participants to treatments

Hands On: Folder waitroom/



[Waiting-Room](#)



Change waiting room options

Start 2 groups simultaneously with random treatment assignment (hint: POOL_SIZE)

 `waitroom.settings.js`

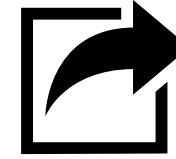
```
// How many clients must connect before groups are formed  
POOL_SIZE: 2,
```

```
// The size of each group  
GROUP_SIZE: 2,
```

A larger pool size allows you to:

- Reshuffle groups after each experiment
- Ensure that randomization is effective (e.g., distribute fast participants across treatments, or other forms of stratified assignment).

Hands On: Folder waitroom/



[Waiting-Room](#)



Change waiting room options

Remove Select Treatments button

 `waitroom.settings.js`

```
// Display buttons to play with bots and select treatment  
ALLOW_PLAY_WITH_BOTS: true,  
ALLOW_SELECT_TREATMENT: false,
```

Hands On: Reading Errors



- Change waiting room options
 - Start 2 groups simultaneously with random treatment assignment (hint: POOL_SIZE),
 - Remove Select Treatments button

 waitroom.settings.js

```
// How many clients must connect before groups are formed  
POOL_SIZE: 2,
```

```
// The size of each group  
GROUP_SIZE: 2,
```

We are modifying an object, and commas separate properties in objects. Remove the comma, and **let's learn how to read errors in Node.JS.**

Hands On: Reading Errors



```
error: GameLoader.buildWaitRoomConf: error reading waitroom.settings file. Game: ultimatum. Err:
C:\Users\balistef\www\nodegame-v5.0.0-dev\games\ultimatum-game\waitroom\waitroom.settings.js:46
```

```
GROUP_SIZE: 2,
AAAAAAAAAAAA
```

```
SyntaxError: Unexpected identifier
```

```
  at Module._compile (internal/modules/cjs/loader.js:721:23)
  at Object.Module._extensions..js (internal/modules/cjs/loader.js:787:10)
  at Module.load (internal/modules/cjs/loader.js:653:32)
  at tryModuleLoad (internal/modules/cjs/loader.js:593:12)
  at Function.Module._load (internal/modules/cjs/loader.js:585:3)
  at Module.require (internal/modules/cjs/loader.js:690:17)
  at require (internal/modules/cjs/helpers.js:25:18)
  at GameLoader.buildWaitRoomConf (C:\Users\balistef\www\nodegame-v5.0.0-dev\node_modules\nodegame-server\lib\GameLoader.js:926:16)
  at GameLoader.loadWaitRoomDir (C:\Users\balistef\www\nodegame-v5.0.0-dev\node_modules\nodegame-server\lib\GameLoader.js:987:17)
  at GameLoader.addGame (C:\Users\balistef\www\nodegame-v5.0.0-dev\node_modules\nodegame-server\lib\GameLoader.js:163:10)
```

Code causing error and
error message

Hands On 4: Reading Errors



```
error: GameLoader.buildWaitRoomConf: error reading waitroom.settings file. Game: ultimatum. Err:  
C:\Users\balistef\www\nodegame-v5.0.0-dev\games\ultimatum-game\waitroom\waitroom.settings.js:46  
  GROUP_SIZE: 2,  
  ^^^^^^^^^^^
```

```
SyntaxError: Unexpected identifier  
    at Module._compile (internal/modules/cjs/loader.js:721:23)  
    at Object.Module._extensions..js (internal/modules/cjs/loader.js:787:10)  
    at Module.load (internal/modules/cjs/loader.js:653:32)  
    at tryModuleLoad (internal/modules/cjs/loader.js:593:12)  
    at Function.Module._load (internal/modules/cjs/loader.js:585:3)  
    at Module.require (internal/modules/cjs/loader.js:690:17)  
    at require (internal/modules/cjs/helpers.js:25:18)  
    at GameLoader.buildWaitRoomConf (C:\Users\balistef\www\nodegame-v5.0.0-dev\node_modules\nodegame-server\lib\GameLoader.js:926:16)  
    at GameLoader.loadWaitRoomDir (C:\Users\balistef\www\nodegame-v5.0.0-dev\node_modules\nodegame-server\lib\GameLoader.js:987:17)  
    at GameLoader.addGame (C:\Users\balistef\www\nodegame-v5.0.0-dev\node_modules\nodegame-server\lib\GameLoader.js:163:10)
```

File name causing the error

Hands On: Reading Errors



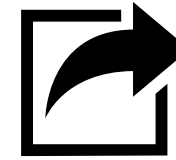
```
error: GameLoader.buildWaitRoomConf: error reading waitroom.settings file. Game: ultimatum. Err.  
C:\Users\balistef\www\nodegame-v5.0.0-dev\games\ultimatum-game\waitroom\waitroom.settings.js:46  
  GROUP_SIZE: 2,  
  ^^^^^^^^^^^
```

Line Number

```
SyntaxError: Unexpected identifier  
    at Module._compile (internal/modules/cjs/loader.js:721:23)  
    at Object.Module._extensions..js (internal/modules/cjs/loader.js:787:10)  
    at Module.load (internal/modules/cjs/loader.js:653:32)  
    at tryModuleLoad (internal/modules/cjs/loader.js:593:12)  
    at Function.Module._load (internal/modules/cjs/loader.js:585:3)  
    at Module.require (internal/modules/cjs/loader.js:690:17)  
    at require (internal/modules/cjs/helpers.js:25:18)  
    at GameLoader.buildWaitRoomConf (C:\Users\balistef\www\nodegame-v5.0.0-dev\node_modules\nodegame-server\lib\GameLoader.js:926:16)  
    at GameLoader.loadWaitRoomDir (C:\Users\balistef\www\nodegame-v5.0.0-dev\node_modules\nodegame-server\lib\GameLoader.js:987:17)  
    at GameLoader.addGame (C:\Users\balistef\www\nodegame-v5.0.0-dev\node_modules\nodegame-server\lib\GameLoader.js:163:10)
```

Note! The line number is not always where the error actually lies. In fact, it is often on a subsequent line. Here, without the comma, the compiler does not even know that a line ended. When you forget a parenthesis, the errored line number can be the last line of the file, which makes it very difficult to find the error's actual position.

Hands On: Reading Errors



[Waiting-Room-v4](#)



```
error: GameLoader.buildWaitRoomConf: error reading waitroom.settings file. Game: ultimatum. Err:
C:\Users\balistef\www\nodegame-v5.0.0-dev\games\ultimatum-game\waitroom\waitroom.settings.js:46
  GROUP_SIZE: 2,
  ^^^^^^^^^^^
```

Stack Trace

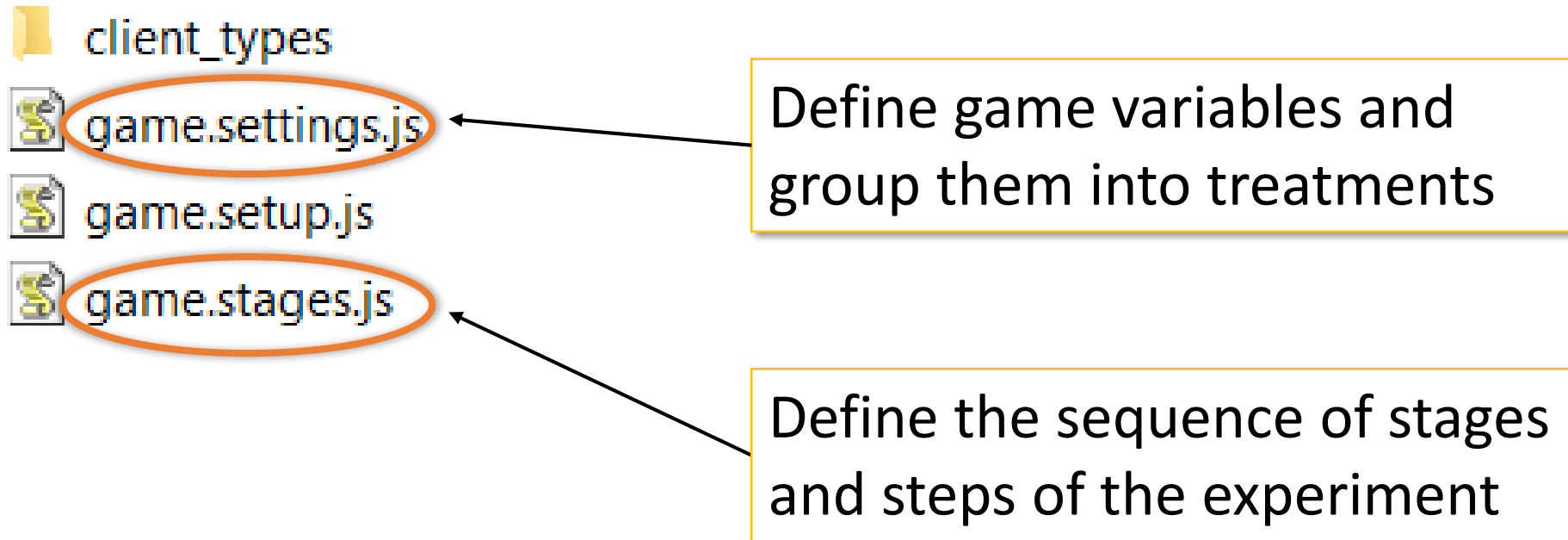
```
SyntaxError: Unexpected identifier
```

```
at Module._compile (internal/modules/cjs/loader.js:721:23)
at Object.Module._extensions..js (internal/modules/cjs/loader.js:787:10)
at Module.load (internal/modules/cjs/loader.js:653:32)
at tryModuleLoad (internal/modules/cjs/loader.js:593:12)
at Function.Module._load (internal/modules/cjs/loader.js:585:3)
at Module.require (internal/modules/cjs/loader.js:690:17)
at require (internal/modules/cjs/helpers.js:25:18)
at GameLoader.buildWaitRoomConf (C:\Users\balistef\www\nodegame-v5.0.0-dev\node_modules\nodegame-server\lib\GameLoader.js:926:16)
at GameLoader.loadWaitRoomDir (C:\Users\balistef\www\nodegame-v5.0.0-dev\node_modules\nodegame-server\lib\GameLoader.js:987:17)
at GameLoader.addGame (C:\Users\balistef\www\nodegame-v5.0.0-dev\node_modules\nodegame-server\lib\GameLoader.js:163:10)
```

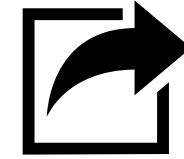
Stack Trace: the stack trace contains all the functions calls until the error happened. *You can generally ignore this part.*

https://en.wikipedia.org/wiki/Stack_trace

Folder game/



Stages Definition



[Stager-API-v4](#)

 `game.stages.js`

We use the stager API to define the sequence (the order matters here!)

A sequence contains stages, and stages contain steps

```
stager
    .next('id_of_stage')

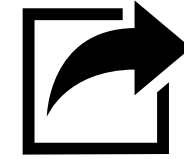
stager
    .step('id_of_step1_within_stage')
    .step('id_of_step2_within_stage')

stager
    .repeat('id_stage_to_repeat', 3)
```

Here, we "chain" two method calls together. We can do it, because each method is returning a `stager` object, so it is a more compact way of writing:

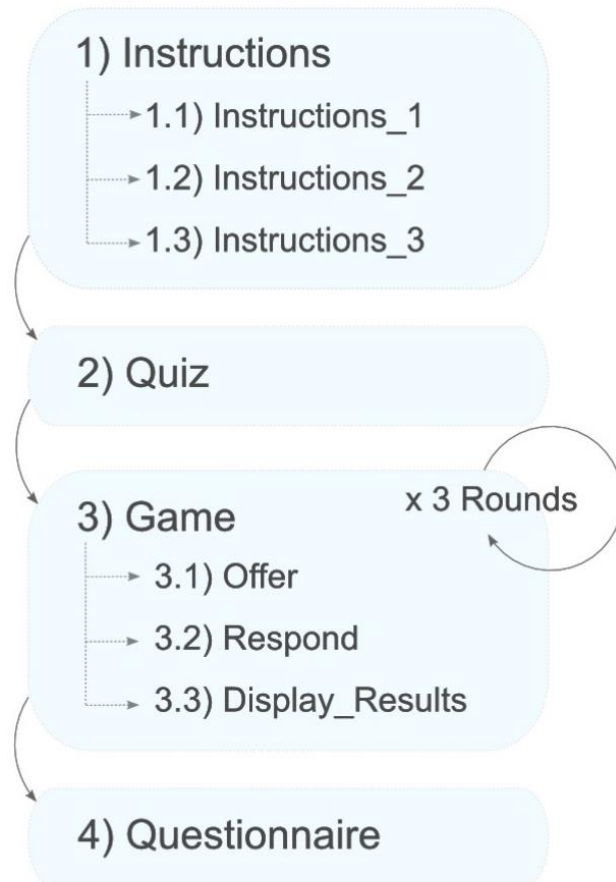
```
stager.step('id_of_step1_within_stage');
stager.step('id_of_step2_within_stage');
```

Stages Definition



[Stager-API-v4](#)

Game Sequence



Code Snippet

```
stager.stage("instructions")
    .step("instructions_1")
    .step("instructions_2")
    .step("instructions_3");

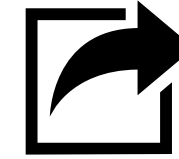
stager.stage("quiz");

stager.repeat("game", 3)

stager.stage("questionnaire");

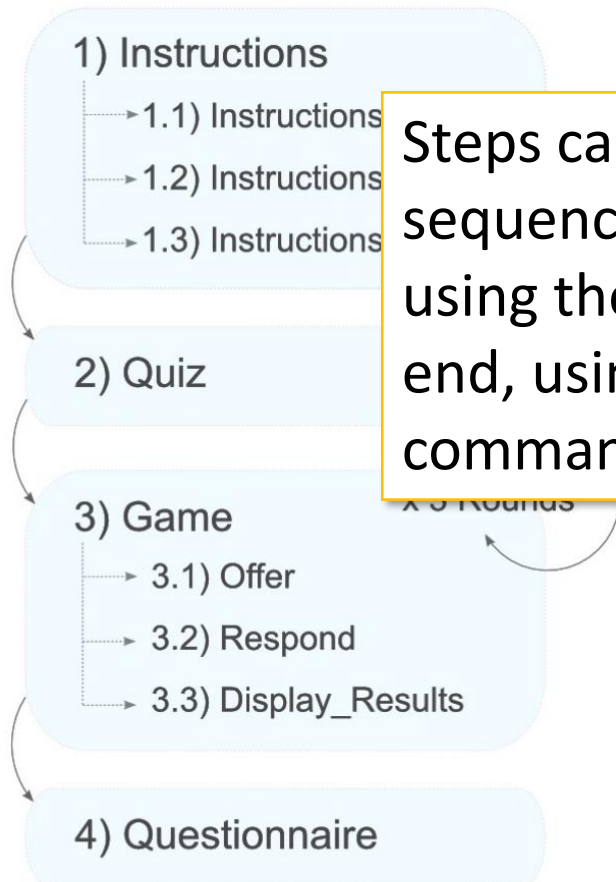
stager.extendStage("game", {
    steps: [ "offer",
            "respond",
            "display_results" ]
});
```

Stages Definition



[Stager-API-v4](#)

Game Sequence



Steps can be added directly in the sequence after a stage command using the **step** command or at the end, using the **extendStage** command

Code Snippet

```
stager.stage("instructions")
    .step("instructions_1")
    .step("instructions_2")
    .step("instructions_3");

stager.stage("quiz");

stager.repeat("game", 3)

stager.stage("questionnaire");

stager.extendStage("game", {
    steps: [ "offer",
            "respond",
            "display_results" ]
});
```

Hands On 4

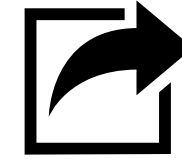


- Skip some stages of the game sequence (very useful for debugging)

 `game.stages.js`

```
// Skip stages from the sequence.  
stager.skip('precache');  
stager.skip('selectLanguage');  
stager.skip('quiz');  
stager.skip('instructions');  
stager.skip('mood');
```

Settings and Treatments



[Settings-and-Treatments-v5](#)

 game.settings.js

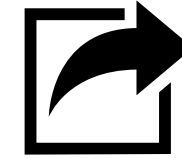
```
/**
 * # Game settings: Ultimatum Game
 * Copyright(c) 2018 Stefano Balietti <ste@nodegame.org>
 * MIT Licensed
 *
 * http://www.nodegame.org
 */

module.exports = {

  // Minimum number of players that must be always connected.
  MIN_PLAYERS: 2,

  // Number of rounds to repeat the bidding. *
  REPEAT: 2,
```

Settings and Treatments



[Settings-and-Treatments-v5](#)

 game.settings.js

Multiline Comments

```
/**  
 * # Game settings: Ultimatum Game  
 * Copyright(c) 2018 Stefano Balietti <ste@nodegame.org>  
 * MIT Licensed  
 *  
 * http://www.nodegame.org  
 */
```

Node.JS makes the content of this object available outside of the file

```
module.exports = {
```

```
  // Minimum number of players that must be always connected.
```

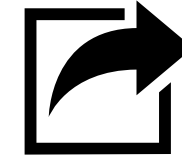
```
  MIN_PLAYERS: 2,
```

Game variables (these two are game-specific)

```
  // Number of rounds to repeat the bidding. *
```

```
  REPEAT: 2,
```

Settings and Treatments



[Settings-and-Treatments-v5](#)

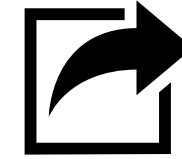
game.settings.js

```
// TIMER.  
// If the name of a key of the TIMER object matches the name of one  
// of the steps or stages, its value is automatically used as the  
// value of the `timer` property of that step/stage.  
//  
// The timer property is read by `node.game.timer` and by VisualTimer  
// widgets, if created. It can be:  
//  
// - a number (in milliseconds),  
// - a function returning the number of milliseconds,  
// - an object containing properties _milliseconds_, and _timeup_  
//   the latter being the name of an event to emit or a function  
//   to execute when the timer expires. If _timeup_ is not set,  
//   property _timeup_ of the game step will be used.
```

```
TIMER: {  
  selectLanguage: 30000,  
  instructions: 90000,  
  quiz: 60000,  
  mood: 60000,  
  questionnaire: 90000,  
  bidder: 30000,  
  respondent: 30000  
},
```

- Variable ***TIMER*** is read by nodeGame's engine
- It defines the max duration (in milliseconds) of each step
- The names of the properties are the ids of the steps in the sequence

Settings and Treatments



[Settings-and-Treatments-v5](#)

 game.settings.js

```
// Available treatments:
```

```
treatments: {
```

```
  standard: {
```

```
    description: "More time to wait and no peer pressure.",
```

```
    WAIT_TIME: 20,
```

```
    instructionsPage: 'instructions.html'
```

```
  },
```

```
  pp: {
```

```
    description:
```

```
      "Introduces peer pressure to players to not disconnect.",
```

```
    WAIT_TIME: 10,
```

```
    instructionsPage: 'instructions_pp.html'
```

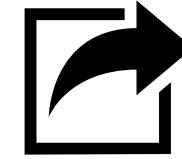
```
  }
```

```
}
```

- Property ***treatments*** is read by nodeGame's engine
- It is an object indexed by the names of the treatments
- Each treatment contains *all the variables defined outside of the treatments object, plus what it is defined inside.*

Treatment names: they will appear in the selection box in the waiting room, together with the description. You are free to choose any name is appropriate for your experiment.

Settings and Treatments



[Settings-and-Treatments-v5](#)

game.settings.js

By assigning a property with the same name, but with different values we can define *controlled differences* in treatments.

```
// Available treatments:  
treatments: {
```

```
  standard: {  
    description: "More time to wait and no peer pressure.",  
    WAIT_TIME: 20,  
    instructionsPage: 'instructions.html'  
  },  
  pp: {  
    description:  
      "Introduces peer pressure to players to not disconnect.",  
    WAIT_TIME: 10,  
    instructionsPage: 'instructions_pp.html'  
  }  
}
```

WAIT_TIME controls how much time to wait for a disconnected player to reconnect. This property is read by nodeGame, which automatically adjusts the reconnect timer.

instructionsPage is a game variable that will use later when extending the steps.

Hands On



- Change the number of repetition of the ultimatum stage

 game.settings.js

```
// Change the number of repetition to 1.  
REPEAT: 1,
```

Hands On



 game.settings.js

```
// Change the number of repetition to 1.  
REPEAT: 1,
```

Reads

 game.stages.js

```
module.exports = function(stager, settings) {
```

```
  stager  
    .next('precache')  
    .next('selectLanguage')  
    .next('instructions')  
    .next('quiz')  
    .next('mood')  
    .repeat('ultimatum', settings.REPEAT)  
    .next('questionnaire')  
    .next('endgame')  
    .gameover();
```