Design and Implementation of Online Behavioral Experiments

nodeGame.org

Stefano Balietti

MZES and Heidelberg

Projects Evaluation

@balietti
@nodegameorg
stefanobalietti.com@gmail.com

Project Evaluation

1. *(Recommended)* Form a group of 3-5 students (size flexible)
2. Pick a topic, **define a research question**
3. **Design and implement** in the nodeGame framework**:**
   a) **an experiment** to answer your research question,
   b) **a software extension** (e.g., a widget) to solve a concrete problem in online behavioral research.
4. **Showcase your work** in class or run it with other participants
5. **Submit your work** together with a short report on **GitHub.com**

Evaluation Criteria

Experiment:

1. Must "**run**" (i.e., no errors)
2. **Experimental workflow**, including instructions, must be suitable for online audience
3. Should **take care of common issues of online experiments** (e.g., validate inputs, waiting room and authorization settings, handle dropouts, etc.)
4. Data collected should answer your research question, **ruling out alternative explanations**

Evaluation Criteria

Experiment:

1. Must "**run**" (i.e., no errors)
2. **Experimental workflow**, including instructions, must be suitable for online audience
3. Should **take care of common issues of online experiments** (e.g., validate inputs, waiting room and authorization settings, handle dropouts, etc.)
4. Data collected should answer your research question, **ruling out alternative explanations**
5. **Bonus.** Quality of code (properly commenting it, properly naming variables, avoiding duplication, etc.)
6. **Bonus.** Originality of research approach

Evaluation Criteria

Software Extension:

1. Must "**run**" (i.e., no errors)
2. How well it **solves the problem** you address (does it generate the right data, does it fail under some circumstances, unambiguous UI, etc.)
3. **Quality of code** (properly commenting it, properly naming variables, avoiding duplication, etc.)
4. **Implementation** choice (re-use of software components, external APIs, etc.)

Evaluation Criteria

Software Extension:

1. Must "**run**" (i.e., no errors)
2. How well it **solves the problem** you address (does it generate the right data, does it fail under some circumstances, unambiguous UI, etc.)
3. **Quality of code** (properly commenting it, properly naming variables, avoiding duplication, etc.)
4. **Implementation** choice (re-use of software components, external APIs, etc.)
5. **Bonus.** Quality of UI or API design.

Projects Evaluation

Your Report

1. **2-5 pages** long
2. Defines **research questions** (or implementation goals)
3. Highlights previous experimental/theoretical **literature** (or existing software)
4. Explains *clearly* what is **your contribution**
5. Discusses **unsolved issues** in your implementation

Projects Evaluation

Your Report

1. **2-5 pages** long
2. Defines **research questions** (or implementation goals)
3. Highlights previous experimental/theoretical **literature** (or existing software)
4. Explains *clearly* what is **your contribution**
5. Discusses **unsolved issues** in your implementation
6. **Bonus.** Could contain analysis of any collected experimental data

Projects Evaluation

Individual Contributions

1. Your report should also specify **individual contributions** to the project (it is fine to say that everybody contributed equally)

Projects Evaluation

Individual Contributions

1. Your report should also specify **individual contributions** to the project (it is fine to say that everybody contributed equally)
2. **Bonus.** Be curious, ask questions and give answers (wrong or correct) in class, online meetings, mailing list, and group chat.

Deadlines

21/10/2020: Define Research Project (and pick a group)

- Nail down your *hypotheses* and *research question*
- Create a *GitHub project repository* (one per group) stating your research question with some preliminary code (*only* the game folder)
- README.md with short summary and references to related literature

Deadlines

21/10/2020: Define Research Project (and pick a group)

- Nail down your *hypotheses* and *research question*
- Create a *GitHub project repository* (one per group) stating your research question with some preliminary code (*only* the game folder)
- README.md with short summary and references to related literature

28/10/2020: Mid-term presentation

- Update Present **current state of work** and get feedback

Deadlines

21/10/2020: Define Research Project (and pick a group)

- Nail down your *hypotheses* and *research question*
- Create a *GitHub project repository* (one per group) stating your research question with some preliminary code (*only* the game folder)
- README.md with short summary and references to related literature

28/10/2020: Mid-term presentation

- Update Present **current state of work** and get feedback

10/01/2021: Final Report

- Submit the final report; code automatically fetched from GitHub

Deadlines

21/10/2020: Define Research Project (and pick a group)

- Nail down your *hypotheses* and *research question*
- Create a *GitHub project repository* (one per group) stating your research question with some preliminary code (*only* the game folder)
- README.md with short summary and references to related literature

28/10/2020: Mid-term presentation

- Update Present **current state of work** and get feedback

10/01/2021: Final Report

- Submit the final report; code automatically fetched from GitHub

11-25/01/2021: Final Presentation

- Proudly present your work and engage in Q&A (doodle to choose the time and date)