

---

# BALLOON ANALOG RISK TASK (BART) - ONLINE VERSION DOCUMENTATION

September 11, 2024

## 1 Running the Task

Follow these steps to run the task locally:

1. Install Visual Studio Code (VS Code).
2. Open the project folder in your terminal and run the following command: **code .**
3. In VS Code, open a terminal and run a Python server using the command:  
**python -m http.server 8000**  
(You can replace 8000 with any available port number.)
4. Open your browser and go to `http://localhost:8000/`.
5. Navigate to the `task` folder.
6. Open `demo.html`.
7. The task will now be running.
8. When you update the task in VS Code, close the previous server by pressing **Ctrl+C** (or **Cmd+C** on Mac), then start a new server with a new port, e.g.:  
**python -m http.server 8080**  
This will reflect the changes you've made.

## 2 Task Sections

The task is divided into the following sections:

1. **Consent Form** – User agreement.
2. **Demographic Form** – Collected data is downloaded in CSV format.
3. **Task Instructions** – Instructions for performing the task.
4. **BART Task** – Main task, with data saved in CSV format.
5. **Questionnaire** – Questionnaire data is also saved in CSV format.
6. At the end of each section, three files are downloaded with the respective section name and date-time stamp.

## 3 Files in the Project

The task is designed using JSpsych. The project contains 7 main files. Here is an explanation of each:

### 3.1 demo.html

Includes all the plug-ins from the `jspsych` folder, and links to `style.css` and `script.js`.

---

## 3.2 style.css

Controls the appearance of the task interface.

## 3.3 script.js

The core script for the task. This file runs all JavaScript functions for the task, including the `runAllTasks()` function, which orchestrates the entire process: first the description, then the task, followed by the questionnaire.

## 3.4 task\_description.js

Contains the task description, consent form, and demographic questions. Once completed, the demographic data is downloaded.

## 3.5 task.js

This file contains the core logic for the BART task. It contains some important parts that I'm going to explain:

### 3.5.1 Experiment Initialization:

The experiment begins with the initialization of jsPsych, where parameters such as the width of the experiment window are specified. The `on_finish` function signals the conclusion of the experiment and resolves the main promise.

### 3.5.2 Audio Files and Variables:

Two audio files, `popSound` and `bankSound`, are preloaded to provide feedback during the experiment. Variables like `trialData` for storing trial data, `TrialNum` for the total number of trials, and parameters related to balloon size (e.g., `BalloonSizeStep`) are initialized.

### 3.5.3 Balloon Characteristics:

The balloon colors and sizes are defined using mean and standard deviation values for each color. The function `getGaussianRandom()` calculates random balloon sizes using the Gaussian distribution for realistic variability in balloon inflation.

### 3.5.4 Trial Construction:

The code constructs two types of trials:

1. normal trials (colored balloons like red, orange, yellow), which are 75% of the trials. These are main balloons that participant collects rewards by starting and stopping the inflation.
2. special trials (either gray balloons or colorful special balloons) which are 25% of the trials. Gray balloons are passive trials with no reward. The participant just needs to inflate the balloon and watch it inflate that's it. Special balloons are yellow, orange, and red balloons with a gray circle around them showing the max radius and the balloon inflates to that size and then the participant gets reward for that trial just by only inflating the balloon and no need to bank.

---

### 3.5.5 Trial Execution:

Each trial is presented using HTML stimuli. The balloon inflation mechanics are tied to buttons, with the `inflate` button increasing balloon size and potentially rewarding the participant, while the `bank` button banks the reward for the trial. For each balloon, data such as reaction time, inflation time, and outcomes (whether the balloon popped or reward was banked) are recorded.

### 3.5.6 Outcome Handling:

When a balloon pops or a reward is banked, the outcome is visually represented (e.g., a "popped" or "banked" message) along with auditory feedback from the `popSound` or `bankSound`. Data related to the reaction time, inflation time, and reward are stored.

### 3.5.7 Data Export:

Upon the completion of the task, trial data is saved as a CSV file. The data-saving function `saveCSV()` prepares the file with the appropriate headers and content before triggering a download for the user.

### 3.5.8 Task Finalization:

The final trial instructs participants to press 'C' to continue, at which point the data is saved, and the experiment concludes.

## 3.6 task\_questionnaire.js

Contains the impulsivity measure questionnaire. Data is downloaded when the participant finishes this section. The questionnaire is based on the paper found [here](#).

## 3.7 task\_balloon.js

This file was intended to use real balloon images for the task. Currently, circles are used instead. This file is not active and can be ignored.

## 4 Project Folders

There are 4 main folders in the project:

### 4.1 favicon\_io

Contains the small icon displayed on the browser tab.

### 4.2 img

Contains images used in the task. Real balloon images are present but not used in the current version. `task_balloon.js` refers to these images but uses circles instead.

---

### 4.3 jspsych

Contains all the plug-ins and packages used by JSpsych.

### 4.4 sound

Contains the sound effects used during the task. The inflate sound is not used, as it was deemed distracting.

## 5 Data Folder

The `data` folder consists of two main parts:

### 5.1 1\_analysis.ipynb

A Jupyter notebook for sample data analysis. It processes data from the following sources:

- **demographic\_data.csv:** Contains demographic information such as gender, age, ethnicity, and race.
- **task\_data.csv:** Contains data from 250 task trials, with the following columns:
  - **balloonType:** Categorized into yellow, orange, red, special yellow, special orange, special red, and gray balloons.
  - **outcome:** Indicates if the balloon popped or was banked (gray balloons are marked as popped, and special balloons are marked as banked).
  - **reactionTime (ms):** The time between balloon onset and the inflate button press.
  - **inflationTime (ms):** The time between pressing the inflate button and the outcome.
  - **reward:** Cumulative reward amount.

### 5.2 1\_nill folder

Contains three CSV files downloaded during the task: `task_data.csv`, `demographic_data.csv`, and `questionnaire_data.csv`. These files were briefly reviewed in the `1_analysis.ipynb` notebook.