

Pointer data and deployment

Meeting 5 | A complete pilot, a verified save path, and realistic extensions

Bartosz Maćkiewicz

Programming Behavioral Experiments in jsPsych

Record interpretable pointer events and distinguish a completed task from a confirmed saved result.

Artifact: a mouse-tracking pilot and a project release checklist.

Keep the advanced feature small

We add one new measure to a familiar decision task.

Mouse tracking uses a supplied extension. Webcam gaze and community extensions are optional discussion.

Final projects do not need an advanced measure for full marks.

What is the participant doing?

1. Click the center Start button.
2. Read a word.
3. Move to Living or Nonliving and click.
4. Return to the same start location for the next word.

A fixed starting action makes trajectories more comparable.

Register the extension globally

```
const jsPsych = initJsPsych({  
  extensions: [{  
    type: jsPsychExtensionMouseTracking,  
    params: { minimum_sample_time: 16 }  
  }]  
});
```

16 ms is a minimum interval, not a guaranteed sampling rate.

Attach tracking to the decision trial

```
extensions: [{  
  type: jsPsychExtensionMouseTracking,  
  params: {  
    targets: ["#left-btn", "#right-btn"],  
    events: ["mousemove", "mousedown", "mouseup"]  
  }  
}]
```

Each selector should identify one element.

Button labels and response indices

```
button_html: (choice, index) => {  
  const id = index === 0 ? "left-btn" : "right-btn";  
  return `    ${choice}</button>`;  
}
```

`button_html` builds each button. The saved response is its zero-based index; IDs let the tracking extension find the targets.

Add the global extension and trial-level parameters. Use the supplied button IDs.

Move the pointer before clicking. Finish and download JSON.

`mouse_tracking_data` contains event records:

- `x, y`: viewport coordinates in pixels;
- `t`: time since the tracked trial started;
- `event`: the recorded event type.

`mouse_tracking_targets` contains element rectangles.

`x` and `y` are viewport coordinates, and `t` is elapsed time within the tracked trial. A 16 ms minimum interval does not guarantee exactly 62.5 samples per second.

Live checkpoint L2: make it interpretable

Find the decision row, not the start-click row.

Check events, time order, and the two target rectangles. Explain why an empty trajectory differs from a missing field.

JSON preserves nested arrays more naturally than a flat table.

What this does not measure directly

A pointer path is not a gaze trace or a direct reading of thought.

Control the start position, record target geometry, and consider the device.

Completion is a screen state. Saving is an operation. Retrieval is evidence.

A static file server does not become a results database.

The supplied completion panel offers CSV and JSON.

Download, reopen, and check the final trial. The files are on the current computer, not on a server.

Use this route when no classroom runner is available.

```
jatos.onLoad(() => {  
  jsPsych.run(timeline);  
});
```

Use the supplied JATOS entry page through the runner. The local entry page remains suitable for development.

```
on_finish: () => {  
  jatos.startNextComponent(jsPsych.data.get().json());  
}
```

This sends the results and advances to the next component, or finishes the study. Let the platform manage the transfer.

Cognition.run: complete the task and open its saved results. JATOS: run the prepared component and retrieve its result.

Check the final trial and the expected task-row count. Local download is the fallback, not an upload to a platform.

Cognition.run is a separate integration

Its documented workflow handles completion and result storage.

Follow its current platform instructions, check package versions, and retrieve a pilot. Do not paste JATOS-specific code into it.

No paid account is needed for the local course activities.

1. Enable tracking on four decision trials.
2. Verify zero-based choice scoring and item labels.
3. Preserve the common start action and geometry.
4. Save and inspect one complete result.

Then spend the protected clinic time on your final project.

Camera initialization → calibration → validation → task.

A coarse area-of-interest question and letter-level gaze timing make very different demands on a measurement system.

Use the observation worksheet; no camera is required for credit.

Ask about compatibility, permissions, data fields, units, maintenance, licensing, and a fallback.

A demo that runs once does not establish measurement validity. Use the ecosystem worksheet for one concrete example.

Can another person:

- run the extracted package;
- explain one row and each condition;
- reproduce a failure test;
- retrieve the data;
- identify the prototype's limits?

Explain the difference between a local download, a submitted request, and confirmed server storage.

Final project: 14 days after this meeting. Use the rubric, pilot log, and README template before submitting.