🧠 Brain–Behavior Mapping: A Neuro-Inspired Simulation Framework

> **Track 3 – Cognitive & Behavioral Analyses**
> Decode neural intent, model dopamine-driven learning, and build an integrated brain-inspired agent.

This project simulates key human brain mechanisms—motor intention decoding and reward-based learning—to create a closed-loop system where "thoughts" drive actions, and behavior adapts through simulated dopamine signals (Reward Prediction Error, RPE).

🎯 Vision

Build a minimal yet biologically plausible **brain–behavior pipeline** that:

- Translates imagined movements (EEG-like signals) into discrete actions.
- Models dopamine-driven learning via Reinforcement Learning (Actor-Critic + RPE).
- Adapts dynamically to changing environments (e.g., reversal learning).
- (Optional) Explains decisions using XAI for transparency.

Final demo: a **mind-controlled gambling task** where the user "thinks left/right" to choose options, and the agent learns optimal strategies through simulated neural plasticity.

👥 Team & Task Distribution (3 Members)

Personne 1 – Neurosignal Processing & BCI

Focus: From brain signals to intention
Responsibilities:

- Load and preprocess EEG datasets using `MOABB`.
- Implement and train a motor imagery classifier (e.g., `EEGNet`, `ShallowConvNet`).
- Simulate realistic EEG if no real hardware is available (`mne`, `scipy`).
- Map decoded intention (`left`/`right`) to a standardized action format.
- Validate accuracy (>70% on BNCI2014001 or similar).

Deliverables:

- `models/bci_decoder.py`
- Trained model weights + evaluation notebook
- Real-time (or simulated) prediction script

Personne 2 – Reinforcement Learning & Cognitive Modeling

Focus: Simulating dopamine and adaptive decision-making
Responsibilities:

- Design a **custom gambling task environment** (2-armed bandit with changing probabilities).
- Implement a **manual Actor-Critic agent** with explicit **Reward Prediction Error (RPE = δ)**.
- Visualize RPE dynamics (phasic dopamine response, extinction, reversal learning).
- Ensure the agent adapts when reward probabilities shift.

Deliverables:

- `envs/gambling_task.py` (Gymnasium-compatible)
- `models/rpe_agent.py`
- Analysis notebook: RPE curves, policy evolution, behavioral adaptation

---

Personne 3 – Integration, UX & Explainability

Focus: Unifying modules + interpretability + user experience
Responsibilities:

- Integrate BCI output (Personne 1) with RL environment (Personne 2).
- Build a simple real-time interface (e.g., `pygame` or `streamlit`) showing:
  - User's "thought" (left/right)
  - Agent's RPE and current policy
  - Reward history
- (Optional) Apply **XAI (SHAP / Integrated Gradients)** to the BCI model to highlight important EEG channels.
- Write documentation, manage Git workflow, and create demo video.

Deliverables:

- `demos/run_simulation.py`
- `utils/viz.py` (RPE + EEG topomaps)
- Final demo video (1–2 min)
- This `README.md` and project report

🛠️ Tech Stack

Component	Tools & Libraries
EEG Processing	`moabb`, `mne`, `scipy`
Deep Learning	`PyTorch`, `torchvision`
Reinforcement Learning	Custom Actor-Critic (NumPy/PyTorch)

Component	Tools & Libraries
Environment	`gymnasium` (custom env)
Visualization	`matplotlib`, `seaborn`, `plotly`, `mne.viz`
XAI (optional)	`shap`, `captum`
Demo / UI	`pygame`, `streamlit`, or `opencv`
Language	Python 3.10+