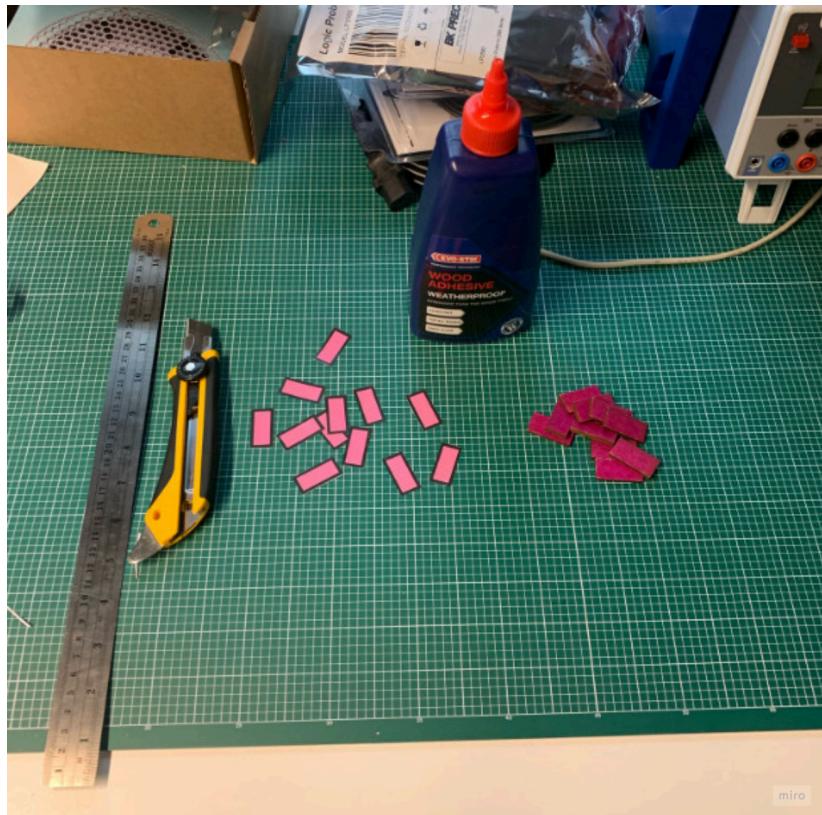


Setup for Solar Sharing Hybrid Physicalisation:

Token Making

1. Using the recommended token print out pdf ([game_code/set_up_docs/game_setup/Player_Tokens.pdf](#)) cut each token including its black border.
2. The paper tokens are 3cm x 1.5cm and can be stuck onto corresponding graspable tokens.
 - a. We are using 3cm x 1.5cm MDF or plywood tokens, (premade using laser cutter).



N.B: It's important for the paper to be stuck on as evenly as possible and as flat as possible - as the corners of the tokens will wear/lift up over time - this may affect object detection.

Board and Camera Setup

1. Set up the board and camera as shown in the image series. The board can be printed from the pdf linked to ([game_code/set_up_docs/game_setup/Battery_Board.pdf](#))

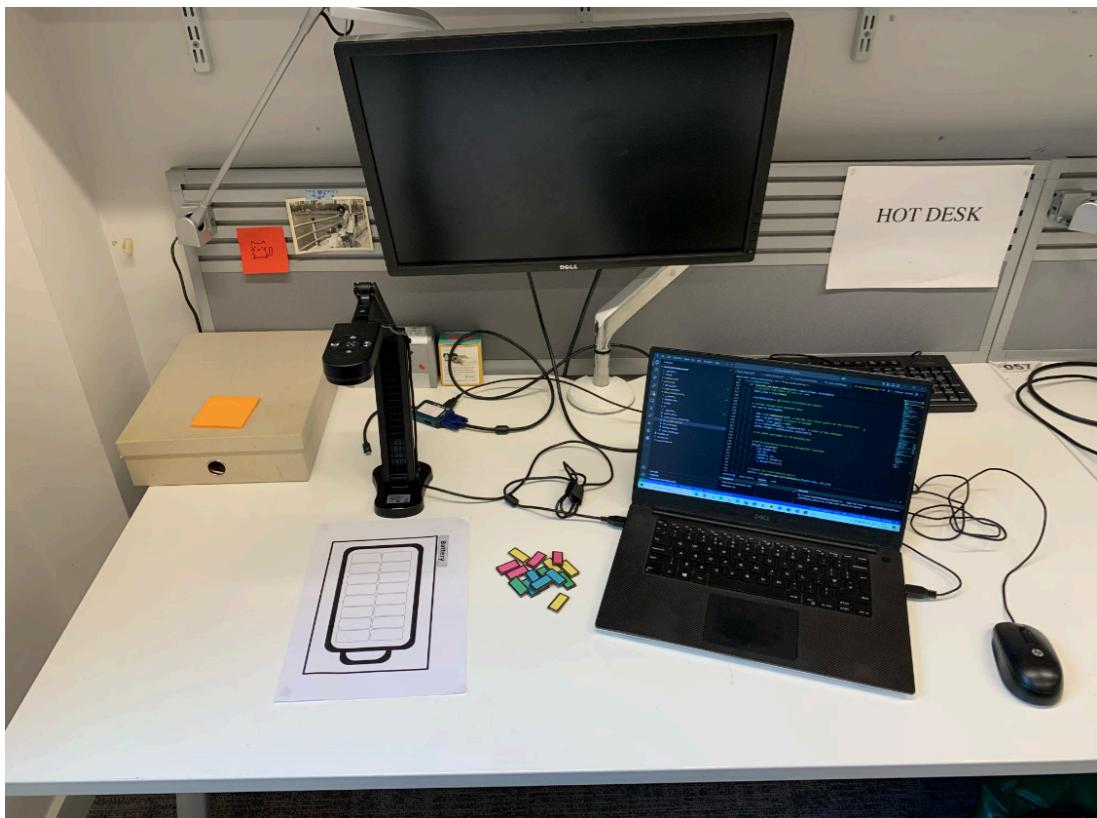


Before Setup:

- Pakoo Camera
- Tokens
- Board Print Out
- USB input port to PC

N.B: Tips for more reliable colour/object detection:

- Place the board and camera in a well and evenly lit location - ample natural light is ideal for this.
 - Shadows that fall unevenly across the board are not recommended.
 - Dark settings with one source of light are also not recommended (this creates shadows and illuminates unevenly)
- Avoid having wires cross the field of view.
- Avoid placing the camera where it might be easily knocked.



Setup:

- Unfold Pakotoo and plug into PC's USB port - from table to camera should be ~40cm vertically.
- Turn on the Pakotoo's ring LED (click on the thumbprint 3 times to reach the brightest level)
- Adjust the Pakotoo's exposure - click on the "increase exposure button 5 times"
- Position board aligned vertically under the camera - blue-tac the board to the table, or other.

Labelling New Training Data

1. To make new labelled datasets for coloured tokens - make use of png files (*game_code/set_up_docs/training_tokens*). Print one out of each colour.
2. Setup camera and board as detailed above.
3. Open the script: *token_labelling.py* → (*game_code/token_model_training/src/token_labelling.py*)
4. For each colour that you want to label the script is run separately. Input the colour name to the code.
5. Remove any items of colour from the board whilst the white patch calibration is completed. Click 'q' to progress the code when the white patch balanced frames are shown.
6. You then have 10 seconds to position the printouts that will be segmented and divided into data folders.
7. Check in the *token_model_training/labelled_data_tokens/"colour"_tokens* folder to see if images have been appended.

Extracting Features

1. Run *colour_feature_extractor.py* → ensure the correct *labelled_data* folder is linked.
2. This will create a CSV file in *tokens/CSV_files*.

k-NN Model Use

1. Run, *knn_training.py* → step through the graphs that display performance by clicking x on the pop up windows.
2. Check to see if a new model and its scaler have been saved to the "models" folder
3. When classifying in the detection code - link the desired model + scaler to the code. It can be possible to try several different models as inputs and see which work better etc.