SD5913 / WEEK 08 / 29 OCTOBER 2026

Build together. Ship something.

Teamwork, cloud environments, delivery and computer vision.

POLYU DES°GN

What is your group idea becoming?

- **A** A game
- **B** An app or creative tool
- **C** An installation
- **D** Something else / still deciding

POLYU DES°GN

Today

01 Project, assessment and news

02 Agile, Scrum and GitHub tracking

03 Git workflow + Codex environments

04 10-minute break

05 Actions, builds and deployment secrets

06 Live camera: OpenCV + MediaPipe

07 PFAD: the two-hour tutorial

POLYU DES°GN

Who gets to build and remix?

Switch 2: new access enables homebrew and mods.

Indie games: fear of AI clones delays a demo.

Agent research: automated monitors can be bypassed.

OpenAI announcement: test what coding agents produce.

POLYU DES°GN

Dessa can listen while it speaks.

Dessa / Block: a research preview of speech-to-speech AI.

Full duplex: it listens and speaks at the same time.

Tone, timing, laughter and interruptions shape the conversation.

Your game: what changes if a character can hear you mid-sentence?

01

Your group project

AN AMBITIOUS IDEA, BUILT AND DELIVERED TOGETHER.

POLYU DES·GN

Register your repository now.

1 Join sd5913; create one public repository for your team.

2 Submit its URL on Canvas; an initial README is enough.

3 Keep building and documenting progress in that repository.

Sunday 6 December 2026, 23:59 HKT

Read the group-project brief →

How the project is marked

25%

Creative idea

Novelty
and value

50%

Code

Works, meets the brief
and handles complexity

25%

Execution

Steady work;
weekly progress

These are shares of the group-project mark.

Make the next step testable.

Who uses it, and what do they do?

What visible or audible response makes the idea special?

What is the smallest working interaction your team can test?

Agree the scope together; credit code, assets and tools.

02

Work as a team

PLAN TOGETHER. BUILD, REVIEW AND ADAPT.

POLYU DES°GN

Three different kinds of support

Agile: values and principles for adaptive software development.

Scrum: a framework for learning and delivering value in Sprints.

GitHub Projects: a tool to track issues, ownership and progress.

Use them to organise the next playable or usable part of your project.

POLYU DES°GN

The Agile Manifesto: four values

Individuals and interactions	over	processes and tools
Working software	over	comprehensive documentation
Customer collaboration	over	contract negotiation
Responding to change	over	following a plan

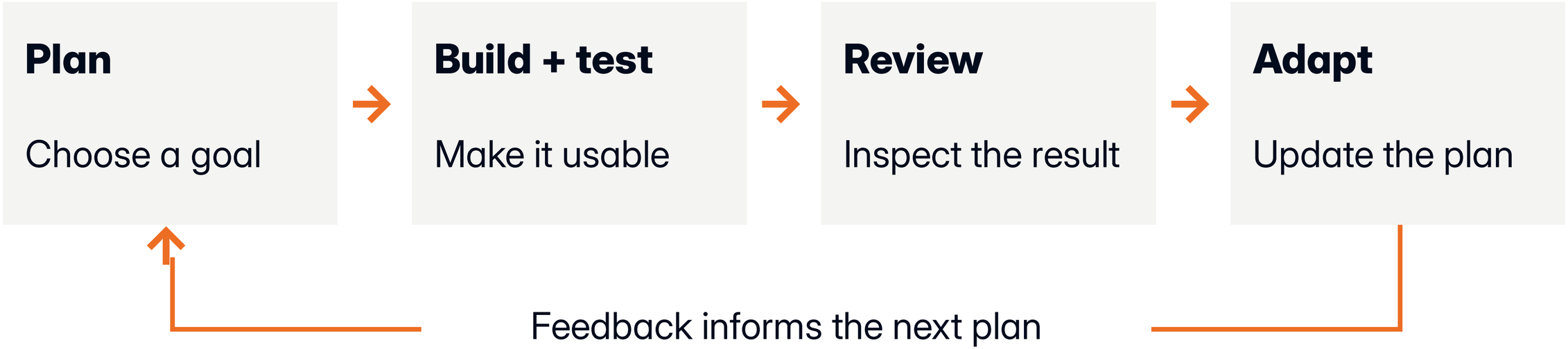
Both sides have value; the items on the left are valued more.

Read the Manifesto and its principles

POLYU DES°GN

Deliver, learn, adapt.

Iterative: improve with feedback. Incremental: add usable pieces.



Transparency → Inspection → Adaptation

POLYU DES°GN

Who is accountable?

Product Owner

Maximise value

Order the backlog

Developers

Plan, build and test

Create a usable Increment

Scrum Master

Help the team improve

Address impediments

One team shares responsibility for a useful result.

From a goal to a usable Increment

Artifact	What it contains	Commitment
Product Backlog	Ordered work to improve the product	Product Goal
Sprint Backlog	Goal + selected work + delivery plan	Sprint Goal
Increment	Usable, integrated, verified work	Definition of Done

POLYU DES°GN

A Sprint contains the learning cycle.

SPRINT / fixed length / one month or less

Planning	→	Daily Scrum	→	Review	→	Retrospective
Agree a goal; select work		15 min daily Inspect + adapt		Inspect result with stakeholders		Improve quality and teamwork

Review the product. Reflect on the process. Adapt the next Sprint.

POLYU DES°GN

Make the work visible in GitHub.

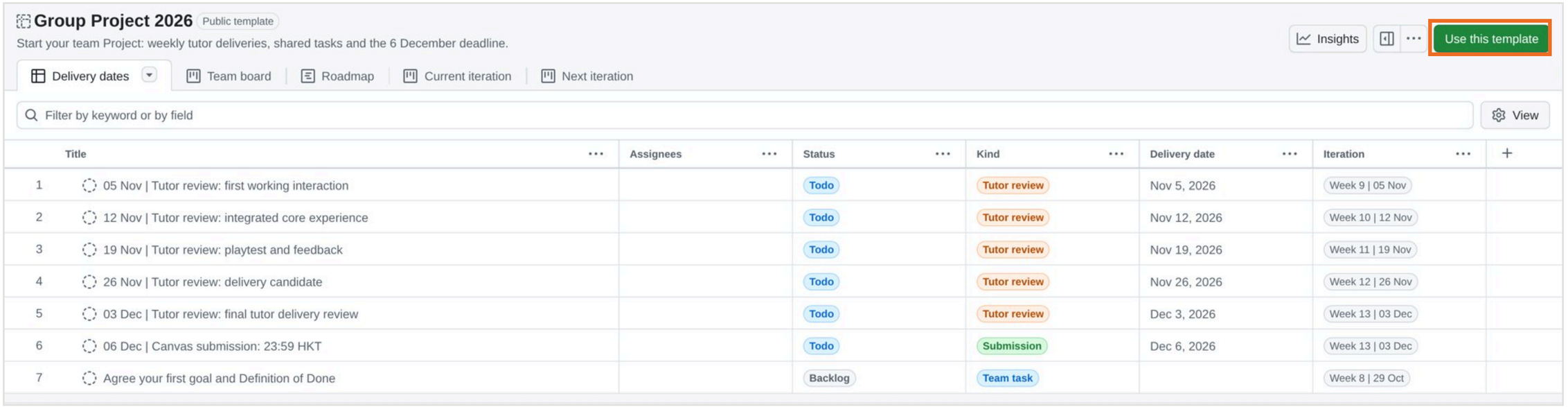
Backlog	Todo	In progress	Review	Done
#15 Pause menu Alice	#12 Start button Bob	#13 Collect a star Alice	#14 Restart level Bob	#11 Player moves Alice

Track Status + Assignee; add Priority and Iteration if useful.

GitHub Projects setup guide →

Start from the course template.

Start here ↓



	Title	Assignees	Status	Kind	Delivery date	Iteration
1	05 Nov | Tutor review: first working interaction		Todo	Tutor review	Nov 5, 2026	Week 9 | 05 Nov
2	12 Nov | Tutor review: integrated core experience		Todo	Tutor review	Nov 12, 2026	Week 10 | 12 Nov
3	19 Nov | Tutor review: playtest and feedback		Todo	Tutor review	Nov 19, 2026	Week 11 | 19 Nov
4	26 Nov | Tutor review: delivery candidate		Todo	Tutor review	Nov 26, 2026	Week 12 | 26 Nov
5	03 Dec | Tutor review: final tutor delivery review		Todo	Tutor review	Dec 3, 2026	Week 13 | 03 Dec
6	06 Dec | Canvas submission: 23:59 HKT		Todo	Submission	Dec 6, 2026	Week 13 | 03 Dec
7	Agree your first goal and Definition of Done		Backlog	Team task		Week 8 | 29 Oct

Use this template → Owner: sd5913 → Name your group's Project

Open Group Project 2026 →

POLYU DES°GN

Set up your team tracking.

1 / Open Group Project 2026 → **Use this template** → **sd5913**.

2 / Set up the name, repository link and teammate Project Write access.

3 / Plan issues in weekly iterations; use the dated tutor review cards.

4 / Track Status; link the PR, tests and review evidence.

A PR description with **Closes #12** closes issue #12 on merge to the default branch.

POLYU DES°GN

Bring a usable delivery each week.

Your assigned tutor / your scheduled tutorial slot that week

05 Nov	First working interaction
12 Nov	Integrated core experience
19 Nov	Playtest and feedback
26 Nov	Delivery candidate
03 Dec	Final tutor delivery review

Canvas deadline: 06 Dec 2026 / 23:59 HKT

Group Project 2026 template →

POLYU DES·GN

Show how your team makes progress.

Plan: ordered tasks, clear owners and a goal for the next cycle.

Build: small, steady contributions and usable Increments.

Review: show what works; record tests, feedback and blockers.

Adapt: document weekly progress and the next improvement.

Link your board and progress notes from the README. Teamwork guide →

POLYU DES•GN

Does this count as an Increment?

Team A: ten closed issues, but the project does not run.

Team B: one playable interaction, integrated and verified against the Definition of Done.

Which team can demonstrate a usable Increment?

Where do they inspect the product? Where do they improve teamwork?

POLYU DES°GN

1. Alice starts a change

1

ALICE / LOCAL

Pull main; branch

Edit, test, commit

→

2

ALICE / LOCAL

Push to GitHub

git push

→

3

ALICE / GITHUB

Open the PR

Request a review

POLYU DES•GN

2. Bob reviews the PR

1

BOB / REVIEW

Read the PR

Run and test it

→

2

BOB / GITHUB

Request a change

Explain the problem

→

3

ALICE / LOCAL

Fix and test

Commit and push

POLYU DES·GN

3. Both update main

1

BOB / GITHUB

Check the fix

Approve the PR

→

2

ALICE / GITHUB

Merge the PR

Integrate into main

→

3

BOTH / LOCAL

Switch to main

Pull the update

POLYU DES·GN

From change to shared main

1 / Branch		2 / Change		3 / Review		4 / Merge
Update main	→	Edit and test	→	Open a PR	→	Merge on GitHub
New branch		Commit; push		Teammate tests		Everyone pulls

A local commit becomes shared when you push. Teammates update when they pull.

POLYU DES°GN

A small change, step by step

```
git switch main
git pull origin main
git switch -c feature/start-button
# edit and run the project checks
git add .
git commit -m "Add start button"
git push -u origin feature/start-button
```

Then open a pull request on GitHub.

POLYU DES•GN

Make it easy to review.

What changed? “Start button opens the first game screen.”

How can a teammate run it and try the change?

What did you test? What is still untested?

Reviewer checks the diff and tries the interaction before merge.

POLYU DES°GN

Two people changed the same line.

Git marks both versions

```
<<<<<<< HEAD
label = "Play"
=======
label = "Start"
>>>>>>> feature/start-button
```

Agree on the result

```
label = "Play"
```

Remove the markers.
Run it; commit the resolution.

Resolve the intention together; then test the interaction.

POLYU DES°GN

Choose how the team collaborates.

1 / Shared repository

ONE TEAM REPOSITORY / teammates have write access

Feature branch → Pull request → **Team main**

2 / Fork + pull request

Contributor's fork
Their own copy + branch

PULL REQUEST →

Team repository
Review + merge to main

Git reference / Both paths need review.

03

Build in the cloud

A CODING ENVIRONMENT IS A PREPARED WORKSPACE.

POLYU DES°GN

Where does the work happen?

Codex cloud		GitHub Actions		Destination
Develop a change	→	Run a fresh job	→	Deliver the result
Tools + repository		Tests + build		Artifact or website

Prepare a workspace → build a change → deliver and try the result.

POLYU DES°GN

Create a cloud environment.

1 / Connect GitHub and choose the repository.

2 / Prepare an environment; review setup and dependencies.

3 / Publish the environment, then start a task with it.

4 / Verify one change: review the diff and check the result.

Official Codex cloud guide

POLYU DES·GN

Choose a starting point.

Capacitor

Web interface

Web + Android

Open template →

Godot

2D game

Desktop + web

Open template →

Python + uv

Python app

Source + desktop bundles

Open template →

Use this template → Owner: sd5913 → open in Codex → check CI.

POLYU DES°GN

Ask for one verifiable change.

“Make the player move faster after collecting a star.”

Name the file or behaviour and the expected result.

Ask for the relevant checks and how to run the result.

Inspect the diff, run it, request a teammate review.

Take a break.
Back in 10 minutes.

04

From commit to delivery

GITHUB ACTIONS RUNS THE WORKFLOW FOR YOUR TEAM.

POLYU DES°GN

Follow the result

Push / PR		Checks		Artifact
Workflow starts	→	Install and test	→	Download and run
Fresh runner		Build on each OS		Or deploy website

Open the run → inspect the jobs → download or visit the result.

POLYU DES•GN

One project, several delivery paths.

Desktop workflow: Windows, macOS and Linux matrix.

Android and iOS: mobile builds and signing.

Production deployment: Cloudflare Worker.

Build artifacts first. Stores and release signing are additional steps.

POLYU DES°GN

Give a job access to a token.

1 / Create a scoped provider token for the destination.

2 / Open repository Settings → Secrets and variables → Actions.

3 / Add a New repository secret: CLOUDFLARE_API_TOKEN.

4 / Reference it in the workflow; keep the value out of Git.

POLYU DES°GN

A secret is supplied to the job

```
jobs:
  deploy:
    environment: production
    runs-on: ubuntu-latest
    steps:
      # checkout, install and build first
      - run: npx wrangler deploy
        env:
          CLOUDFLARE_API_TOKEN: ${{ secrets.CLOUDFLARE_API_TOKEN }}
```

Configure the destination and token separately.

POLYU DES°GN

The build is green. Is it delivered?

Open the Actions run: which job ran, and which artifact exists?

Download and run the artifact, or open the deployed destination.

Environment-scoped secret missing? Check the job names that environment.

Provider tokens need access to the correct destination.

05

Camera → interaction

OPENCV CAPTURES FRAMES. MEDIAPIPE FINDS STRUCTURE.

POLYU DES°GN

From light to an action

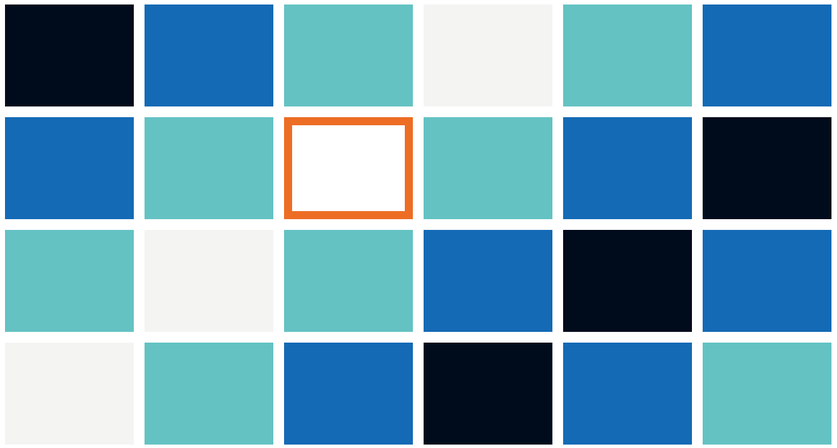
OpenCV		MediaPipe		Your code
Camera → frame	→	Model → landmarks	→	Map the result
BGR → RGB		Coordinates + confidence		Draw or change state

A model predicts structure. Your code decides what happens next.

POLYU DESIGN

What is a camera frame?

A grid of colour values



One pixel: three channels



OpenCV: BGR
MediaPipe input: RGB

Video = frame → frame → frame over time

POLYU DES°GN

Classes, objects and methods

Class	Object	Method call
Ball	red	red.update(0.1)
VideoCapture	camera	camera.read()
HandLandmarker	tracker	tracker.detect_for_video(...)

Create an object once. Call its methods as the program runs.

POLYU DES°GN

Read a frame from the camera

```
camera = cv2.VideoCapture(camera_index)
ok, frame = camera.read()
if not ok:
    raise RuntimeError("Camera frame unavailable")
frame = cv2.flip(frame, 1)
rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
```

One method call returns a success flag and an image array.

POLYU DES°GN

Configure the tracker

```
base = mp.tasks.BaseOptions(model_asset_path=str(model_path))
options = mp.tasks.vision.HandLandmarkerOptions(
    base_options=base,
    running_mode=mp.tasks.vision.RunningMode.VIDEO,
    num_hands=1,
)
```

Options describe the task; the .task file contains the trained model.

POLYU DES°GN

Call a method for each frame

```
with HandLandmarker.create_from_options(options) as tracker:
    # Repeat for each RGB frame and increasing timestamp
    image = mp.Image(image_format=mp.ImageFormat.SRGB,
                     data=rgb)
    result = tracker.detect_for_video(image, timestamp_ms)
```

The method returns a result object; it does not choose a game action.

POLYU DES°GN

Look at what the model returns.

Face detector: a box around each detected face.

Hand Landmarker: 21 points per hand, including index fingertip 8.

Move closer, rotate your hand, hide it: when does tracking fail?

Modern source and run commands

POLYU DES°GN

Find the index fingertip

result.hand_landmarks[0][8].x

hand_landmarks

List of detected hands

→

[0][8]

First hand → fingertip 8

→

.x

Horizontal position



0 / left edge

0.5 / centre

1 / right edge

POLYU DES°GN

What if no hand is found?

```
if result.hand_landmarks:
    x = result.hand_landmarks[0][8].x
else:
    x = None
selected = tile_for_x(x)
```

Check the list before indexing; None means no input.

POLYU DES°GN

Turn a position into a choice

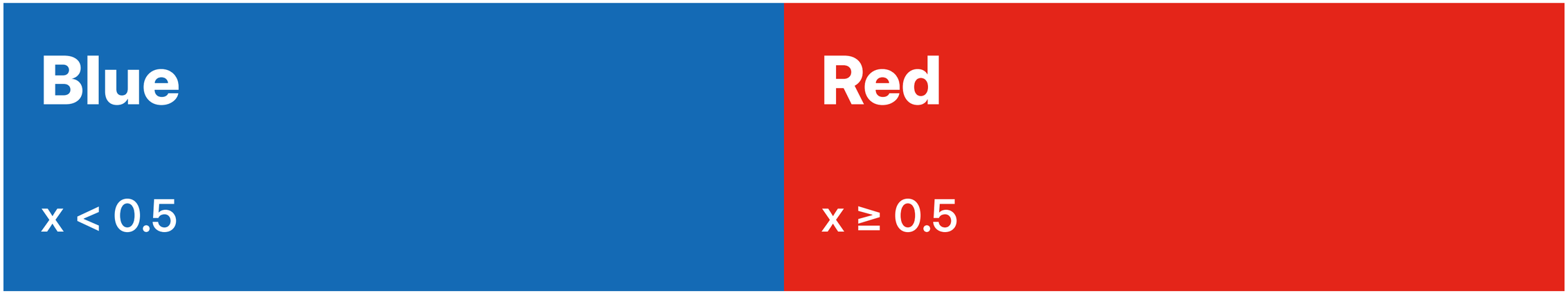
```
def tile_for_x(x):
    if x is None or not isinstance(x, (int, float)):
        return None
    if not math.isfinite(x) or not 0 ⩽ x ⩽ 1:
        return None
    return "blue" if x < 0.5 else "red"
```

Predict: 0.49 → ? 0.50 → ? None → ?

POLYU DES°GN

A coordinate becomes a choice.

Index fingertip / landmark 8 / horizontal position x



No hand → None → neither tile is selected.

Open the hand-to-tile example →

POLYU DES°GN

Draw a point on the frame

```
height, width = frame.shape[:2]
for point in points:
    pixel = (int(point.x * width),
             int(point.y * height))
    cv2.circle(frame, pixel, 2, colour, -1)
```

Normalised coordinates × image size → pixel coordinates.

POLYU DES°GN

Compare two kinds of output.

Gesture Recognizer: a trained model returns a gesture label and score.

Pose Landmarker: body points can drive an interaction.

A score is model confidence, not proof of intent.

Examples and source

POLYU DES°GN

A mask changes the background.

The model estimates a person mask for each frame.

OpenCV combines foreground with a replacement background.

Try hair, a raised hand and a busy background.

All Week 8 vision exercises

POLYU DES°GN

Try all nine vision examples.

Camera → face / hand / pose → mesh / gesture

Combine models. Try segmentation. Question expression labels.

Open the PFAD Week 8 tutorial →

Could this inspire your project?

Computer vision is optional. Try hand tracking in a browser →

Build together. Show your work.

One idea. One shared repository. A working next step.

SD5913 / WEEK 08 / 29 OCTOBER 2026