

Intro to P5.js + Agile Techniques

Workshop 2

Ruzanna Chitchyan, Jon Bird, Pete Bennett

TAs: Alex Cockrean, Leifan Lei, Guoda Laurinaviciute, Ahmed El Ashray, Senyi Luo

Today's Workshop

- Project examples from last year (5 mins)
- Introduction to P5.js (30 mins)
- Develop an app (~ 60 mins)
 - Practice Pair Programming
 - Practice Kanban board use
- Try some (very light) evaluation (10 mins)



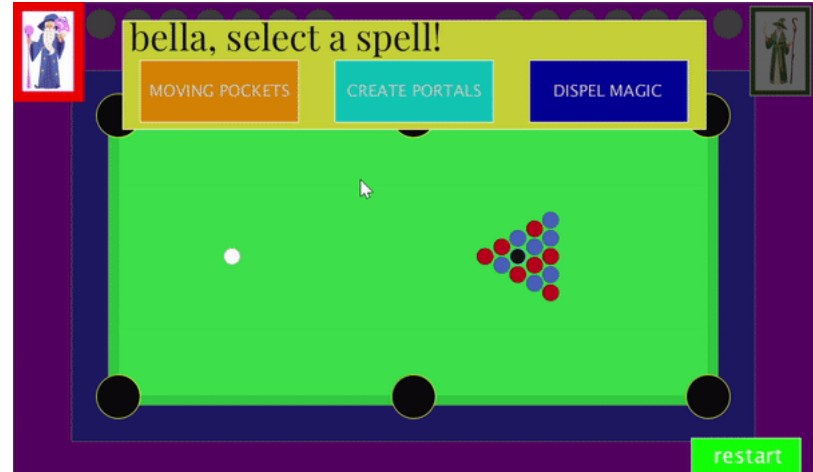
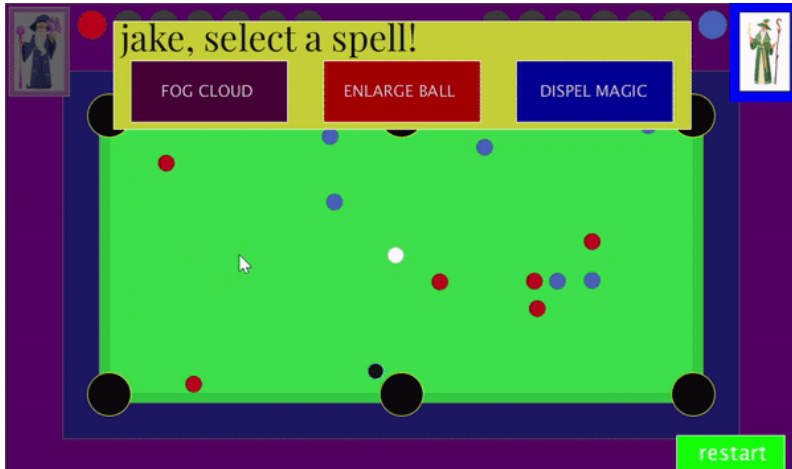
Coursework

- Any questions about the coursework brief?
- Have you made contact with everyone in your team? Let us know now if not!
- Have you made a commit on your repo?
 - Team Photo
 - List of games (inspiration and ideas)



Game Example: Wizard Pool

- "pool with a twist of magic"
- Challenges: Ball collision, spell system, tutorial



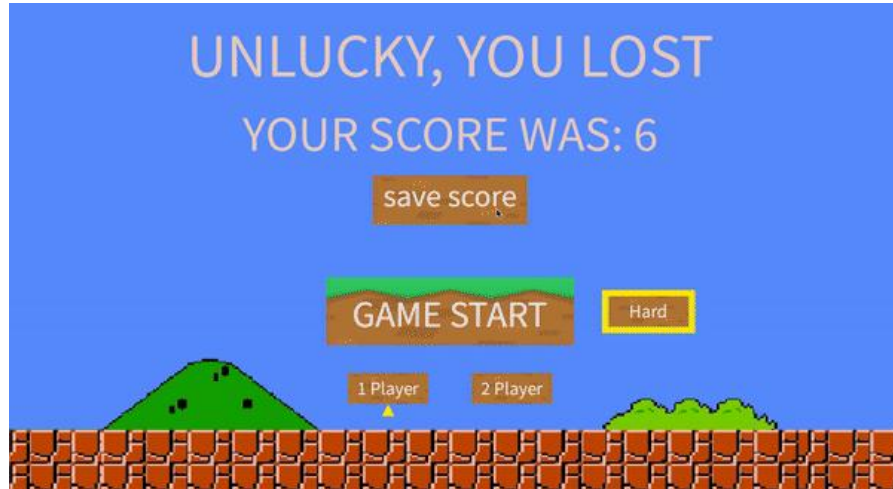
Game Example: JUPITER X Resource War

- Based on the rogue-like game spelunky, but... based in space and has an unkillable ghost enemy [\[github\]](#)
- Challenges: generative map, collision detection, performance optimisation



Game Example : Topsy Turvy

- Classic platformer but... with gravity reversal
- Challenges: physics engine, multi-player mode, high-scores log



Simple Paint

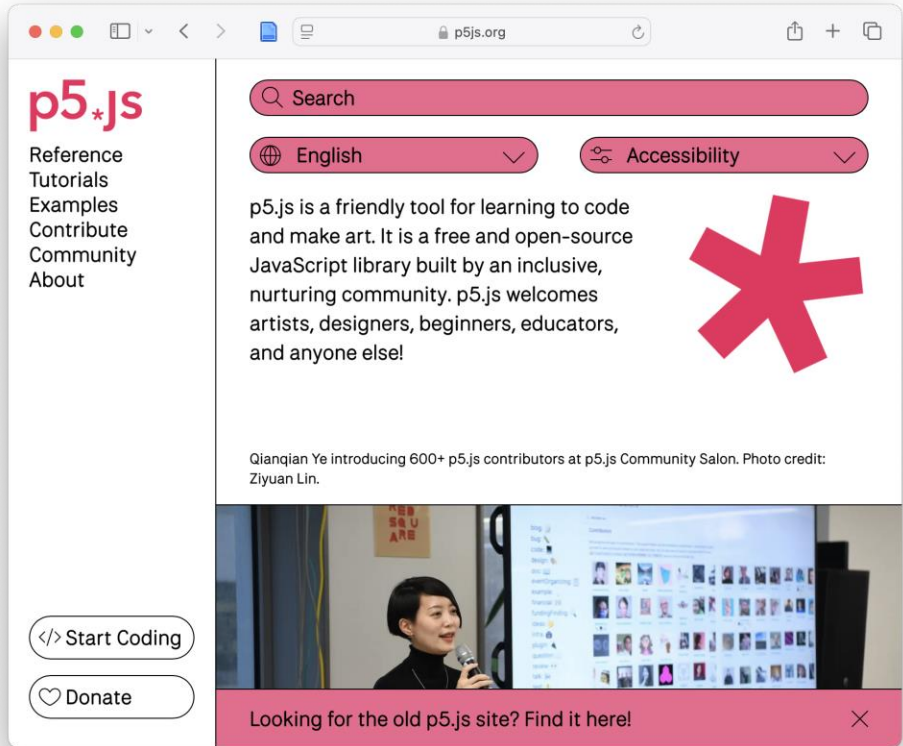
- Today you are going to develop in pairs a **simple drawing tool** in P5.js.
- Inspired by the classic Microsoft Paint, but... this is only a starting point, you will be adding any features you like.
- You will be swapping your paint app with another pair at the end to draw a **portrait**
- *...keep this use in mind before going off and making an abstract generative geometric paint program!*



[image](#)

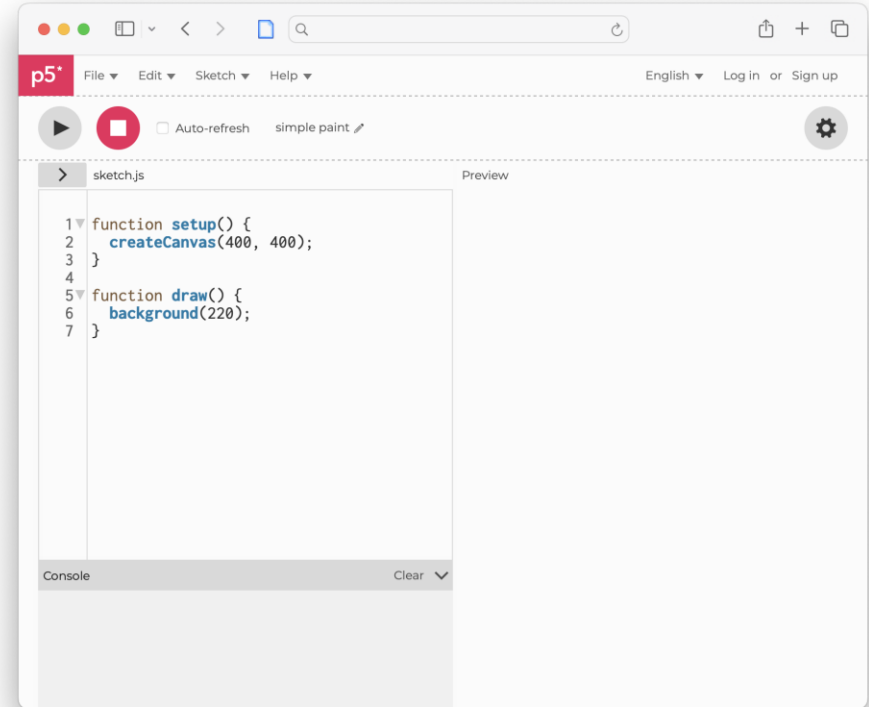
Follow Along Example

- Follow along with this quick demo.
- This will start you all off with a very bare bones Minimum Viable Product (MVP).
- Start by opening www.p5js.org in your browser, then click on the “Start Coding” button here



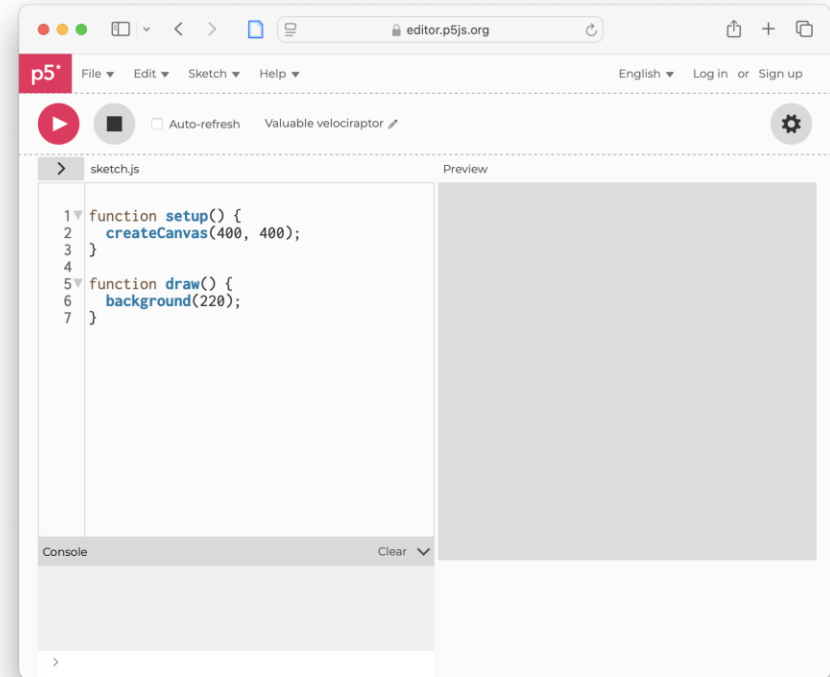
P5.js editor

- This is the online P5.js editor:
www.editor.p5js.org
- You can start coding in this online interface, but can easily migrate to other editing environments such as VSCode:
[instructions here](#)



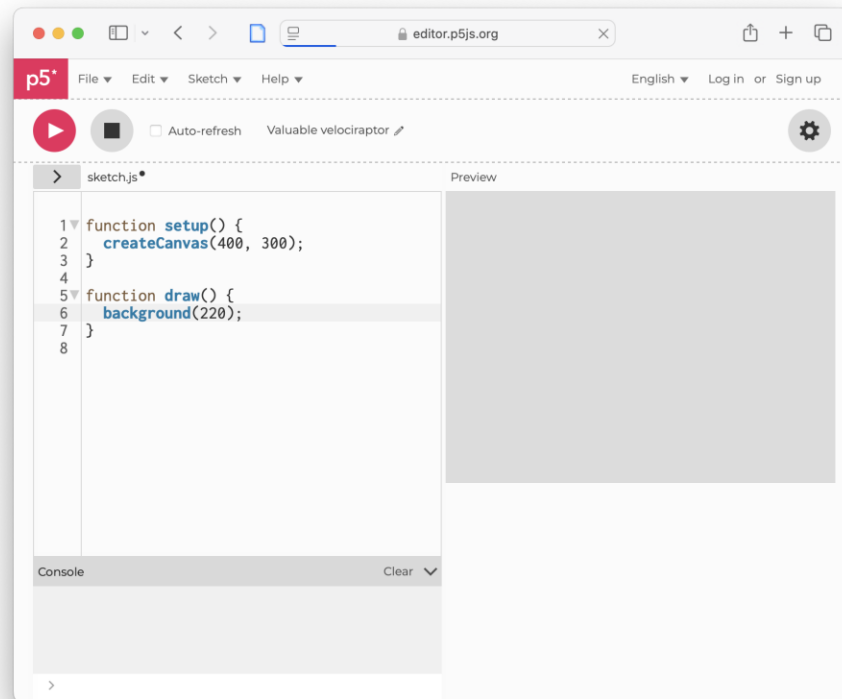
Run + Stop

- Use the start and stop buttons to start and stop your sketch from running. Turn on auto refresh for code to be run when updated.
- There are two main functions:
 - function **setup()** {
 // runs once at the start
}
 - function **draw()** {
 // runs every frame in a loop
}



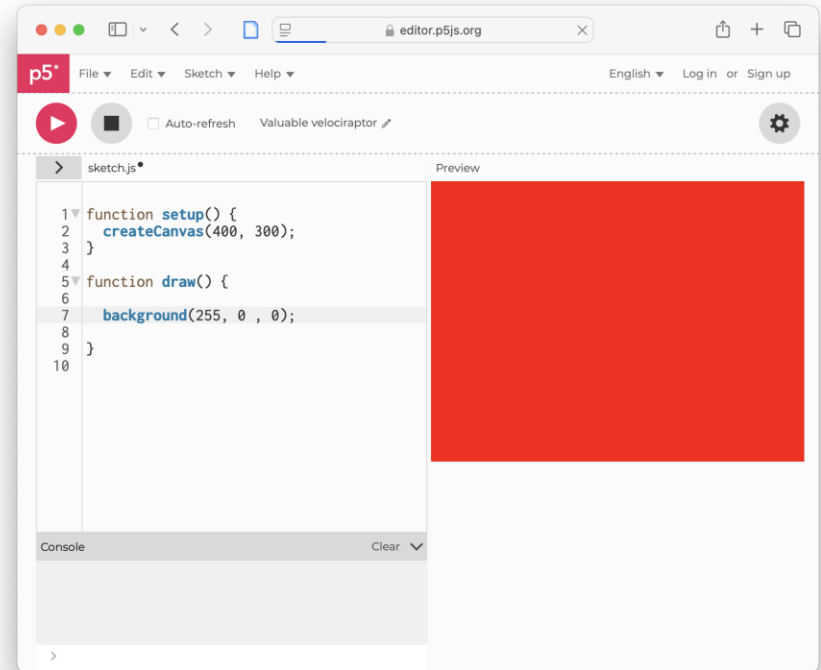
Canvas Size

- First task is to change the canvas size with `createCanvas()`:
 - `createCanvas(width, height);`
 - `createCanvas(400, 300);`
- The size is set in pixels
- Note: semi-colons help readability, but they're not always required. Have a look about exceptions in this post: [let's talk about semicolons in javascript](#)



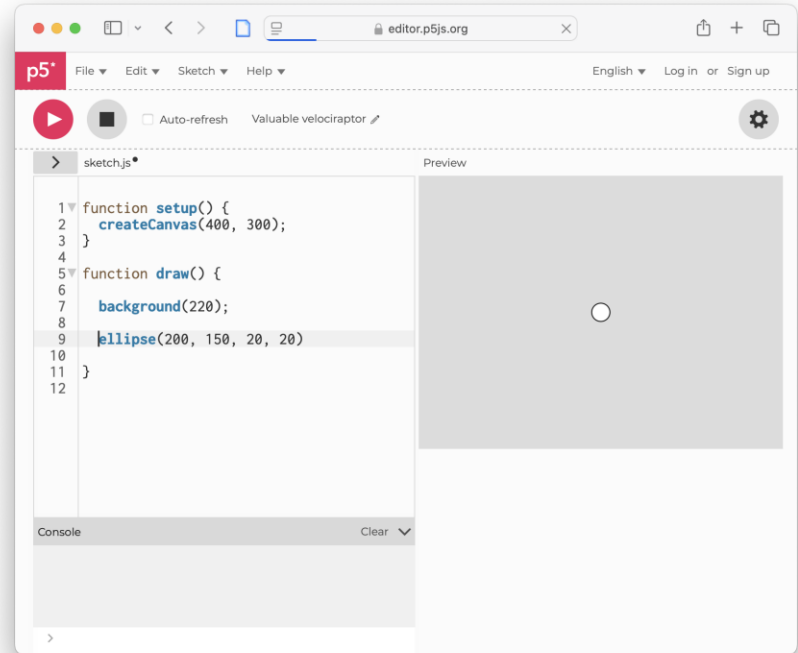
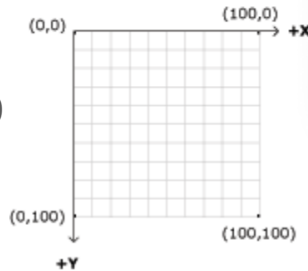
Background

- Background simply draws a colour over the entire canvas. All values are 0-255
- Pass a single value for greyscale
 - `background(255)` = white
 - `background(0)` = black
- Or three values for colour
 - `background(red, green, blue)`
 - `background(255, 0, 0)`



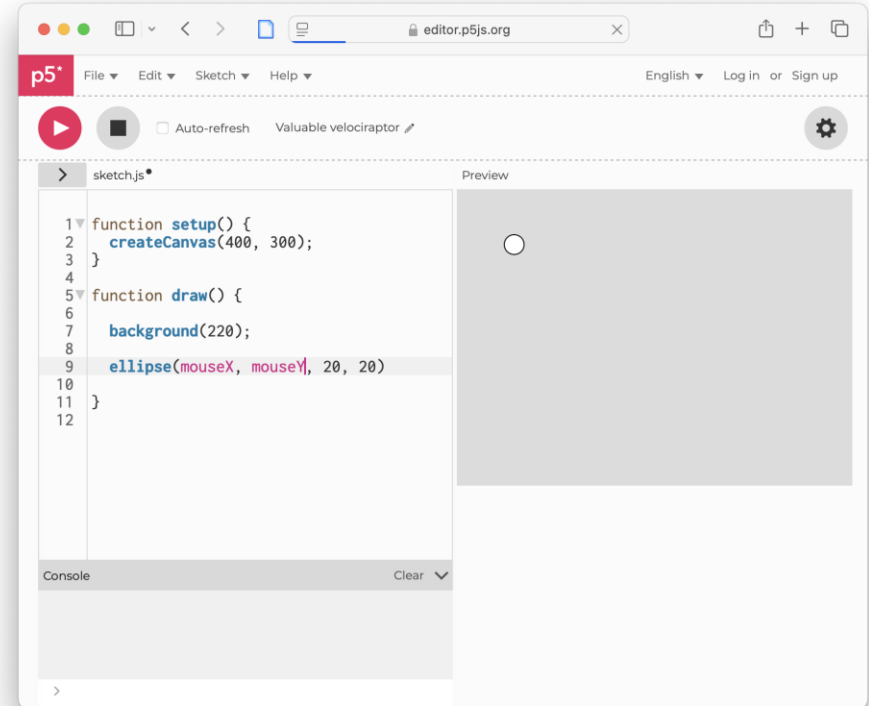
Draw a Circle

- Now draw a circle using the ellipse function that takes four arguments:
 - `ellipse(x, y, width, height);`
 - `ellipse(200, 150, 20, 20);`
- To draw a rectangle use:
 - `rect(x, y, width, height);`
- Coordinate system: **top left is 0, 0**



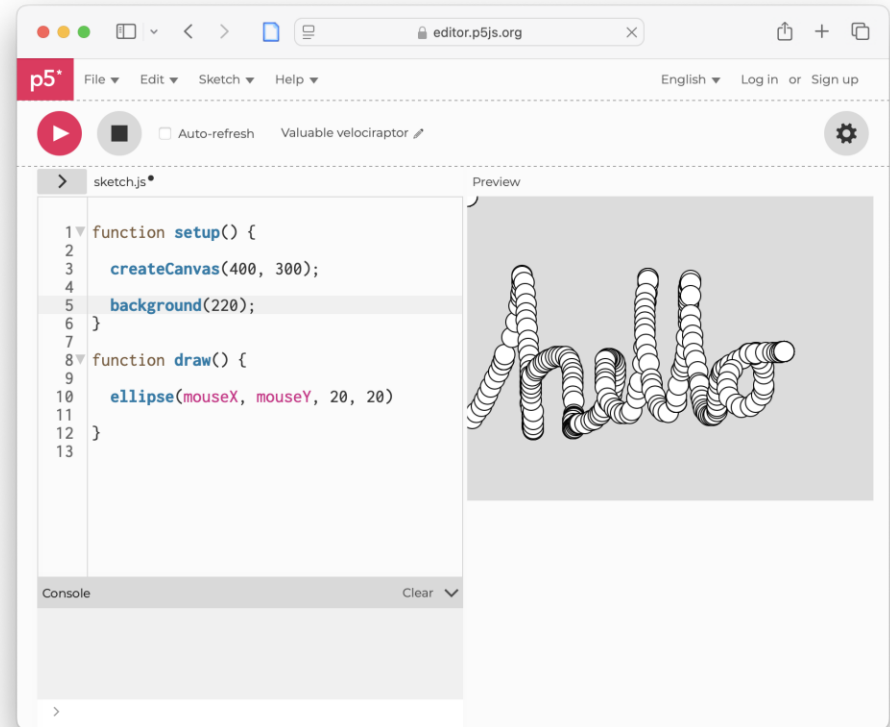
mouseX & mouseY

- P5.js has some handy global variables predefined that you can access
 - **mouseX** returns the current mouse x pos
 - **mouseY** returns the current mouse y pos
- The IDE highlights the variable pink to show that it's a predefined variable.
- Other useful ones include **width**, **height** of the sketch window. And previous mouse positions **pmouseX** and **pmouseY**.



leaving a trace

- Let's move the `background()` call up into `setup` so it only runs once at the start
- You now have a basic drawing app!



MVP achieved!

- You all now have the starting point of a very basic paint program.
 - Try painting the person next to you.
- Your job for the remainder of the workshop is to work in pairs using pair programming and a kanban board to improve on this.
- You will be testing your creation out on another person (with no instructions!) to make a short life drawing portrait at the end of the workshop. But... here are some starting points:



Changing Colour

- Colours values are between 0-255,
- fill(*red, green blue*)
 - Changes the fill colour for the next drawing command.
- stroke(*red, green, blue*)
 - Change the colour for the outline of the next shape
- If you add a fourth argument you can set the opacity (*red, green, blue, opacity*)



[image](#)

Canvas / Background

- Change the background colour of the sketch by calling `background(r,g,b)`
- Try loading an image as a background using `image()` ... note, do this in setup, rather than the draw loop!
- Explore using paper, canvas textures, a sketchbook or use a photograph (say of sky and clouds for a cloud drawing app)



Random

- Try randomising some elements with:

[random](#)(*max*)

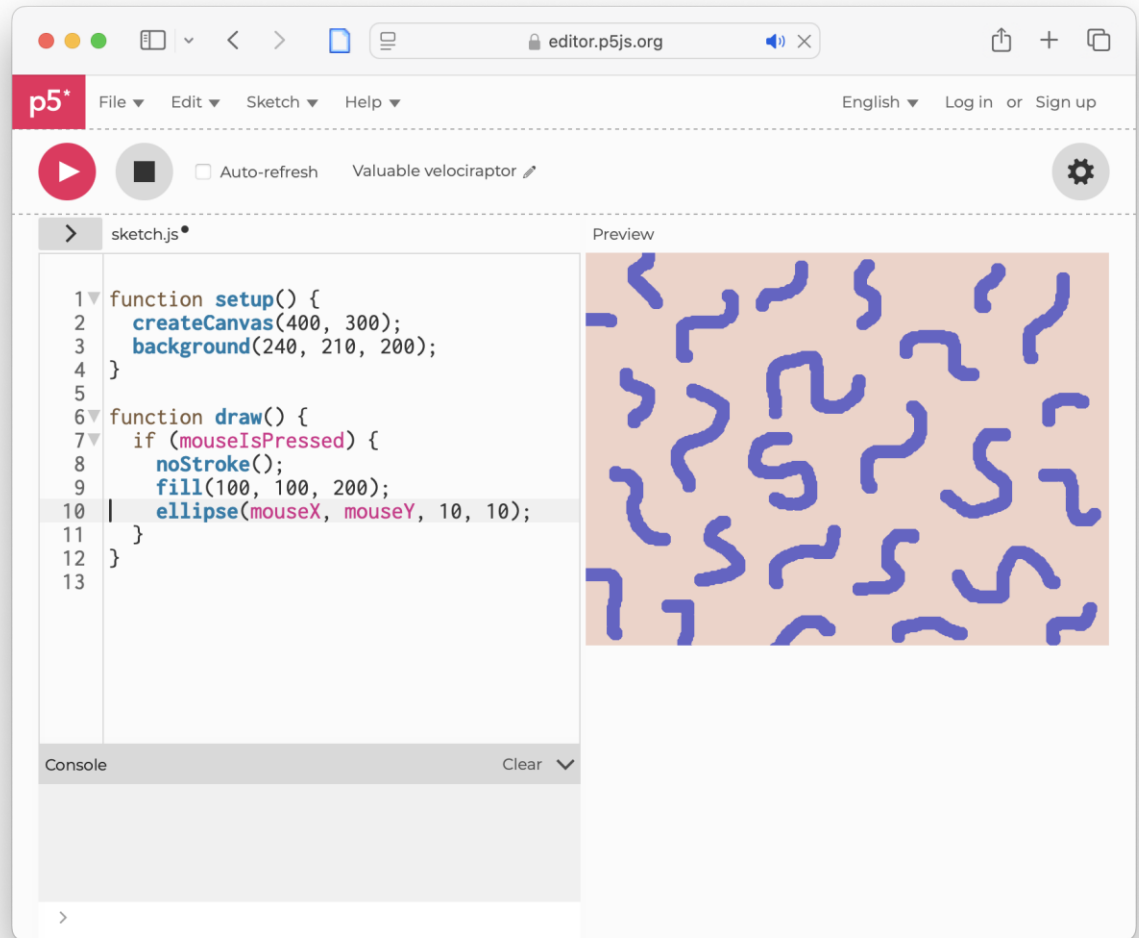
[random](#)(*min*, *max*)

- For instance filling a shape with a random colour:
 - `fill(random(255, random(255), random(255));`



[image](#)

pen up – use the predefined **mouseIsPressed** variable to only draw when the mouse is being pressed

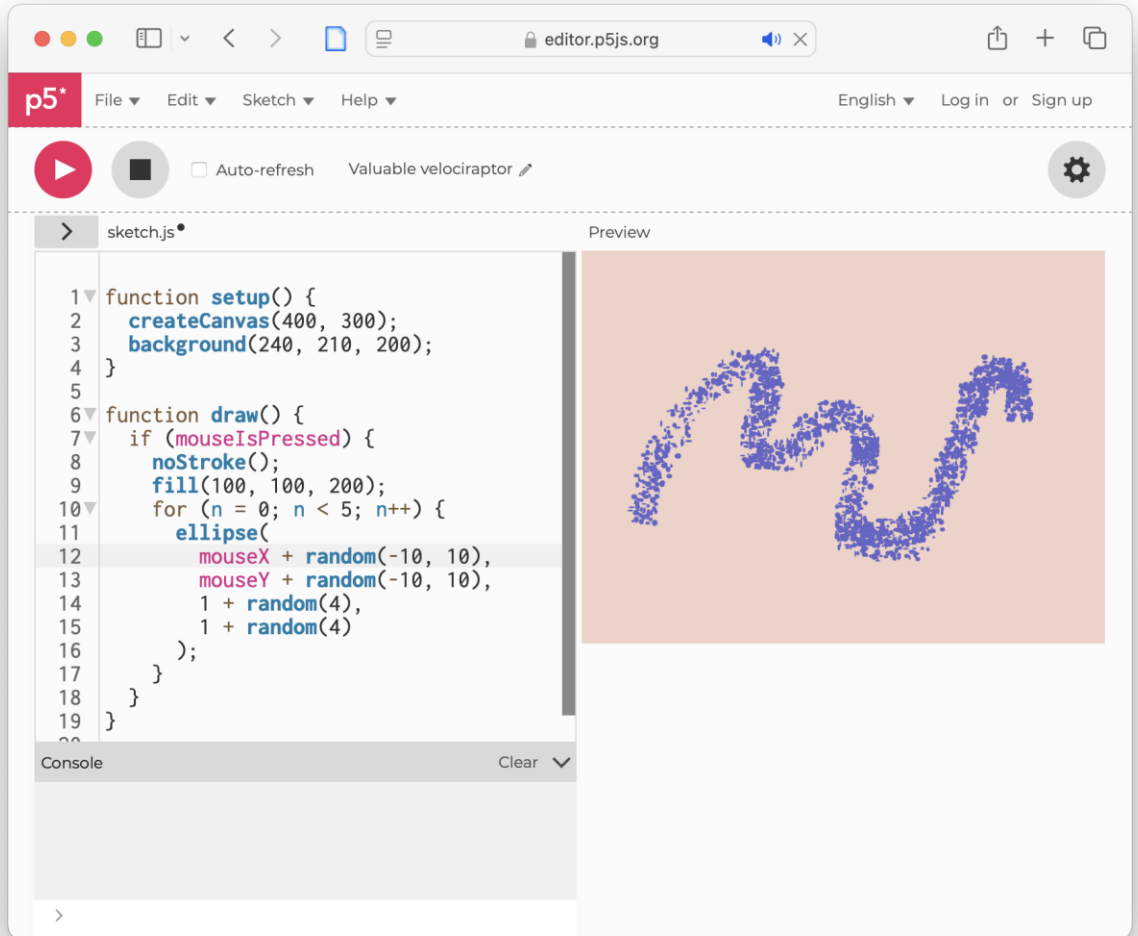


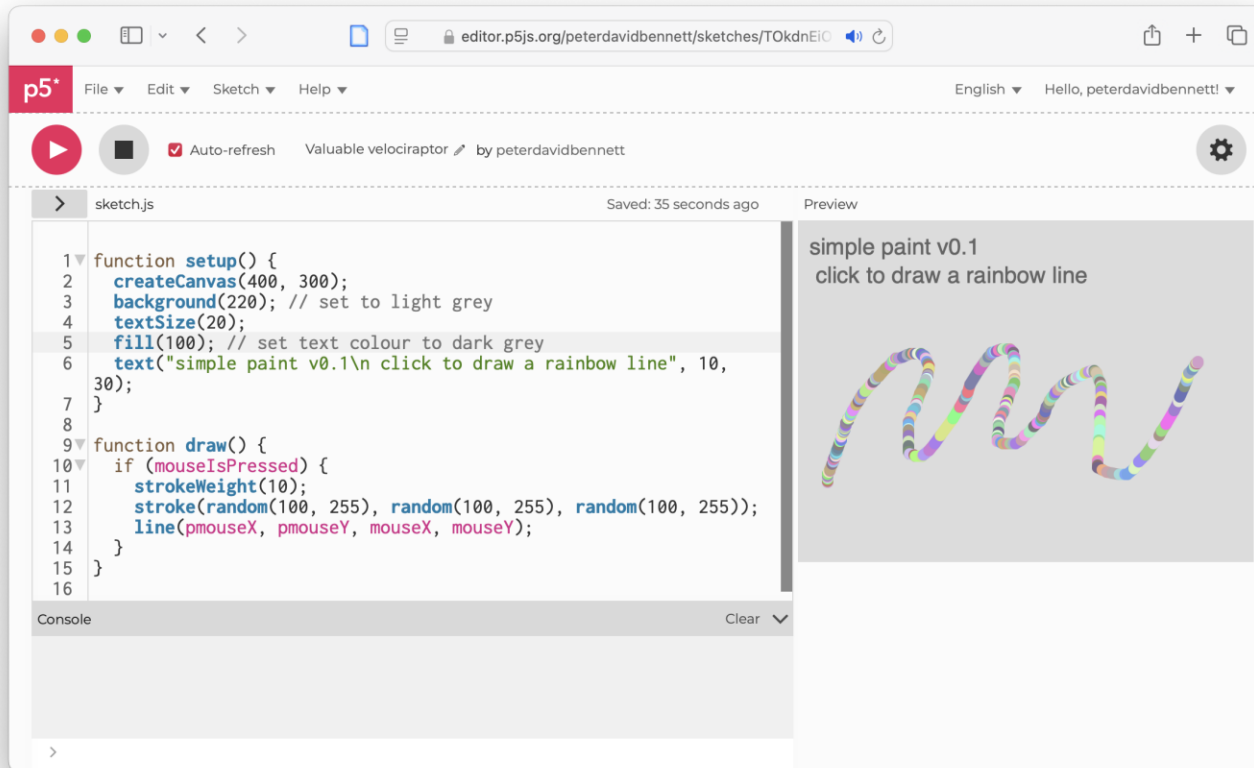
Spraypaint Example

Could be improved, perhaps use **cos()** and **sin()** to create a circular area rather than a square?

How about subtly changing the colour and opacity of each dot?

Can you make it so that the spraypaint slowly runs out of paint?





Documentation + Commenting - make sure that a new user can work out how to use your app, and please remember to use comments, either `//` or `/*multiline*/` so that your code is legible

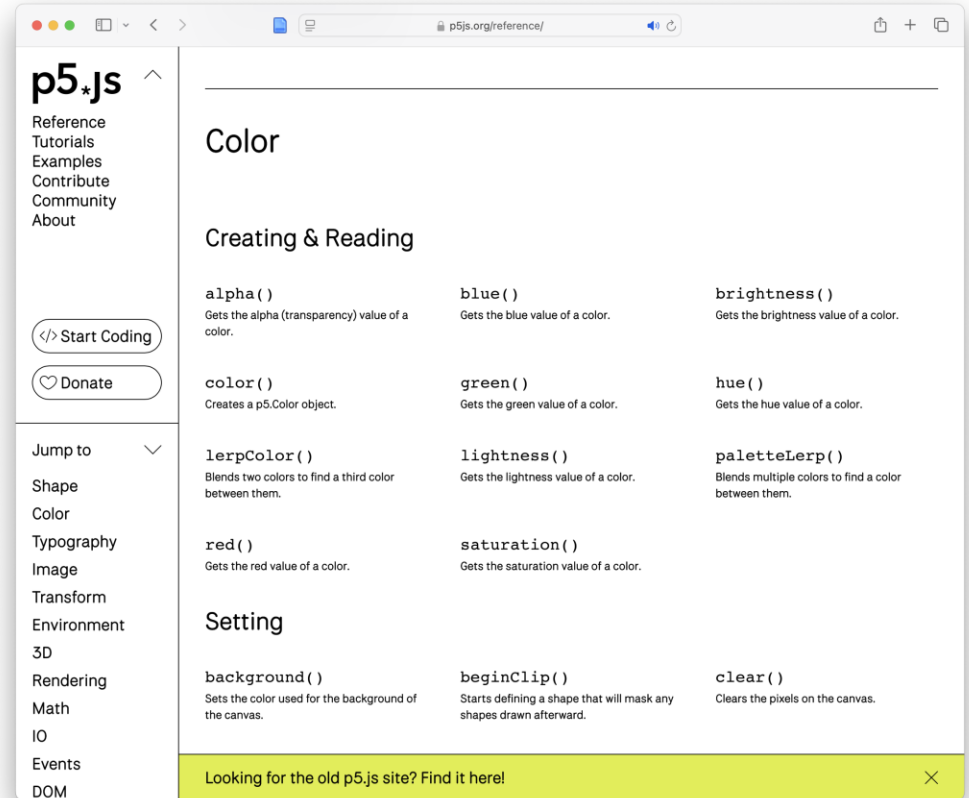
Challenges

These are potential challenges that you could start to fill your Kanban board up with:

- Add a way to change brush colour
- Change the brush size
- Add key commands `keyPressed()`
- Save an image with `save()` (“example.jpg”)
- Multiple brush types, select with a key
- Use an `image()` as a brush, rotate randomly to vary the stroke
- Add an eraser. Can this be on right click?
- Add a way to reset/clear the whole canvas
- Auto-scribbler, automatically nudge the pen around the drawing position
- Symmetry – draw a second point on the opposite side of the canvas
- Create smooth lines by drawing between last and current mouse position:
`line(pmouseX, pmouseY, mouseX, mouseY)`
- Add opacity to your brush by passing a fourth parameter (0 = transparent, 255 = opaque):
`fill(red, green, blue, alpha)`
- Create lined or gridded paper using a for loop.
- Change the `strokeWeight()` of your line based on the speed of mouse movement. Use `pmouseX` and `pmouseY` along with `mouseX` and `mouseY` to determine the distance moved
- **NOTE:** please add **documentation** – someone’s going to be using your system without you being able to explain it. Use `print()` to send text to console or even better `text()` to write to screen.

Going Further

- Many more functions documented on the P5.js website:
 - reference: <https://p5js.org/reference/>
 - tutorials: <https://p5js.org/tutorials/>
- Look through the reference and see what functions could be interesting to try out!



Pair Programming + Kanban

- Partner up. Swap roles regularly.
- Use a Kanban board to plan and track which features you will be working on. Three columns, move features across:
 - *Not Started - > In Progress -> Done*
- Start simple! Keep features small.
- Consider adding a 'shelved' or 'parked' column to put features in that aren't working out (given ~60min timescale). Don't get stuck!

~60
mins



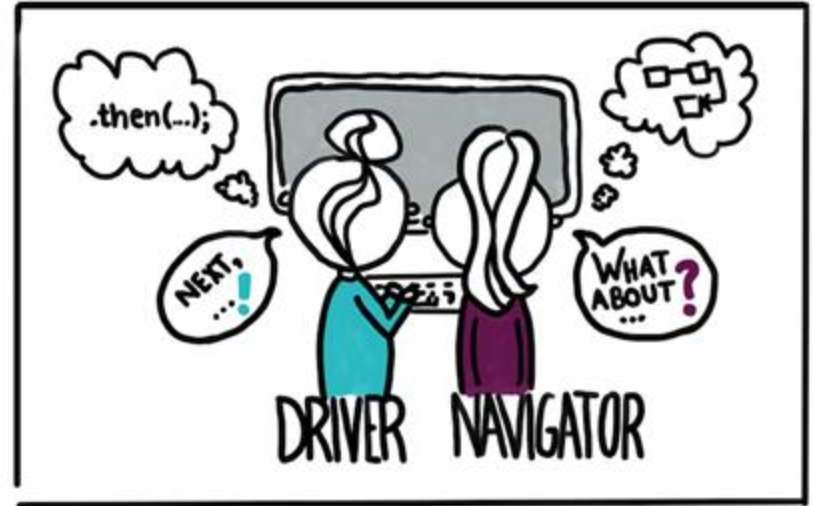
Pair Programming Roles

Driver (Helm)

- The person typing
- Focused on the short-term goal
- Places longer-term goals on backburner
- Talks through what they are doing

Navigator (Tactician)

- Observes the drivers work
- Reviews code in real-time
- Notes suggestions
- Scans horizon for longer-term issues



Pair Programming Tips

- **Distractions.** Don't check your phone/mail. Stay focused. Build in more individual time if you need it.
- **Micro-management.** Stick to higher level comments, and avoid saying things such as "now type..."
- **Impatience.** Don't jump right in when the driver makes a typo. They may have seen it and just haven't gone back to correct. Avoid breaking their flow.
- **Keyboard hogging.** Make sure to stick to a rotation schedule and avoid sticking to one role.



The dark side of pair programming.

User Testing – Draw a Portrait

10
mins

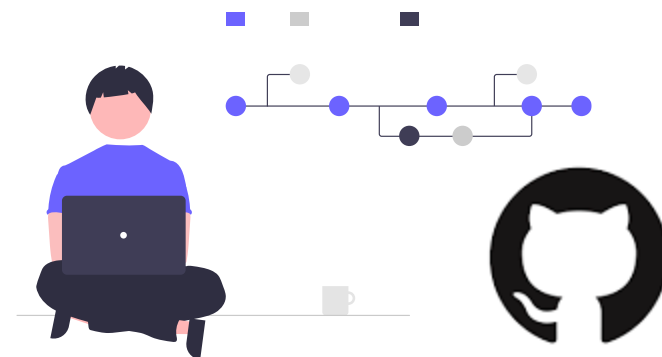
- Find another pair and swap over.
- **Without any instruction**, use their paint app to draw a portrait of your pair programming partner (take turns)
- Show the creators of the app how you used it, and your artistic results!
- What worked? What didn't? Did you enjoy it? What would you improve? What was similar with your own? How was the resulting portrait?



[image](#)

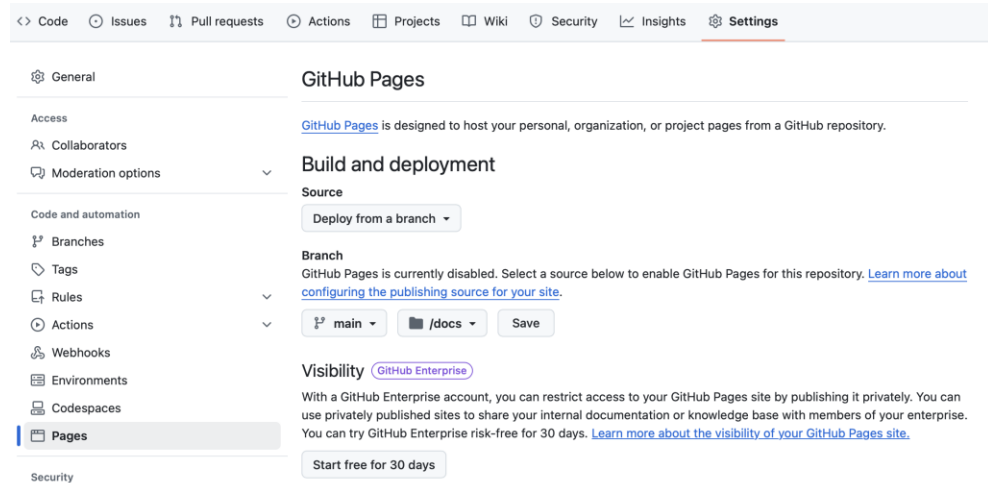
Homework / Groupwork

- Upload one of your team member's drawing apps from today. You have now deployed a working app that's publicly available! (*instructions on next slide*)
- Continue brainstorming ideas for your game. Decide on **TWO IDEAS** to bring along to next weeks workshop.
- **Add these two ideas to your repo!** One paragraph each. Inspiration images are encouraged!



Deploying your app to Github pages

- For a full video walkthrough, see here:
[Hosting a p5.js sketch with GitHub Pages](#)
- In short, go to your Github repo's:
Settings -> Pages
- Select Source as “Deploy from a branch”
- Select “main” branch and the folder “/docs”
- Click save, then place your sketch in the docs folder. Done!



Week	Lecture (Mon 2-3pm) CHEM BLDG LT2	Lab (Tues 2-4pm) MVB 2.11	Discussion/Guest Talk (Thurs 10-11am) QUEENS BLDG F.101a/b/c (Bill Brown Suite)	Homework (~10hrs per person)
1	Introduction & Process [slides]	Form Teams, Project Brief [slides] & Setup Repo [slides, exercises]	Jobs Post MSc: meet past graduates: Dr Dan Bennett , lecturer, UoB (2018 graduate) Ali Jardine , Data Platform Engineer at Depop, UK (2024 graduate) [slides]	Research games and create a list of inspiration on team repo (including game mechanics). Develop possible game ideas
2	Agile Software Development [slides]	Intro to P5.js & Agile Techniques [slides]		Deploy paint app using Github Pages + Write up two game ideas
3	Requirements Engineering	Review Your Two Projects, Paper Prototyping & Requirements Gathering		Refine Paper Prototype (translate to digital wireframe) & Collect additional requirements
4	Object Orientated Design	Object Orientated Activity (Classes)		Add requirements section to report (repo)
5	Project Management	Planning Poker, Ethics (Information Sheet, consent form)		Develop a working prototype over reading week!
6	Reading Week, no lecture	Reading Week, no lab		Games Jam (sprint over reading week)
7	HCI Evaluation - Qualitative	Think Aloud and Heuristic Evaluation		
8	HCI Evaluation- Quantitative	Quantitative		
9	Software Quality & Testing	Software Quality & Testing		
10	Sustainability & Accessibility	Add Accessibility & Sustainability Sections to Report		
11	Privacy, Security & AI	Add Privacy, Security & AI Sections to Report	Dr Pauline Anthonysamy , Senior Staff Privacy Engineer at Google Zurich, Switzerland	
12	Coursework Feedback & Marking Scheme	Game Demo	Project Report Hand in at 1pm (no talk)	

keep up to date!



Upcoming Events - Testathons

- **5th February & 5th March**, 2-4pm in 2.11 (here!) ... should be in your diary.
- This is an evaluation swap (trade) – bring your app and test others. Spend half your time testing and half testing other peoples.
- Bring anything from paper prototype to full game
- Highly recommended to attend both!
- Any questions, ask [Sarah Connolly](#)



Summer Project Info

- **2pm on Thursday 23rd** in Chemistry LT2
- Please attend for a general briefing session on your summer project, including individual vs group projects and timeline for project selection.

