

attendance code:

????

# Object Orientated Design

## Workshop 4

**Ruzanna Chitchyan, Jon Bird, Pete Bennett**

TAs: Mitch Lui, Craig Barnfield, Kira Clements, Ollie Myers

Week	Date	Lecture Monday 11:00-12:00 PHYS BLDG G44 FRANK	Workshop Monday 13:00-15:00 MVB 2.11 PC	Groupwork
1	23/01/22	Introduction and Process <a href="#">[slides]</a> <a href="#">[materials]</a>	Teams, Waterfall Method and Project Brief <a href="#">[slides]</a> (case study) <a href="#">[project brief]</a>	Research games, create list on team repo. Install Processing
2	30/01/22	Agile Software Development <a href="#">[slides]</a>	Intro to Processing, Agile Techniques <a href="#">[slides]</a>	Decide on two game ideas
3	06/02/22	Requirements Engineering	Paper Prototyping, Requirements, Ideas Clinic	Collect requirements. Decide on final idea
4	13/02/22	Object Orientated Design	Classes Activity	Add requirements section to report
5	20/02/22	Implementation	Case Study, Sprint Prep, Continuous Integration	Develop a working prototype over reading week!
6	28/02/22	READING WEEK	GAMES JAM	
7	06/03/22	Project Management	IN CLASS TEST (assessing lectures 1-4)	Define team roles
8	13/03/22	HCI - Qualitative	HCI Qualitative Task	Add qualitative assessment (of your choice) to report
9	20/03/22	HCI - Quantitative	HCI Quantitative Task	Add quantitative assessment (of your choice) to report
	27/03/22	EASTER week 1	SPRINT 1	
	03/04/22	EASTER week 2	SPRINT 2	
	10/04/22	EASTER week 3	SPRINT 3	
10	17/04/22	Software Engineering Extended	IN CLASS TEST (assessing lectures 5-9)	Develop Game
11	24/04/22	Coursework Feedback		Finish Report
12	01/05/22	Bank Holiday Monday (no class)	Demo Day Weds/Thurs (tbc)	Submit Report

# Today's Workshop

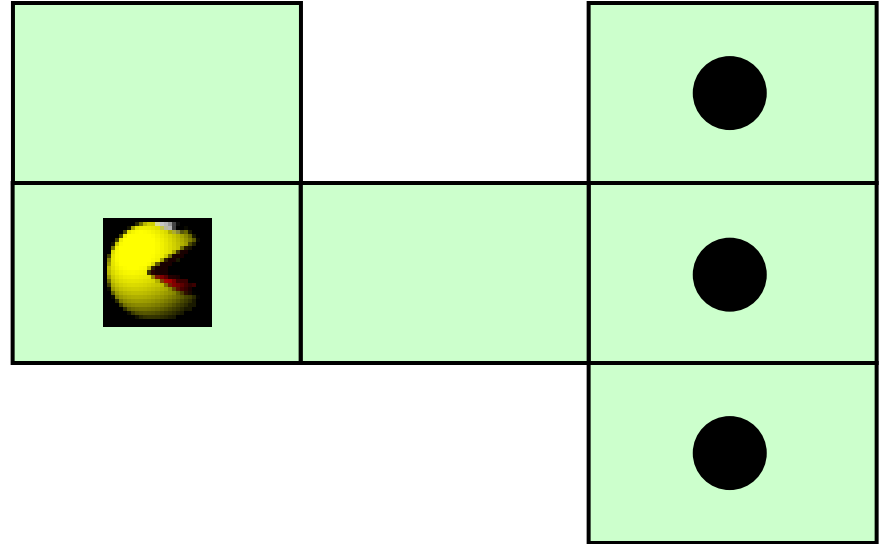
- Identify classes in Pacman (45mins)
- Classes challenges in Processing (45mins)
- Homework. Your team should now have one game idea. Draw up a class diagram for your game, add it to your repo and begin basic implementation.



# Class Diagrams

# (Simplified) Pacman Game

- The game is played on a **gridded board** which consists of **fields**. There are **various figures** that can be placed in fields. These are the **pacman** as well as **marble** and **pill**.
- **Marbles and pills are edible by pacman.**
- We need to keep a **record** of eaten figures.
- The game ends when all the marbles have been eaten.



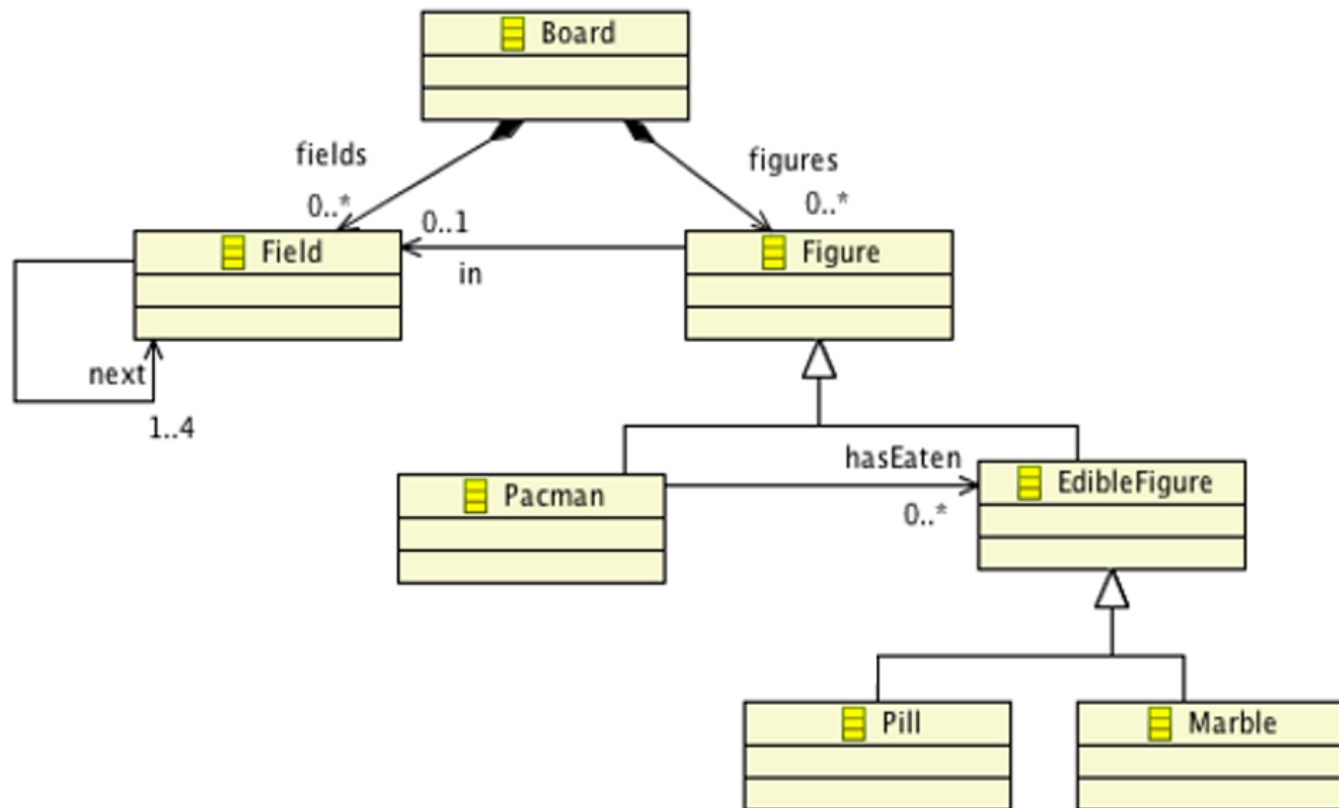
# To do (15 min)

Sketch a class diagram, making sure to note:

- Classes
- Associations
- Inheritance
- Cardinalities
- Feel free to note attributes and methods, if you want.



<https://en.wikipedia.org/wiki/Pac-Man>



# To do (20 min)

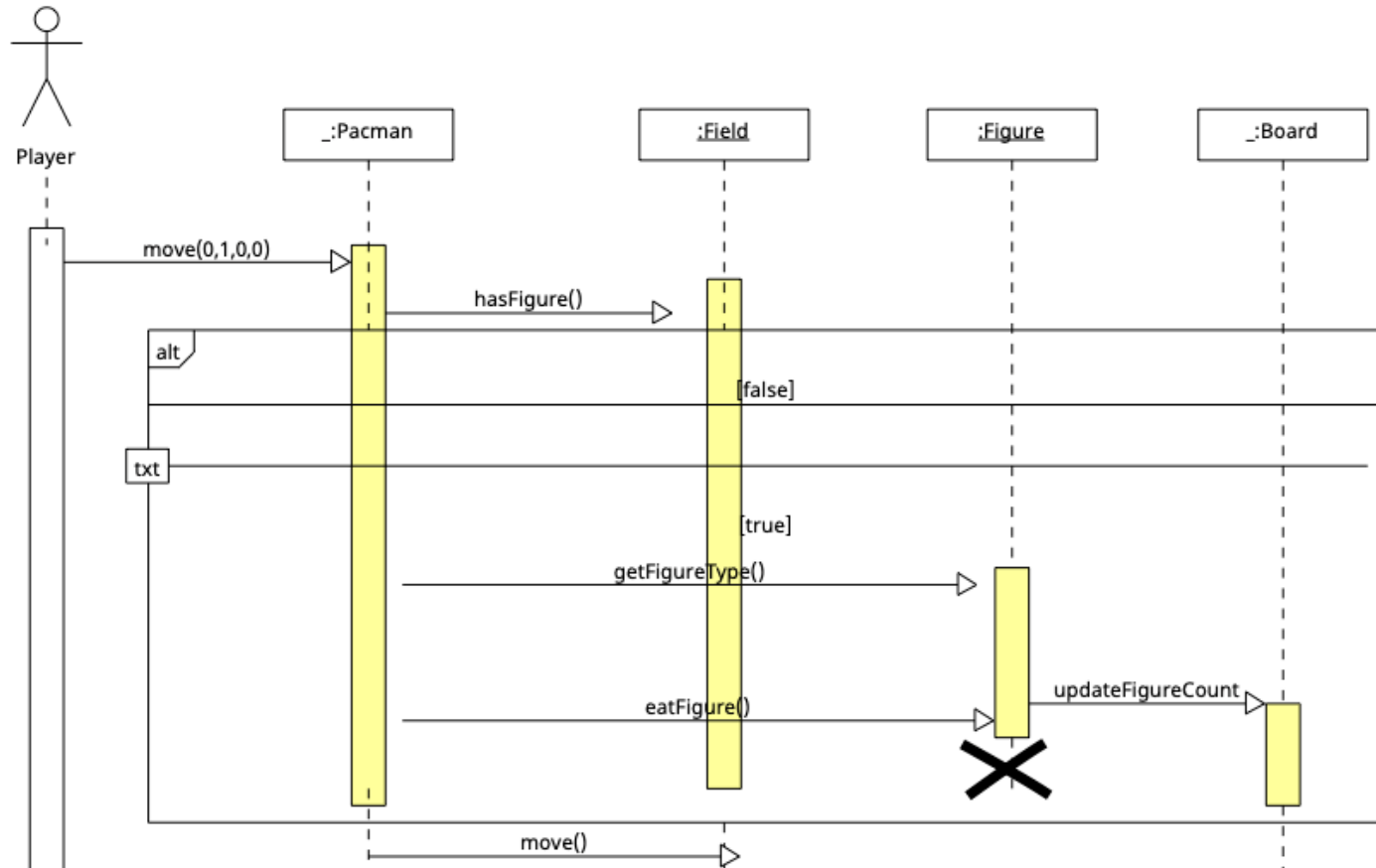
Sketch a

- communication or
- a sequence

diagram



# Example Solution: Sequence Diagram



**45  
mins**

# Classes in Processing

Processing Download Documentation **Learn** Teach About Donate

Tutorials Examples Books

## Text Tutorials

A collection of step-by-step lessons covering beginner, intermediate, and advanced topics.

**Getting Started**  
by [Casey Reas and Ben Fry](#)  
Welcome to Processing! This introduction covers the basics of writing Processing code.  
Level: Beginner

**Processing Overview**  
by [Casey Reas and Ben Fry](#)  
A little more detailed introduction to the different features of Processing than the Getting Started tutorial.  
Level: Beginner

**Coordinate System and Shapes**  
by [Daniel Shiffman](#)  
Drawing simple shapes and using the coordinate system.  
Level: Beginner

**Color**  
by [Daniel Shiffman](#)  
An introduction to digital color.  
Level: Beginner

**Objects**  
by [Daniel Shiffman](#)  
The basics of object-oriented programming.  
Level: Beginner

**Interactivity**  
by [Casey Reas and Ben Fry](#)  
Introduction to interactivity with the mouse and keyboard.  
Level: Beginner

**Typography**  
by [Casey Reas and Ben Fry](#)  
Working with typefaces and text.  
Level: Beginner

**Strings and Drawing Text**  
by [Daniel Shiffman](#)  
Learn how use the String class and display text onscreen.  
Level: Intermediate

**Objects Tutorial**  
by [Daniel Shiffman](#)  
The basics of object-oriented programming.  
Level: Beginner

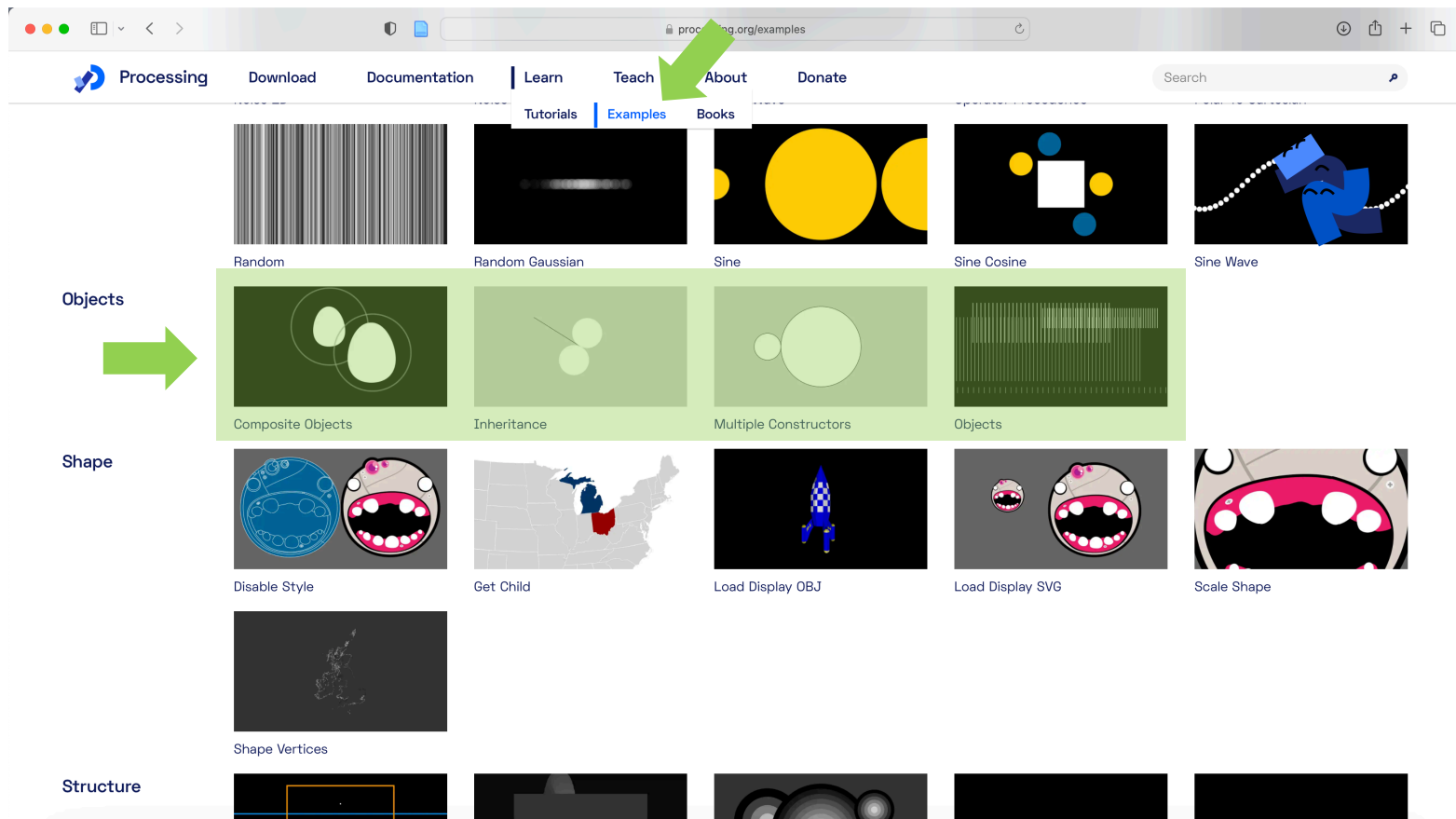
**Objects Tutorial**  
by [Daniel Shiffman](#)  
The basics of object-oriented programming.  
Level: Beginner

**Objects Tutorial**  
by [Daniel Shiffman](#)  
The basics of object-oriented programming.  
Level: Beginner

**Objects Tutorial**  
by [Daniel Shiffman](#)  
The basics of object-oriented programming.  
Level: Beginner

Open "https://processing.org/tutorials" in a new tab

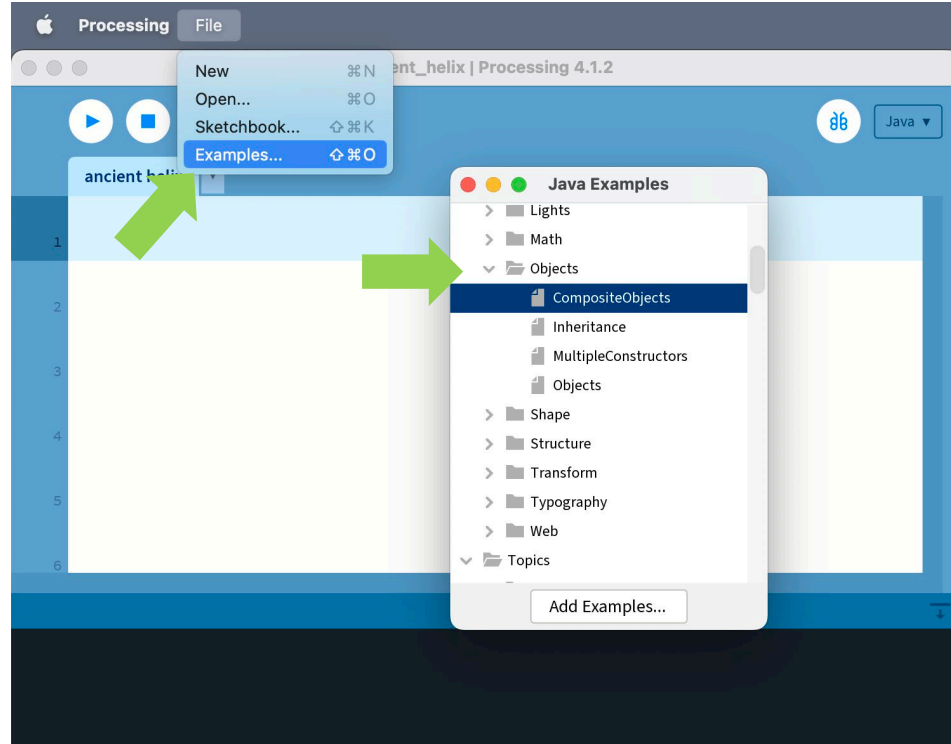
**Objects Tutorial** - <https://processing.org/tutorials/objects>



**Objects Examples** - <https://processing.org/examples>

# Classes Task

- All these examples are in Processing!
- Save example to a new folder then run.
- There's warm up tasks to familiarise yourself with the code, then some class-based challenges
- Consider pair programming, try different challenges across your team
- Feel free to play around and go beyond the challenges.



# Composite Objects

## Warming up:

- Draw a face on the egg using **ellipse()**, **rect()**, **line()**, **arc()** etc. Tip: place these in the Egg Class 'display' function, consider where to add them to the function's sequence of drawing. Don't forget colour with **fill()** and **stroke()**

## Challenges:

- Create a new class and add it to the composite EggRing class. Consider starting with something simple like placing the egg on a podium (or a hat?).

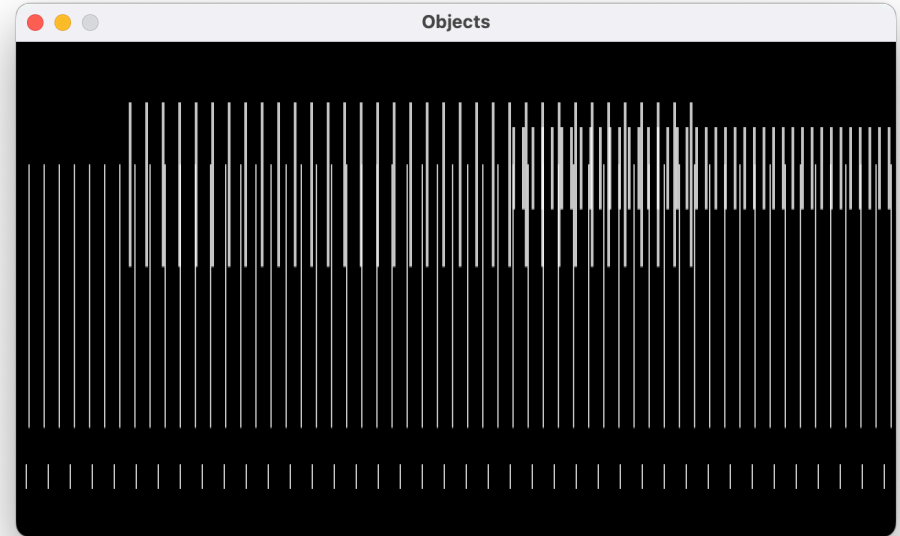
# Objects

## Warming up:

- Randomise the colour of each set of lines.
- Change the lines to circles.

## Challenges:

- Rather than have 4 MRect variables, try creating an array of MRect.
- Create a new 'update' function within MRect that reduces the number of bars by one. Try calling this function on every mouse press.



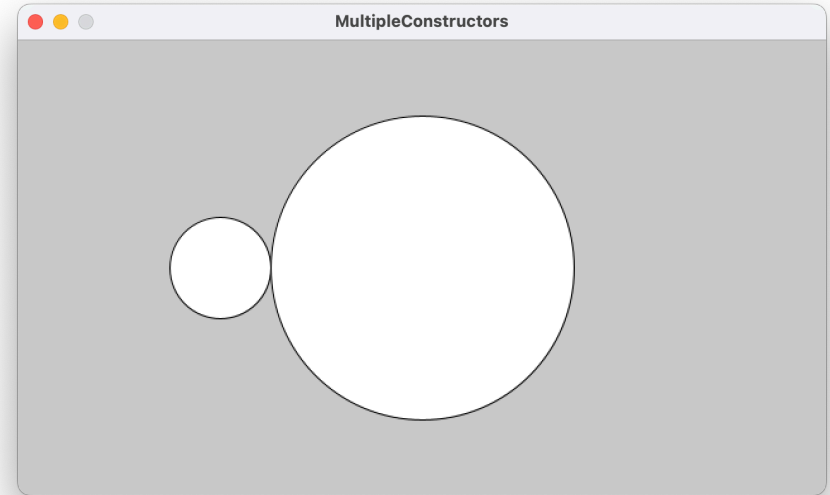
# Multiple Constructors

## Warming up:

- Comment out `noLoop()` so that the sketch loops. Add `background(200)` into draw loop to clear background each frame. Create a `mousePressed()` function to change the x, y and radius of `sp2` with every mouse click (random or mouse pos)

## Challenges:

- Add a fourth argument of your choice to the second constructor, perhaps a Boolean determining whether the circle is filled, or an opacity value. Update the code to make use of this fourth argument.





# Inheritance

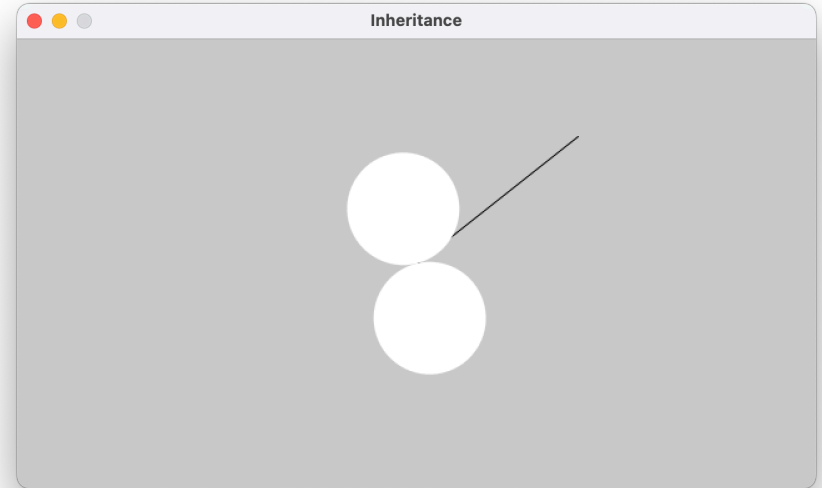
## Warming up:

- Increase the speed of both the spin arm and spin spots (from the superclass)
- Update the Spin superclass update method so that the spinning gradually slows down

## Challenges:

- Create a new subclass that draws a stationary rectangle. Use the angle variable of the superclass to change the colour or width of the rectangle
- Create a new superclass 'Bounce' that enables bouncing rather than spinning

<https://processing.org/examples/inheritance.html>



# homework / groupwork

- Finish working through the examples in your team
- Your team should now have one game idea.
- Draw up a class diagram for your game, add it to your repo
- Begin basic implementation of your classes (whilst keeping a Minimum Viable Product in mind)

