Object Orientated Design

Workshop 5

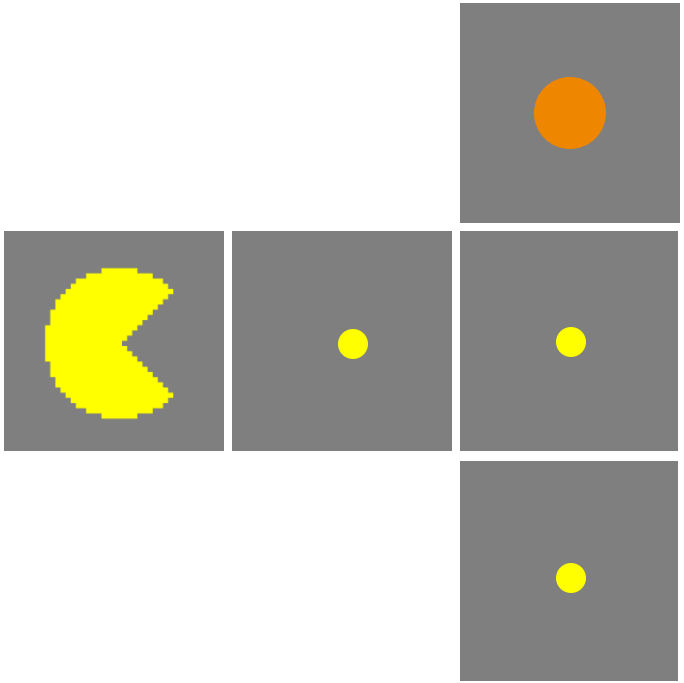
Ruzanna Chitchyan, Jon Bird, Pete Bennett
TAs: Alex Elwood, Alex Cockrean, Casper Wang

Today's Workshop

- Pac-man case study (45mins)
- Then either:
  - Classes challenges in Processing (45mins)
  - Draw class diagram for your game (45mins)
- Homework. Draw up a class diagram for your game, add it to your repo and begin basic implementation.

(Simplified) Pac-man Game

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



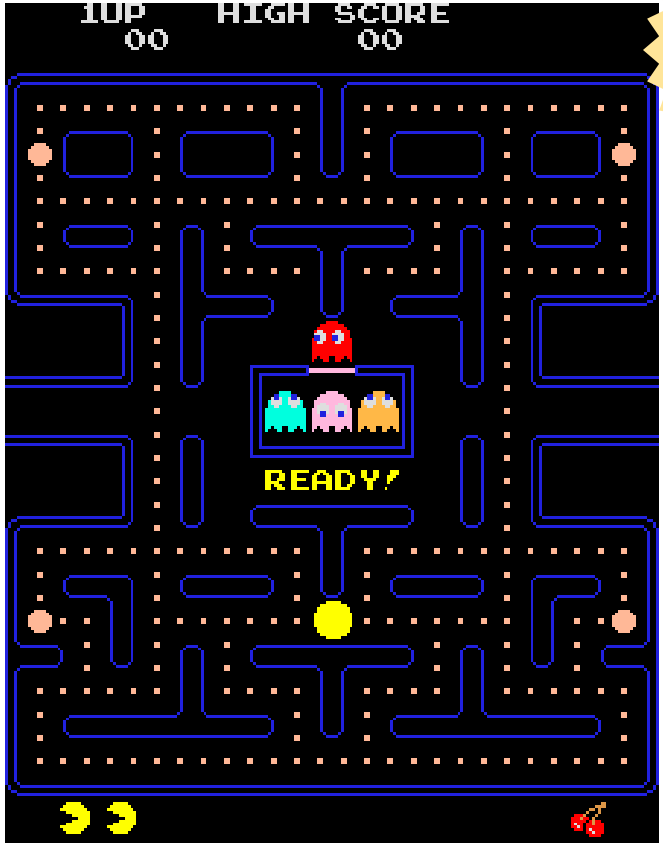
Class Diagram Task

15 mins

Sketch a class diagram of simplified pacman, making sure to note:

- Classes
- Associations
- Inheritance
- Cardinalities

Feel free to note attributes and methods, if you want.

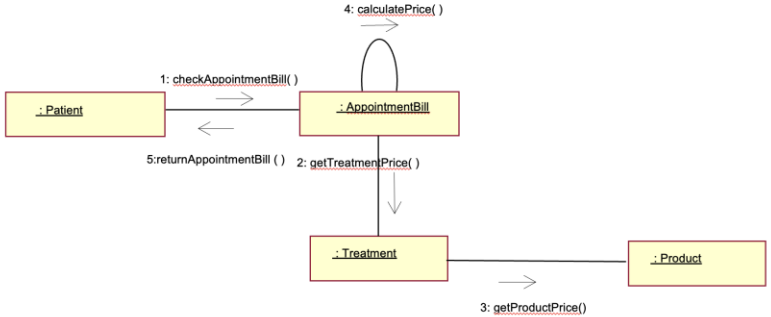


https://en.wikipedia.org/wiki/Pac-Man

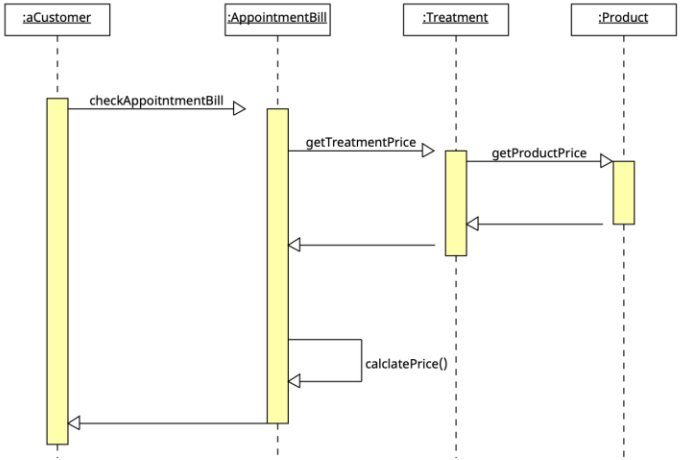
Communication/Sequence Task

15 mins

Sketch a **communication** or a **sequence** diagram for simplified pac-man



communication diagram

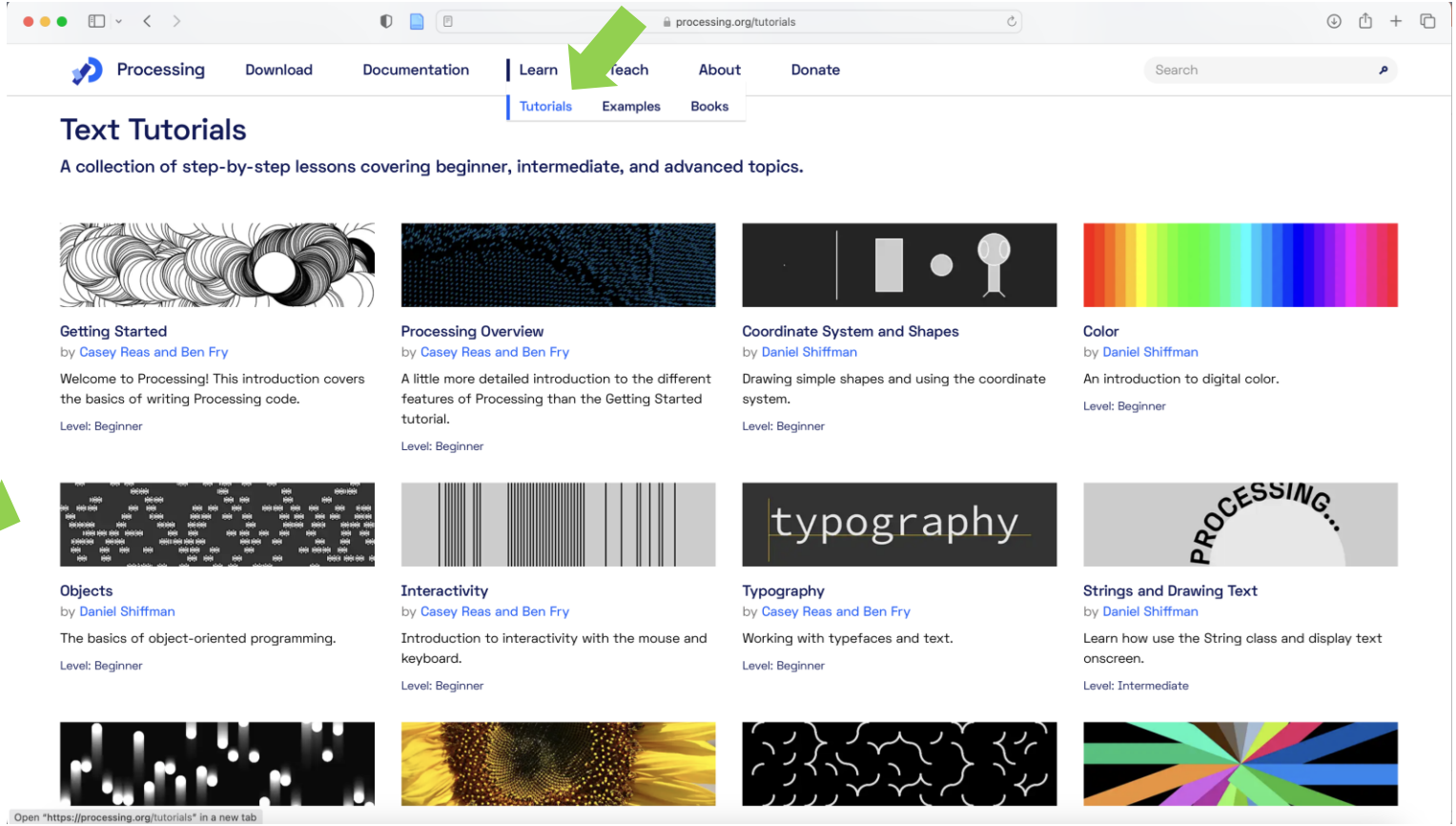


sequence diagram

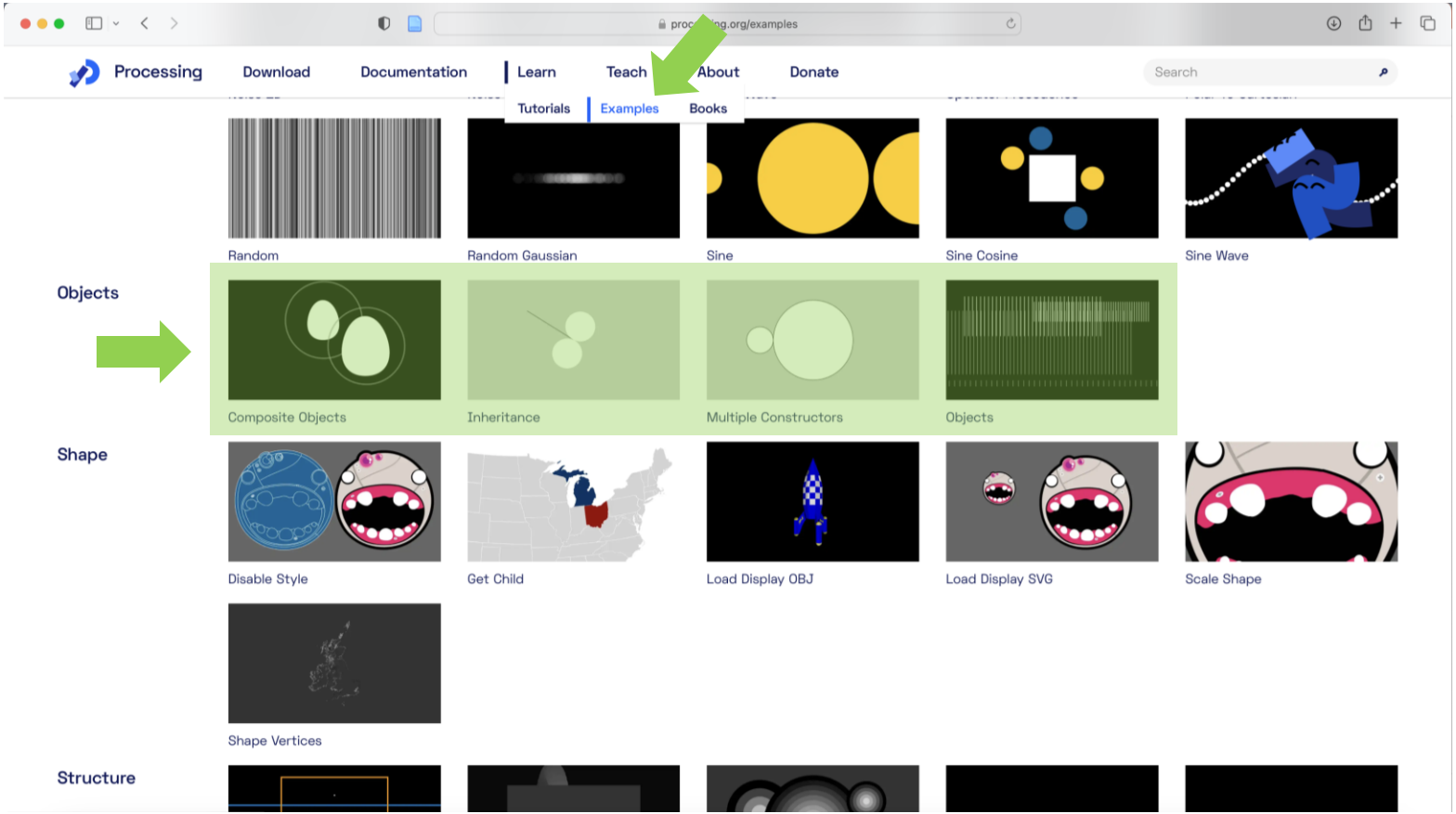
45 mins

Classes in Processing

Optional section if you would like to play around with using classes in Processing with TA support. If you do choose to do this, then please complete the class diagram for your game as homework.



Objects Tutorial - https://processing.org/tutorials/objects



Objects Examples - https://processing.org/examples

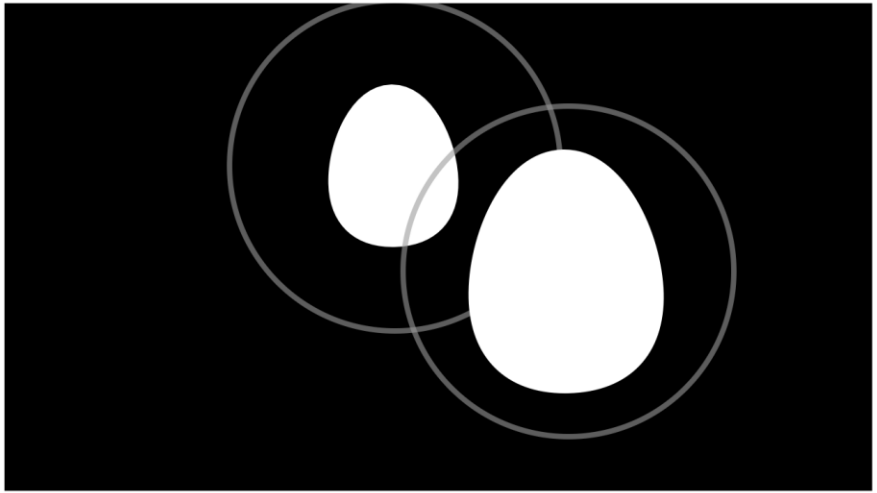
Composite Objects

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 forgot 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?).



“An object can include other objects.”

https://processing.org/examples/compositeobjects.html

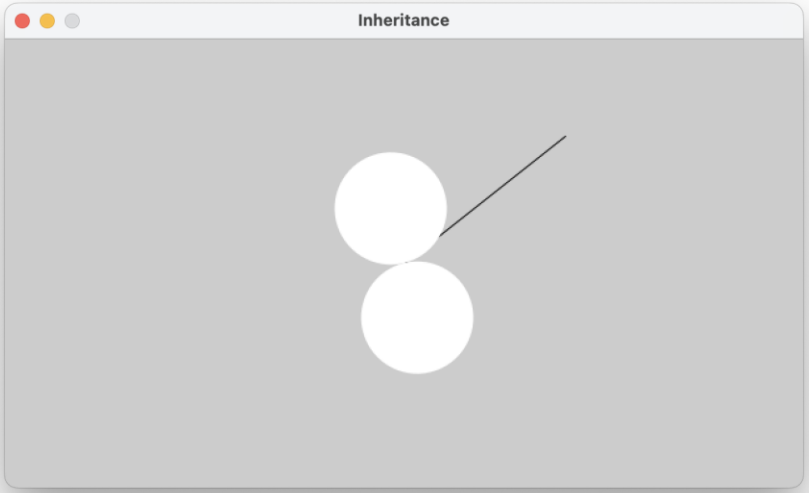
Inheritance

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



“A class can be defined using another class as a foundation.”

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

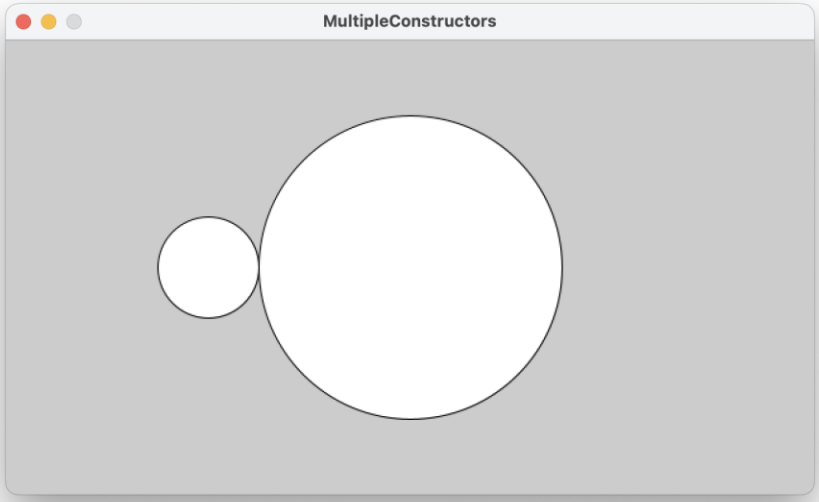
Multiple Constructors

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.



"A class can have multiple constructors that assign the fields in different ways."

https://processing.org/examples/multipleconstructors.html

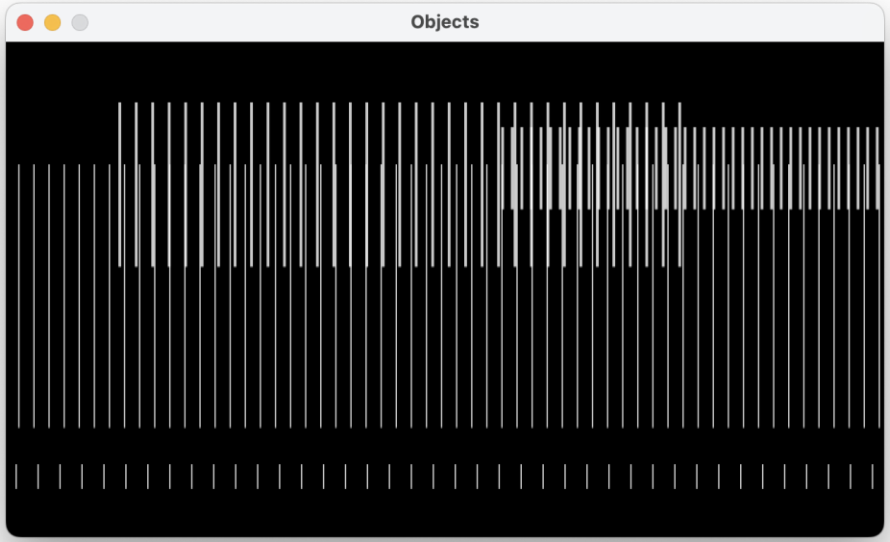
Objects

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.



“Move the cursor across the image to change the speed and positions of the geometry.”

https://processing.org/examples/objects.html

Classes for your game

45 mins

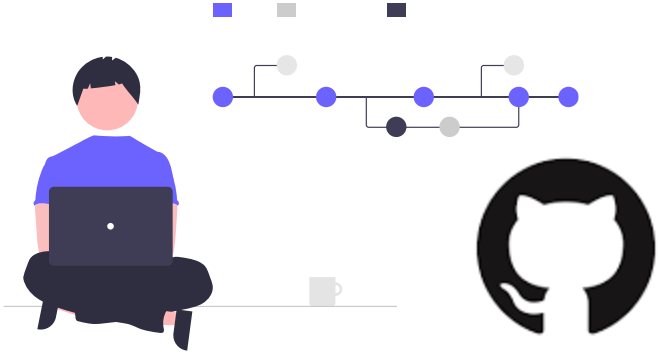
- Create a class diagram for your game making sure to note:
  - Classes
  - Associations
  - Inheritance
  - Cardinalities

Feel free to note attributes and methods, if you want.

- All teams need to do this before next week, please complete as homework and upload to your Github repo.

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)



Summer Project Drop-in

- **Tomorrow!**
  Tuesday 20th February 1pm – 2pm,
  Ivy Gate G01

- Come along to discuss group formation, individual supervisor choice, find team-mates or if you have general questions.

- Please confirm supervisor or submit group choices by **end of reading week**.