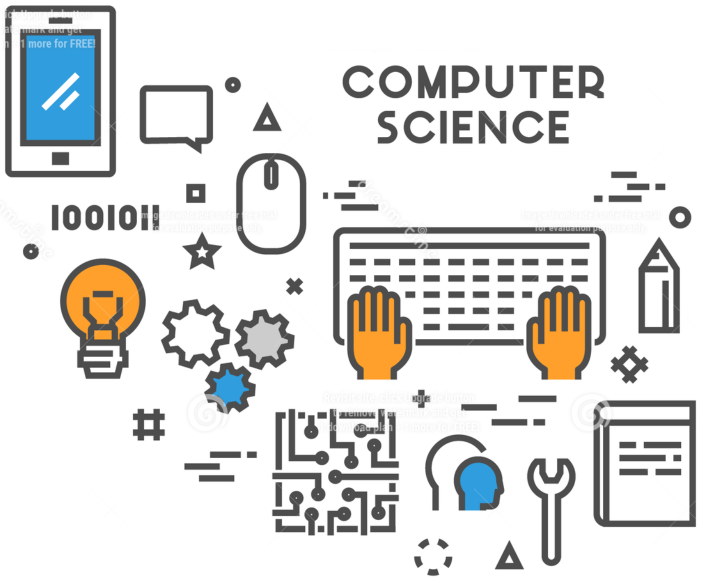
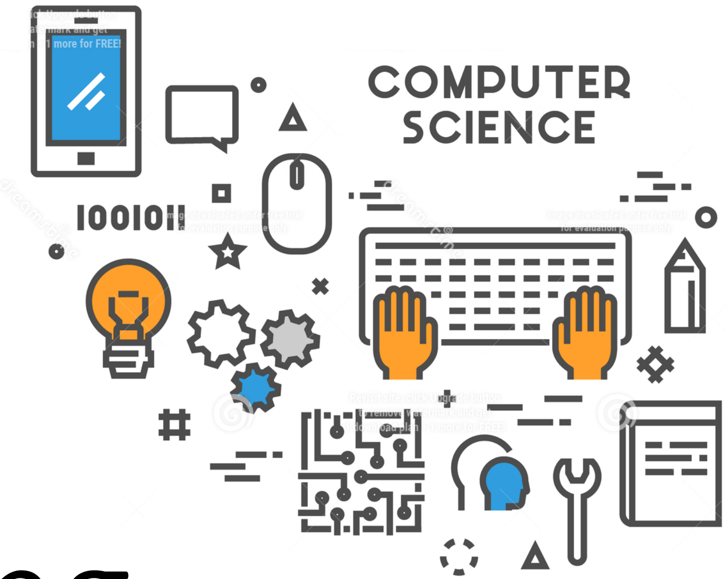


https://spoti.fi/3nOCYMJ

Music for todays challenges!



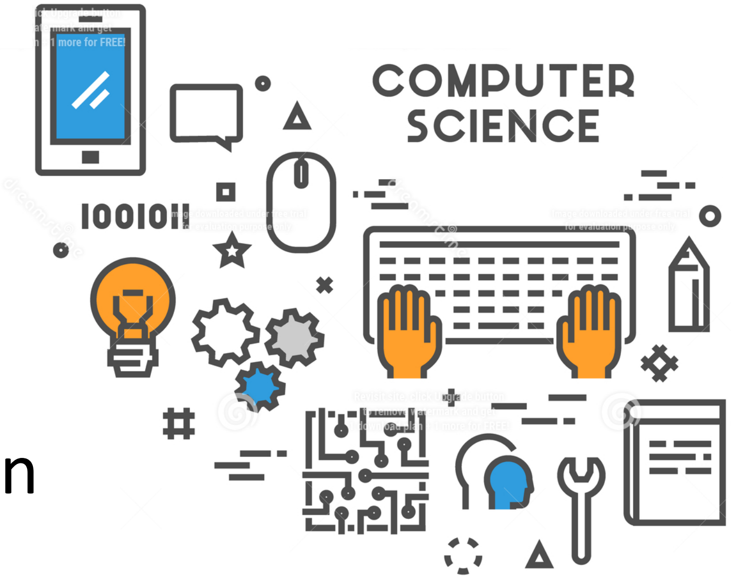
Scientific Computing

Seminar 3 – Week 3 – Fall 2020

Maja Hanne Kirkeby – Computer Science

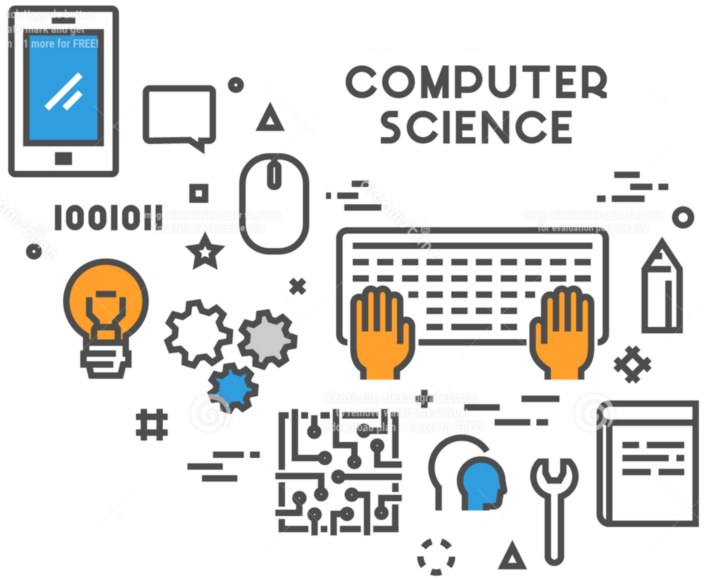
Ulf Rørbæk Pedersen – Physics

Code-along while loop



- Make the ball move across the screen with the pixel-velocity per timestep and print out the position and the timestep.
- Fill the background with white
- What is the greatest pixel-velocity that does not move the ball?

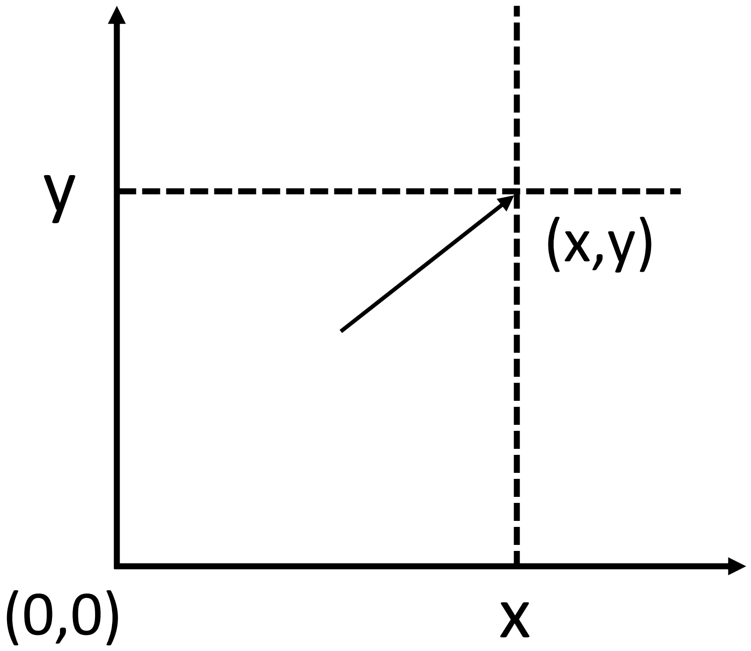
Goals and To-Do



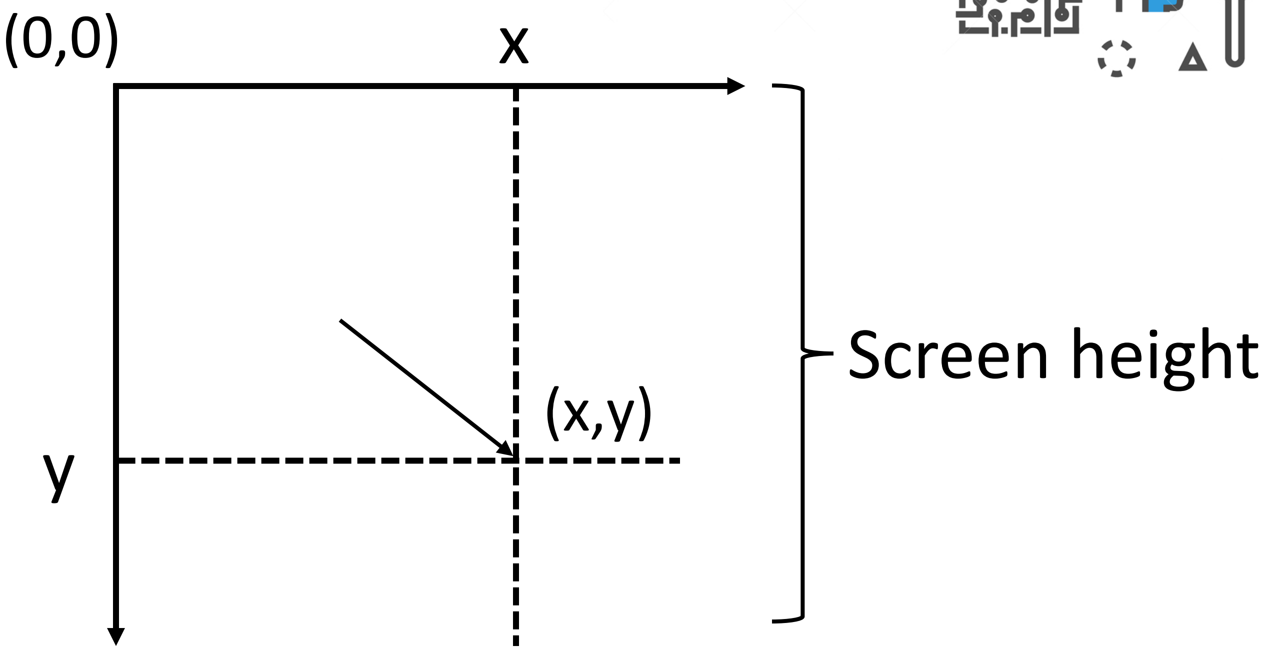
- Install and run PyCharm
- How does Pygame work:
  - The game loop
  - The coordinate system + Drawing in Pygame
  - User interaction via Pygame
- How to transfer a Scientific Calculation to a Simulation in Pygame?
- How can we ensure “correctness”?

Drawing in Pygames

Normal coordinate system



Screen coordinate system

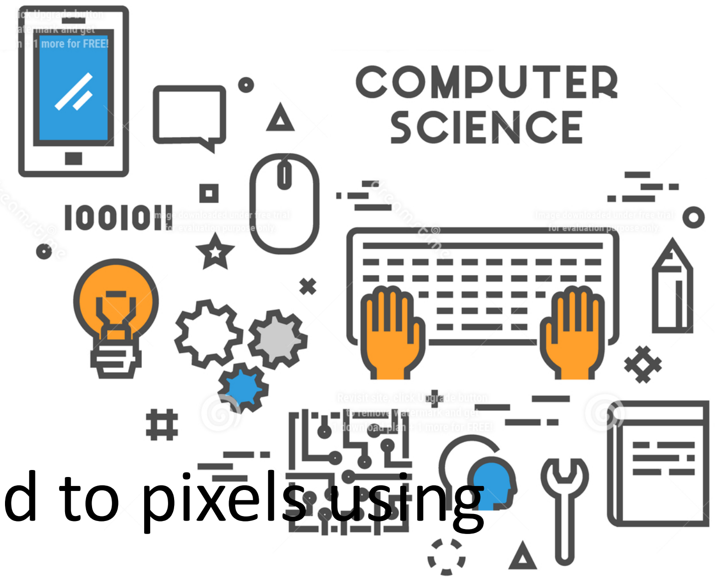


Challenge - Make the ball move

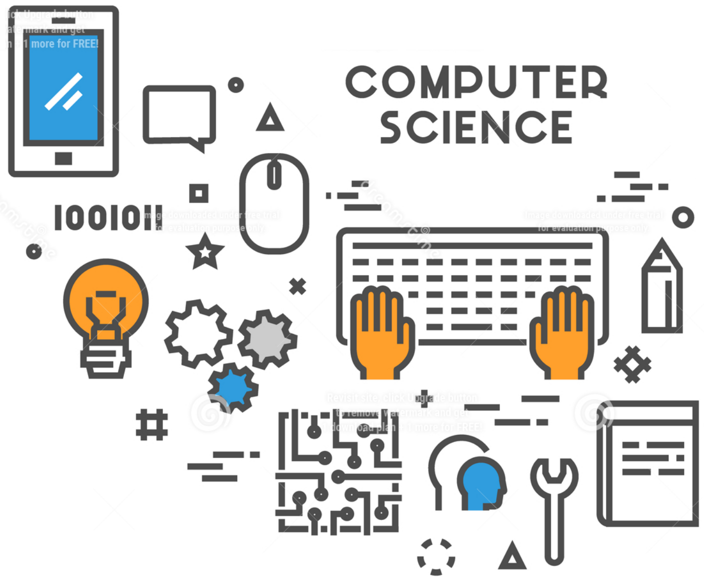
Open moving_ball.py

1. Update the following code so that the current time is updated with the length of the timestep (dt), ball's position (pixels) is updated using the velocity (in pixels)
2. insert a black background so that it looks like the ball is moving
3. what is the greatest velocity that does not move the ball?

Code-along Make the ball jump



- Open moving_ball.py
- 1. The ball's real-world diameter should be converted to pixels using the scale.
  2. change the convert function to convert from real-world coordinates to pixel coordinates
  3. The program should calculate and updates the ball's position, velocity and acceleration in real-world coordinates,
  # and convert it to a pixel position, such that the ball is drawn on the screen at the correct pixel position.



Events in pygame

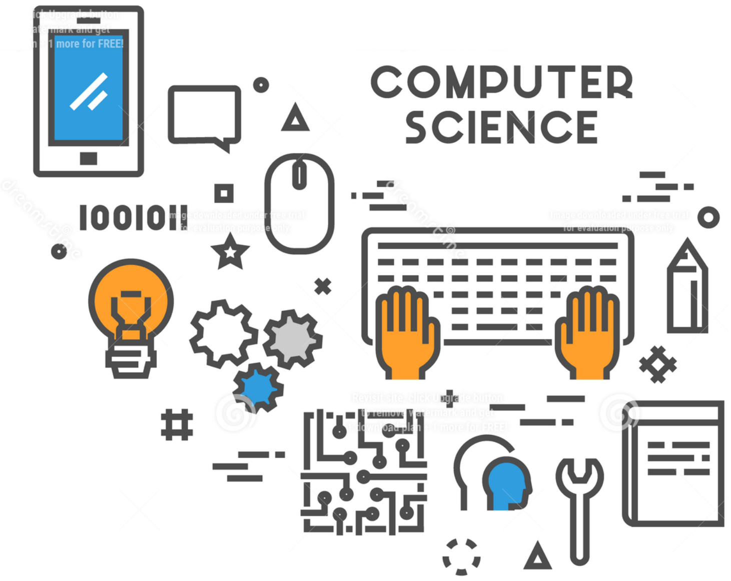
Check out

https://www.pygame.org/docs/

https://riptutorial.com/pygame/example/18046/event-loop

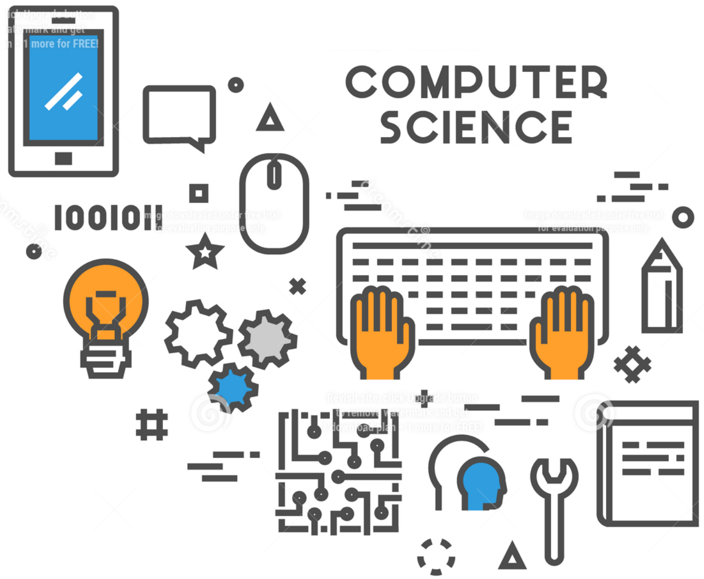
Event type	attributes
QUIT	none
KEYDOWN	key, mod, unicode, scancode
KEYUP	key, mod
MOUSEMOTION	pos, rel, buttons
MOUSEBUTTONUP	pos, button
MOUSEBUTTONDOWN	pos, button

```
if   pygame.event == pygame.QUIT :
     pygame.event == pygame.KEYDOWN
       pygame.quit()
```



https://www.pygame.org/docs/ref/event.html

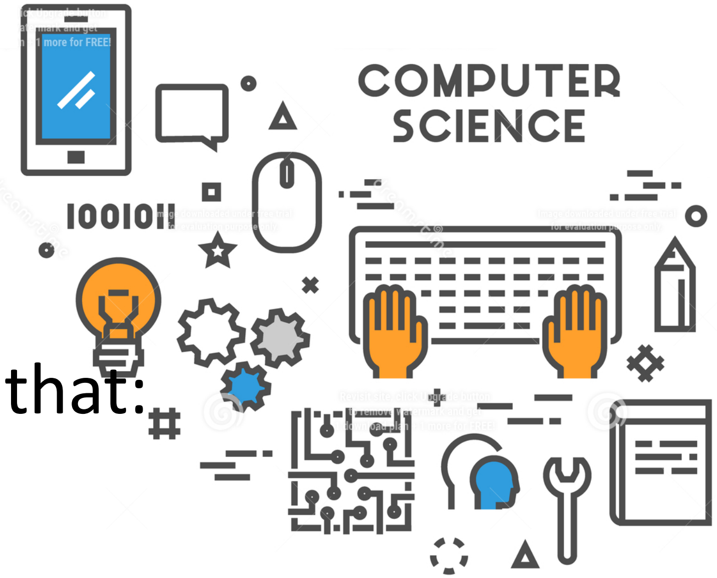
Pygame constant	Description
K_SPACE	space
K_RETURN	return
…	
K_UP	arrow up
K_DOWN	arrow down
…	
K_a	a



```
if  ( pygame.event == pygame.KEYDOWN and
      pygame.key == pygame.K_q):
          pygame.quit()
```

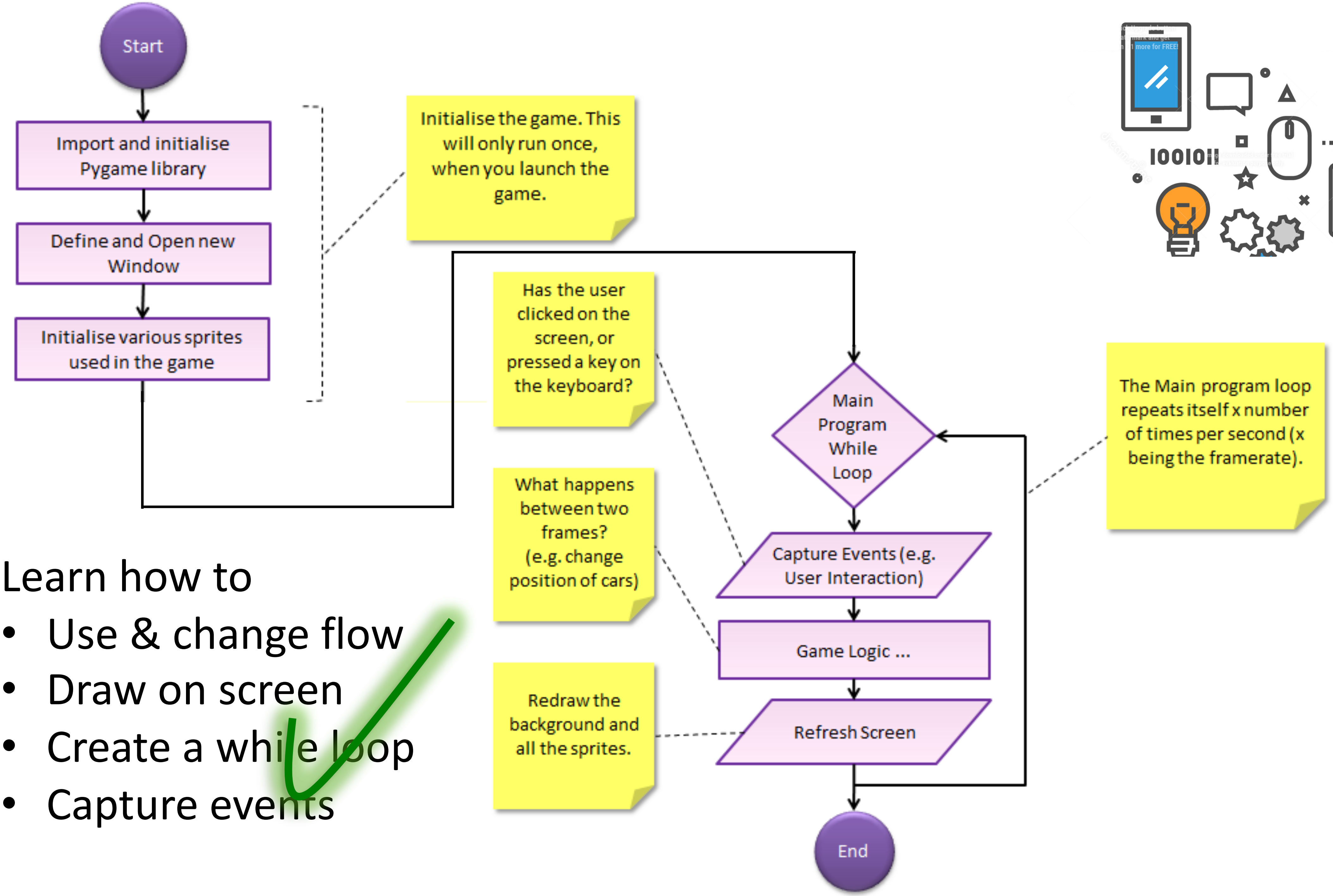
https://www.pygame.org/docs/ref/key.html

Pair-programming



1. Open waiting_ball.py, and change the program so that:
2. The program stops when ‘q’ is pressed
3. If spacebar is pressed the ball should stop/start moving, i.e., moving_ball should change between True and False
4. user should be able to move the ball up and down using arrows (when it doesn't move)
5. if mouse button is pressed, the ball should be moved to the mouse position

COMPUTER SCIENCE



Start

Import and initialise Pygame library

Define and Open new Window

Initialise various sprites used in the game

Initialise the game. This will only run once, when you launch the game.

Has the user clicked on the screen, or pressed a key on the keyboard?

What happens between two frames? (e.g. change position of cars)

Redraw the background and all the sprites.

Main Program While Loop

Capture Events (e.g. User Interaction)

Game Logic ...

Refresh Screen

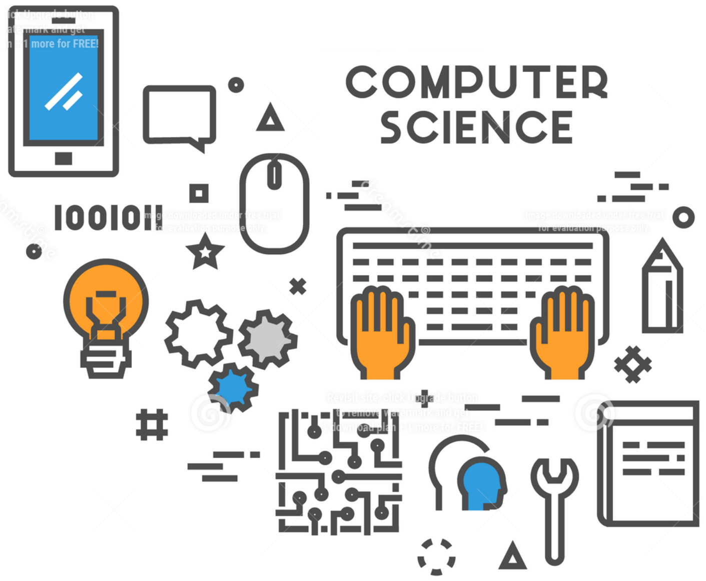
End

The Main program loop repeats itself x number of times per second (x being the framerate).

Learn how to

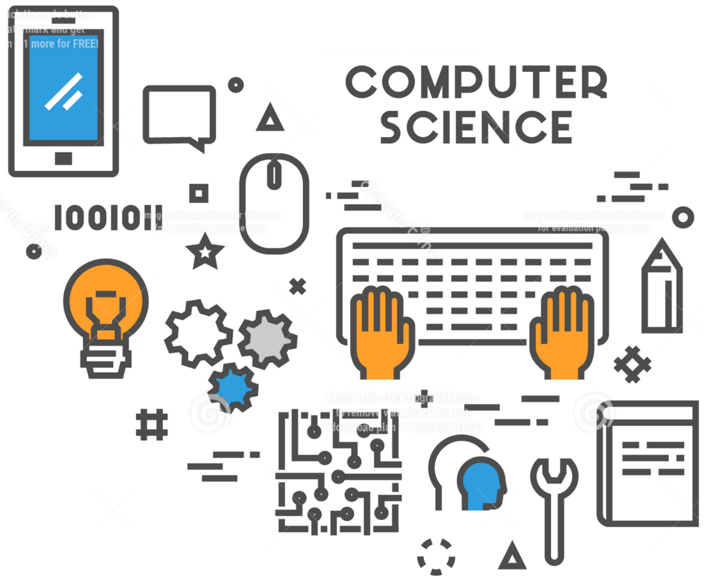
- Use & change flow
- Draw on screen
- Create a while loop
- Capture events

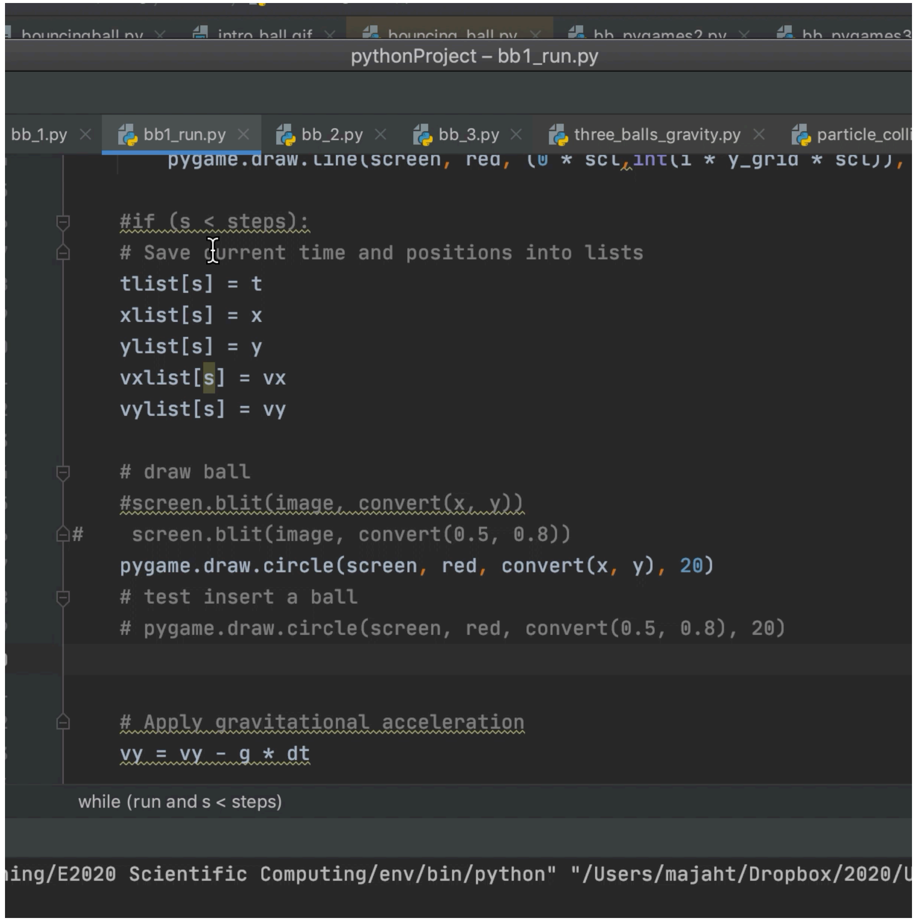
Goals and To-Do



- ✓ Install and run PyCharm
- ✓ How does Pygame work:
  - ✓ The game loop
  - ✓ The coordinate system + Drawing in Pygame
  - ✓ User interaction via Pygame
- ✓ How to transfer a Scientific Calculation to a Simulation in Pygame?
- How can we ensure “correctness”?

Goal





```
        pygame.draw.line(screen, red, (0 * scl, int(i * y_grid * scl)),

    #if (s < steps):
    # Save current time and positions into lists
    tlist[s] = t
    xlist[s] = x
    ylist[s] = y
    vxlist[s] = vx
    vylist[s] = vy

    # draw ball
    #screen.blit(image, convert(x, y))
#    screen.blit(image, convert(0.5, 0.8))
    pygame.draw.circle(screen, red, convert(x, y), 20)
    # test insert a ball
    # pygame.draw.circle(screen, red, convert(0.5, 0.8), 20)


    # Apply gravitational acceleration
    vy = vy - g * dt
```

while (run and s < steps)

```
ing/E2020 Scientific Computing/env/bin/python" "/Users/majaht/Dropbox/2020/U
```