Tutorial 15

- during this tutorial we will
  - understand more about the PGE API
  - start to create a test program `frozenbubble`
  - understand how PGE integrates with PyGame

Initially

- work through this weeks lecture and download the pge source and build it according to the lecture instructions
- see if you can fix the foreground/background draw box (polygon) bug
  - take a copy of `examples/breakout/breakout.py`
  - create a foregound box or polygon at easy to identify coordinates and debug `python/pge.py`

Using PGE to create a game

- full documentation about PGE is available ⟨`http://floppsie.comp.glam.ac.uk/Glamorgan/gaius/pge/homepage.html`⟩

Creating frozenbubble using PGE

- firstly open up a command line terminal and run `frozen-bubble`
- we will try and implement this game using PGE and Python

Creating frozenbubble using PGE

- start with the breakout game source code ⟨`http://floppsie.comp.glam.ac.uk/Glamorgan/gaius/pge/example_games.html`⟩ and adapt it
- throw away all the blue boxes and associated callbacks
- see that the modified game still runs to completion

Frozenbubble

- add some circles at the top and a ball at the bottom of the screen
- make the ball fire using up arrow and use the direction of the mouse pointer
- introduce call backs for bubble collision
- consider what API changes need to be made to implement frozen bubble
- consider how you would implement the bubble colour touching problem

Conclusion

- we have
  - understood more about the PGE API
  - created a small test application which can be used to drive forward changes in PGE