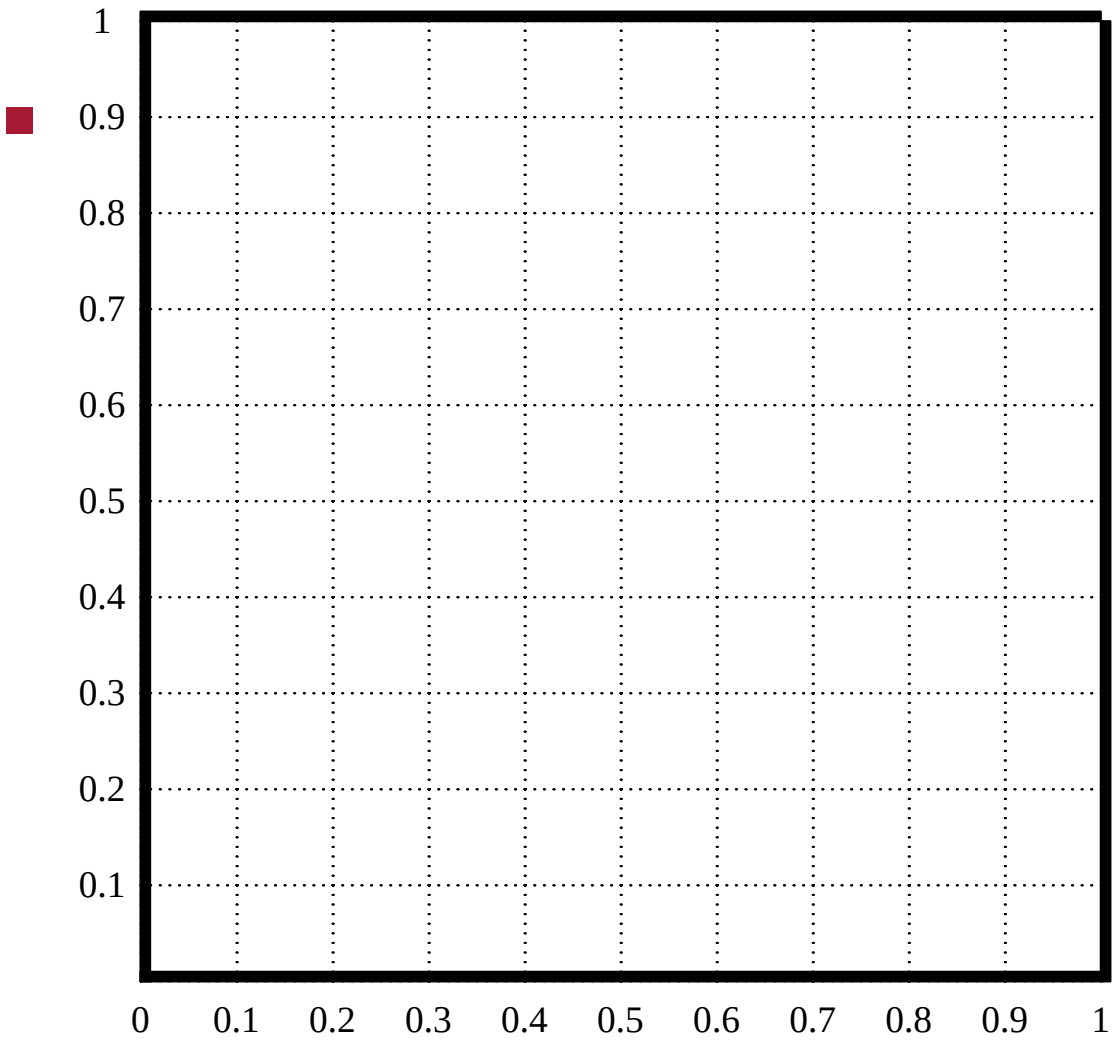
Tutorial 13

- during this tutorial we will
  - understand the PGE API
  - create a small snooker game using PGE
  - understand how PGE integrates with PyGame

Using PGE to create a game

- full documentation about PGE is available ⟨`http://floppsie.comp.glam.ac.uk/Glamorgan/gaius/pge/homepage.html`⟩
- any PGE game or simulation must fix a board frame around our universe

Using PGE to create a game



Framing the universe in PGE

```
def placeBoarders (thickness, color):
    bottom = pge.box (0.0, 0.0, 1.0, thickness, color).fix ()
    left = pge.box (0.0, 0.0, thickness, 1.0, color).fix ()
    right = pge.box (1.0-thickness, 0.0, thickness, 1.0, color)\
           .fix ()
    top = pge.box (0.0, 1.0-thickness, 1.0, thickness, color)\
           .fix ()
    return [bottom, left, right, top]
```

- which can be invoked by:

```
sides = placeBoarders (0.01, brown)
```

Run breakout

- examine the breakout game source code ⟨`http://floppsie.comp.glam.ac.uk/Glamorgan/gaius/pge/example_games.html`⟩
  - take a copy of this code and run it from the command line using the Python interpreter

- read the code and mentally make a note about the various sections of code and what they are performing

Implement snooker in PGE

- now create a new simple PGE program which just creates the boarders (you will need a `main` which can be borrowed from the breakout example)

- now take a copy and introduce a single moving circle (cue ball) which is dropped from the point `0.5,0.5`
  - gravity must be turned on!

Implement snooker in PGE

- now turn gravity off and give this circle an impulse upwards (see the breakout code for a working example of an impulse)
  - check the PGE documentation for more details

- add some snooker ‘‘red’s’’
  - observe what happens when the cue ball hits the red balls

Implement snooker in PGE

- now add a background green to the PGE world
- now add some pockets
- implement some callbacks for the pockets
  - to delete a ‘‘red’’
  - reposition the cue ball
  - and calculate the ‘‘break’’ value
- continue to work on this program as part of some directed learning throughout the week

Conclusion

- we have
  - understood more about the PGE API
  - created a small snooker game using PGE
  - understood how PGE integrates with PyGame

- you might want to read around the subject by reading this paper The Construction of a Predictive Collision 2D Game Engine ⟨`../../../Papers/paper21/ieee.pdf`⟩