COMP140 - GAM160

Session 1 – Object Orientated Programming

0000 - 0005	Introduction
0005 - 0030	Brief Lecture
0030 - 0100	Activity - Maintainable Code
0100 - 0130	Coffee Break
0130 - 0200	Activity - Single Responsibility Principle
0200 - 0250	Activity - Alternative Controllers
0250 - 0300	Debrief

Before We Begin: Module Pace

- COMP140-GAM160 has two assessments:
  - Game Project
  - Hardware Project
- Slightly different weightings for BA & BSc, see brief on Learning Space
- Because the second assessment is a continuation of the first, the module is front-loaded with quite a steep learning curve to set you up with the concepts that you need for these over the first 4-5 weeks.
- The pace and complexity of the material might seem high for the first few sessions but then it settles down!

Before We Begin: Key Dates

- As per University Policy all submission dates are on the Learning Space
- But!!!
  - We have a formative assessment submission on **Week 3** & **Week 5**
- **Please keep your eye on learning space and email!**

23-Jan-17	26	1	Study Block 2 (11 weeks)
30-Jan-17	27	2	
06-Feb-17	28	3	
13-Feb-17	29	4	
20-Feb-17	30	5	
27-Feb-17	31	6	
06-Mar-17	32	7	
13-Mar-17	33	8	
20-Mar-17	34	9	
27-Mar-17	35	10	
03-Apr-17	36	11	
10-Apr-17	37		Easter
17-Apr-17	38		EM
24-Apr-17	39	12	Study Block 2 (1 week)
01-May-17	40	13	Assessment
08-May-17	41	14	Assessment / Mark Submission
15-May-17	42	15	Assessment Boards / Feedback
22-May-17	43	16	Assessment Boards / Feedback

Before We Begin: Study Advice

Practice, practice, practice!

- Don’t be afraid of making mistakes, they are an excellent way of developing your experience.
- Malcolm Gladwell’s book, Outliers, states that it takes 10,000 hours to become an expert in any topic. How many hours of programming experience do you have? How many do you want to have by graduation?
- Please ask questions! There are not any daft questions when it comes to programming

Before We Begin: Study Advice

Research!

- We try to direct your learning in an appropriate sequence, but it is likely that your will need to view the material phrased several different ways before it gels for you. So read other people's views.
- Google is your friend!
  - Read Unity forums, stackoverflow, dotnetpearls, reddit, etc…
  - Watch youtube tutorials
- Refer to the module reading list on the learning space
- Use your lynda.com account

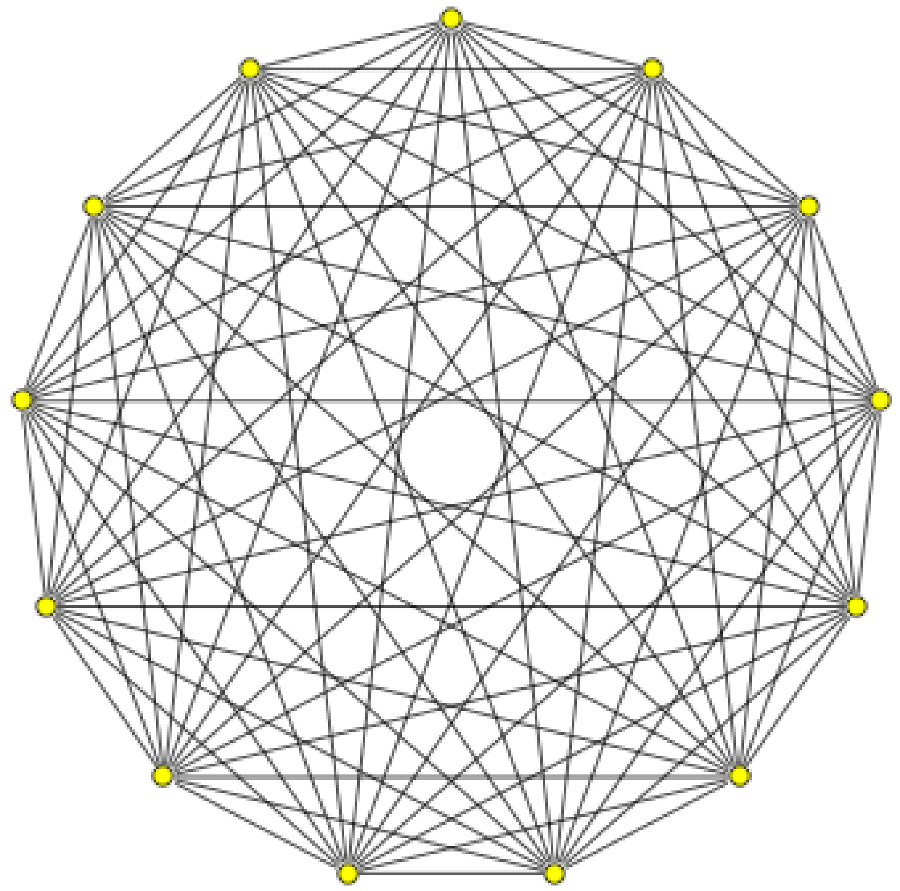
Research Task

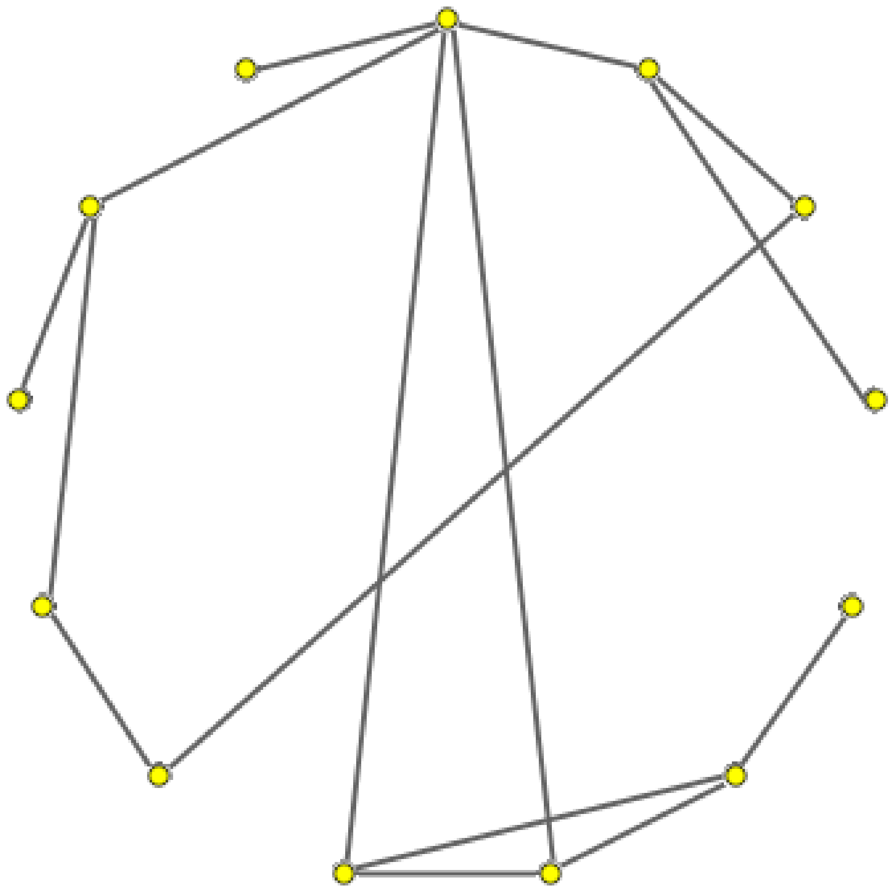
- Spend a few minutes researching ‘Maintainable Code’
    - Make notes as you go
- Group discussion:
    - What does the term mean?
    - How can we apply it to game development ?

Maintainable Code

- Choose appropriate names
  - Consider your classes, variables, methods, etc.
- Write clear comments
  - The code tells you how, the comments tell you why
- Reduce dependencies
  - i.e. references to other classes that are required for your code to compile
- Write less code
  - Code that doesn’t exist cannot have bugs. Remove all extraneous code!

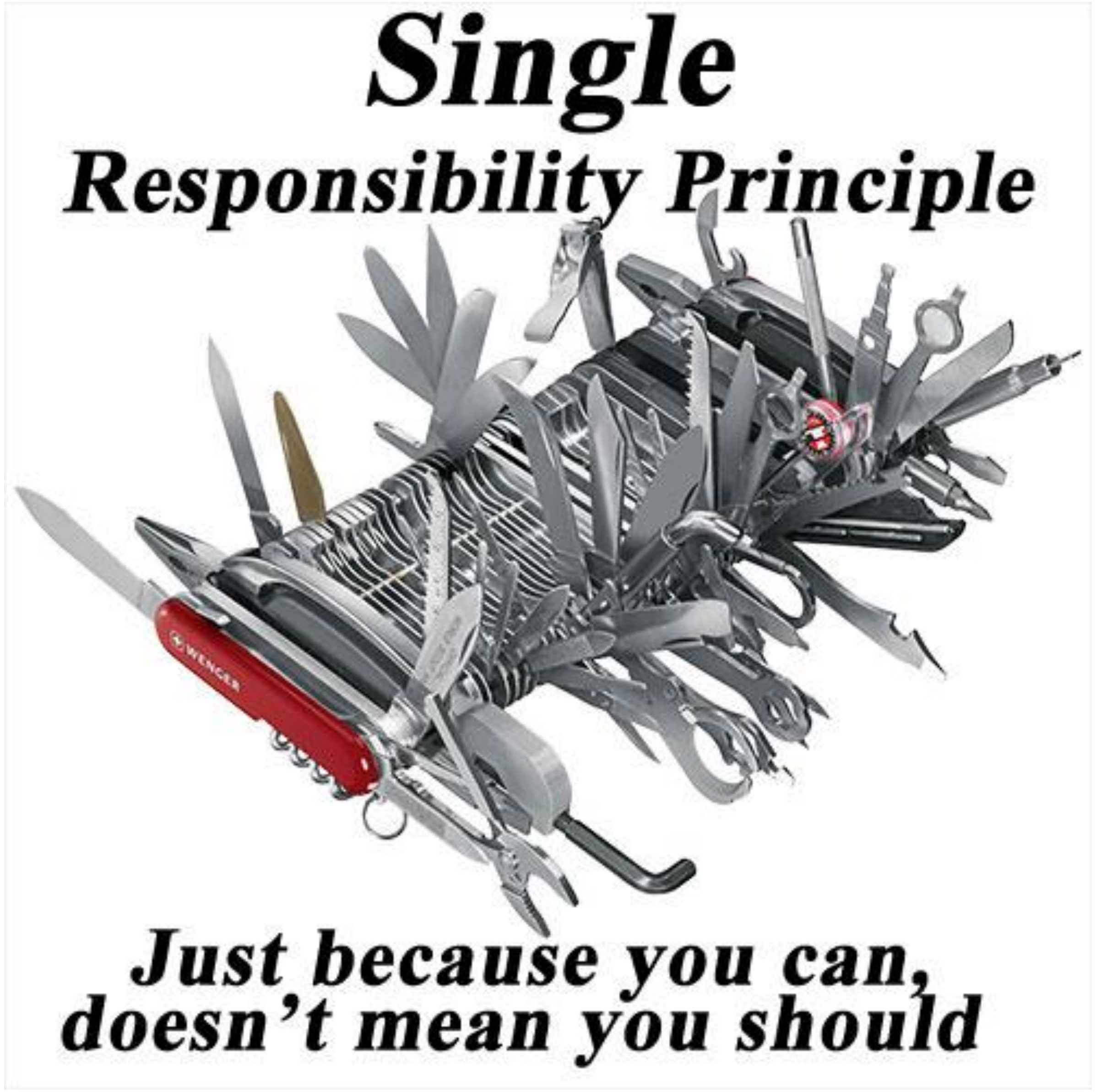
Dependencies





Research Task

- Spend a few minutes researching the 'Single Responsibility Principle'
    - Make notes as you go
- Group discussion:
    - What does the term mean?
    - Try to provide an example describing code that you have written in the past which would have benefitted from separating a large class into single responsibilities.


Single
Responsibility Principle
Just because you can,
doesn't mean you should

Single Responsibility

To give an example of this principle, let’s say we have a Player class. Player handles input, physics, weapons, health, and inventory. That’s a rather long list of things the Player is responsible for. We should break this into multiple different components that each do just one thing.

- PlayerInput – responsible for handling input
- PlayerPhysics – responsible for driving physics simulation
- WeaponManager – responsible for managing player weapons
- Health – responsible for player health
- PlayerInventory – responsible for player inventory

(From: http://bit.ly/210O96X)

Workshop Task

- The other goal of this module is to create custom controllers
- Break into groups of 3 or 4
- Each group member look at a different game that uses an alternative controller
  - https://twitter.com/ShakeThatButton
- Group Discussion:
  - What kind of interactions does this game have?
  - How is this linked to the game design?
  - What kind of hardware is used?
- Each group should share their favourite game with the class