

COMP140-GAM160: Creative Computing: Hacking &
Further Game Programming

1: Introduction to Object Orientated Programming

Learning outcomes

- ▶ **Understand** Object Orientated principles
- ▶ **Understand** the Single Responsibility principle
- ▶ **Implement** a simple class hierarchy

Assignments

Assignment 1

- ▶ The first assignment is all about designing a custom controller and a game
- ▶ You will also develop a version of Pong which uses a custom controller
- ▶ You will have series of worksheets which are part of the course work

Assignment 2

- ▶ You will build the controller and game
- ▶ The design of your game can change slightly
- ▶ For the controller, there is an expectation that you will buy some additional components
- ▶ Finally, you will lose marks if you submit a controller which is a bare breadboard
- ▶ We expect for you to build something or embedded the controller into an item

Examples from Last Session

- ▶ Safe - <https://youtu.be/X4wB3AakSvA>
- ▶ Tank - <https://youtu.be/AL3LrcRskig>
- ▶ Skateboard - <https://youtu.be/Wj4Eb0yUejE>
- ▶ Powerglove - <https://youtu.be/dp9xM55eZUM>
- ▶ Snooker - <https://youtu.be/4XFZ4PMoPTE>

Alt-Controller

Notable Alt-Controller Games

- ▶ Steel Battalion -
<https://www.youtube.com/watch?v=rGgxRsaGdcA>
- ▶ Deep VR - <https://www.polygon.com/2015/3/2/8133675/deep-vr-meditation>
- ▶ Space Box -
https://www.gamasutra.com/view/news/290700/ALTCTRLGDC_Showcase_Spacebox.php
- ▶ Line Wobbler -
<http://wobblylabs.com/projects/wobbler>
- ▶ GDC Alt-Ctrl 2017 Roundup
<https://www.youtube.com/watch?v=IoqAJ7ynuhw>
- ▶ Nintendo Labo - <https://www.nintendo.co.uk/Nintendo-Labo/Nintendo-Labo-1328637.html>

Object Orientated Principles

Classes and Objects

- ▶ Most programming languages have a pre-defined set of data types (int, float, bool etc)
- ▶ We can add our own data types by declaring and defining Classes
- ▶ Classes are a collection of data and functions which operate on the data
- ▶ We can then use these classes like any built-in data type

Class Examples - C++

```
class Player
{
public:
    Player()
    {
        Health=100;
    };

    void TakeDamage(int health)
    {
        Health-=health;
    };

    void HealDamage(int health)
    {
        Health+=health;
    };

    ~Player(){};
private:
    int Health;
};
```

Naming Tips

- ▶ Classnames should be proper nouns (e.g. Player, Enemy, Orc, Goblin, etc)
- ▶ Function names should be verbs (e.g Attack, TakeDamage, SetName, etc)
- ▶ All names should be descriptive
- ▶ Comments! You should add comments before each function!

Coding Standards

- ▶ UE4 Coding Standards -
`https://docs.unrealengine.com/en-us/Programming/Development/CodingStandard`
- ▶ Google C++ Coding Standards - `https://google.github.io/styleguide/cppguide.html`

Single Responsibility Principle

Intro

- ▶ This is a principle from Computer Science that a class should have a single responsibility
- ▶ Take for example a Player class which handles input, physics, weapons, health and inventory
- ▶ This should probably be split into several classes (Remember Class naming from previous slide)

- ▶ You have been given a starting project which contains a Player class
- ▶ Carry out the following tasks
 1. Research 'Single Responsibility Principle' (5 - 10 minutes)
 2. Break the Class into several new classes
 3. Post your Classes onto the Slack channel or forum
- ▶ C++ Starter Project Live Coding - <https://github.com/Falmouth-Games-Academy/COMP140-Examples>

Exercises

Activity - Alt-Controller Research

- ▶ Review the Shake that Button Twitter Account - <https://twitter.com/ShakeThatButton> & Past alt.ctrl.GDC entries
- ▶ Pick 3 games and note down the following information in a Google Doc or similar
 - ▶ Name
 - ▶ URL
 - ▶ Screenshot
 - ▶ brief description
 - ▶ What you find interesting about it
- ▶ Do this for the next 30 minutes
- ▶ **Keep this document, it will feed into your coursework!**

Activity - Single Responsibility Principle

- ▶ You have been given a starting project which contains a Player class
- ▶ Carry out the following tasks
 1. Research 'Single Responsibility Principle' (5 - 10 minutes)
 2. Break the Class into several new classes
 3. Post your Classes onto the Slack channel or forum
- ▶ Unity Starter Project -
`https://github.com/Falmouth-Games-Academy/bsc-course-materials/raw/2017-18/COMP140/01/GAM160-Ex1.zip`
- ▶ C++ Starter Project - Live Coding