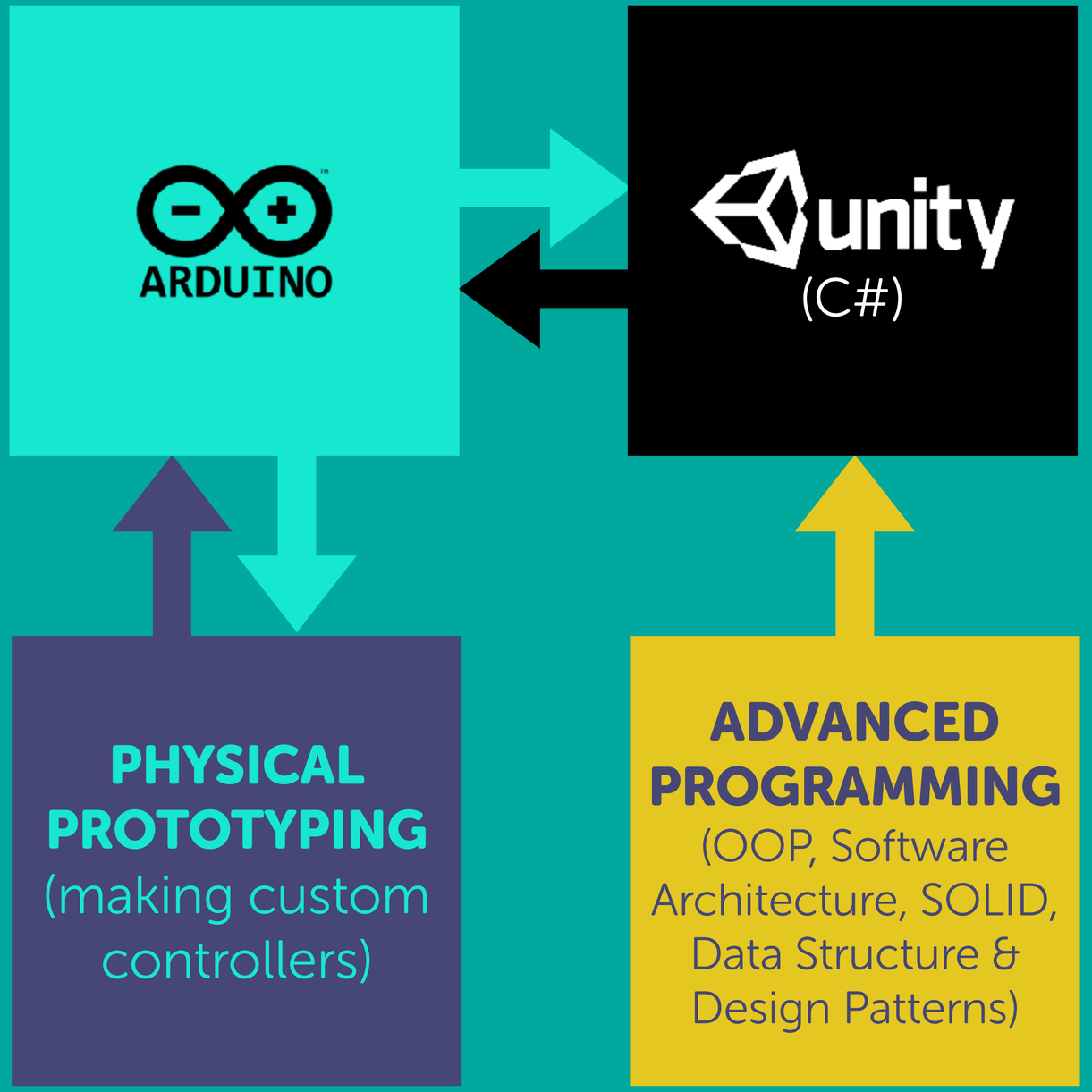
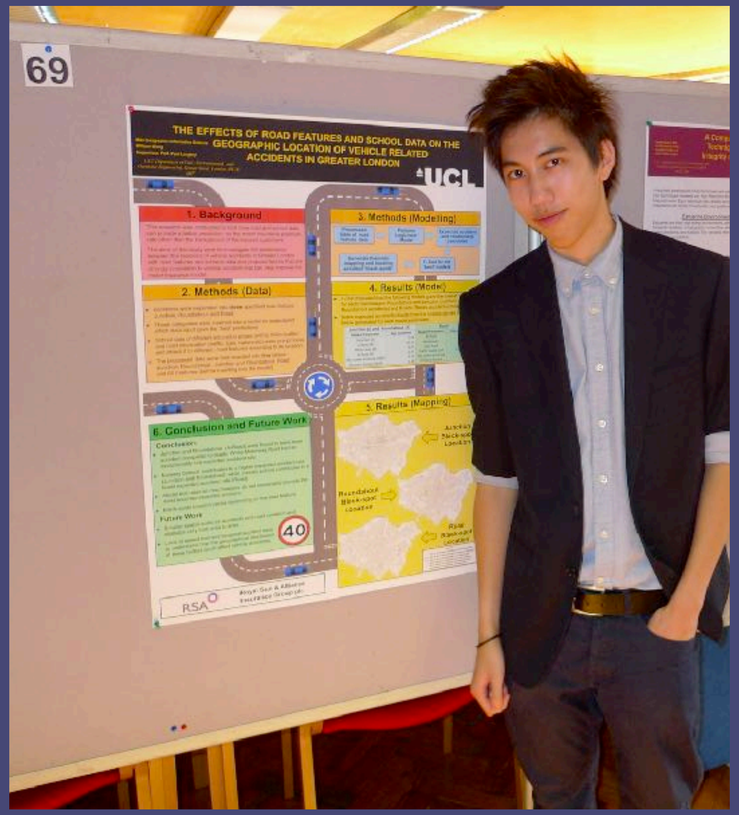
COMP140
Work FLOW



Poster

- Detail the **hardware** of the **control system**
- Detail the **design** of the **control system**
- Detail the **elements** of the **game/experience**
- **UML Diagram** of the **software architecture**

Poster - What is it in practice?



POSTER		
Overview of game and system	**Screenshots** of interface	**Photos** of the control system hardware and/or someone using it.
Pseudo code describing a particular process	**UML** diagrams	**Describe** use of any design patterns, data structures and OOP principles.

COMP140 - Roadmap

Wk 1	Wk 2	Wk 3	Wk 4	Wk 5	Wk 6	Wk 7	Wk 8	Wk 9	Wk 10	Wk 11
Arduino Basics and Electrical Circuits	**OOP Programming** • Classes, Constructors, Member & properties of classes • using Uduino to connect Arduino and Unity	**Software Architecture** • Game loops, • input handlers • Engine architecture. • Handler events from virtual interfaces in Unity to physical actuators in Arduino and vice versa.	**Independent Research and development** (Teaching Staff are interviewing)	**UML** • Understand UML Notation • Resolve the control flow of your game and controller • Define and resolve implementation issues.	**Studio Practice** Focus on developing your Individual computing project, documentation and planning of your process.	**Data Structures** • Exercises to develop a clearer understanding and application of data structures. • Evidence of how you can apply it to your own project.	**Design patterns** General, reusable solutions to a commonly occurring problem within a given context in programming. 3 principle types • Creational • Structural • Behavioural	**Poster Demo** This week you will demonstrate your poster to your peers and a range of academic staff.	**Peer Review** This week you will demonstrate your poster to your peers and a range of academic staff.	**Viva** This week you will demonstrate your poster to your peers and a range of academic staff.

Sessions

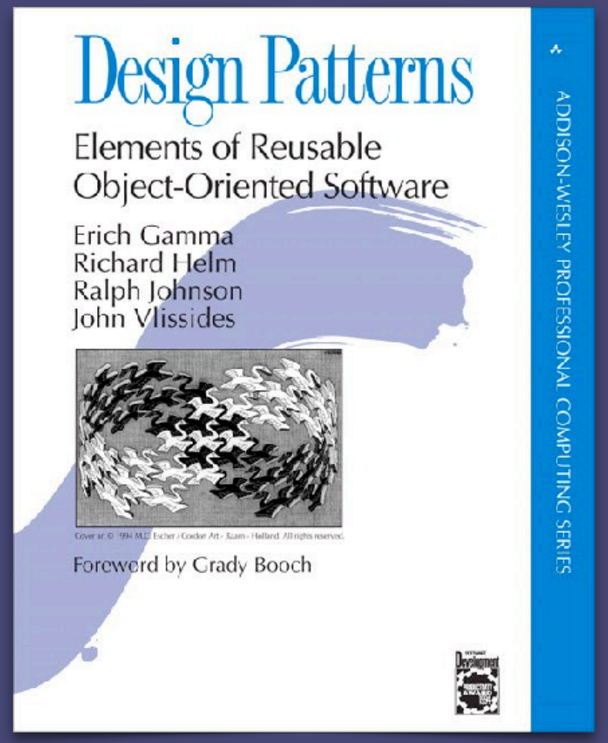
Arduino & Physical Computing Workshop	**Arduino & Physical Computing Workshop**	**Arduino & Physical Computing Workshop**		**Arduino & Physical Computing Workshop**						
	Programming Workshop	**Programming Workshop**		**Programming Workshop**		**Programming Workshop**	**Programming Workshop**			
Group Seminar	**Group Seminar**	**Group Seminar**		**Group Seminar**		**Group Seminar**	**Group Seminar**			

Design Patterns Part 1

COMP 140 - Individual Creative Computing Project

Introduction

- '**Gang of Four**' - Erich Gamma, John Vlissides, Richard Helm & Ralph Johnson
- Design patterns are distilled knowledge of how to design object-oriented software. They are simple and elegant solutions to specific problems
- Design patterns establish consistency that helps developers build and modify code safely avoiding common architecting problems.
- Using consistent methods to fix and to avoid issues.
- Having a shared language to understand common problems
- Adhering to the solid principles of object-oriented programming.



Learning Outcomes

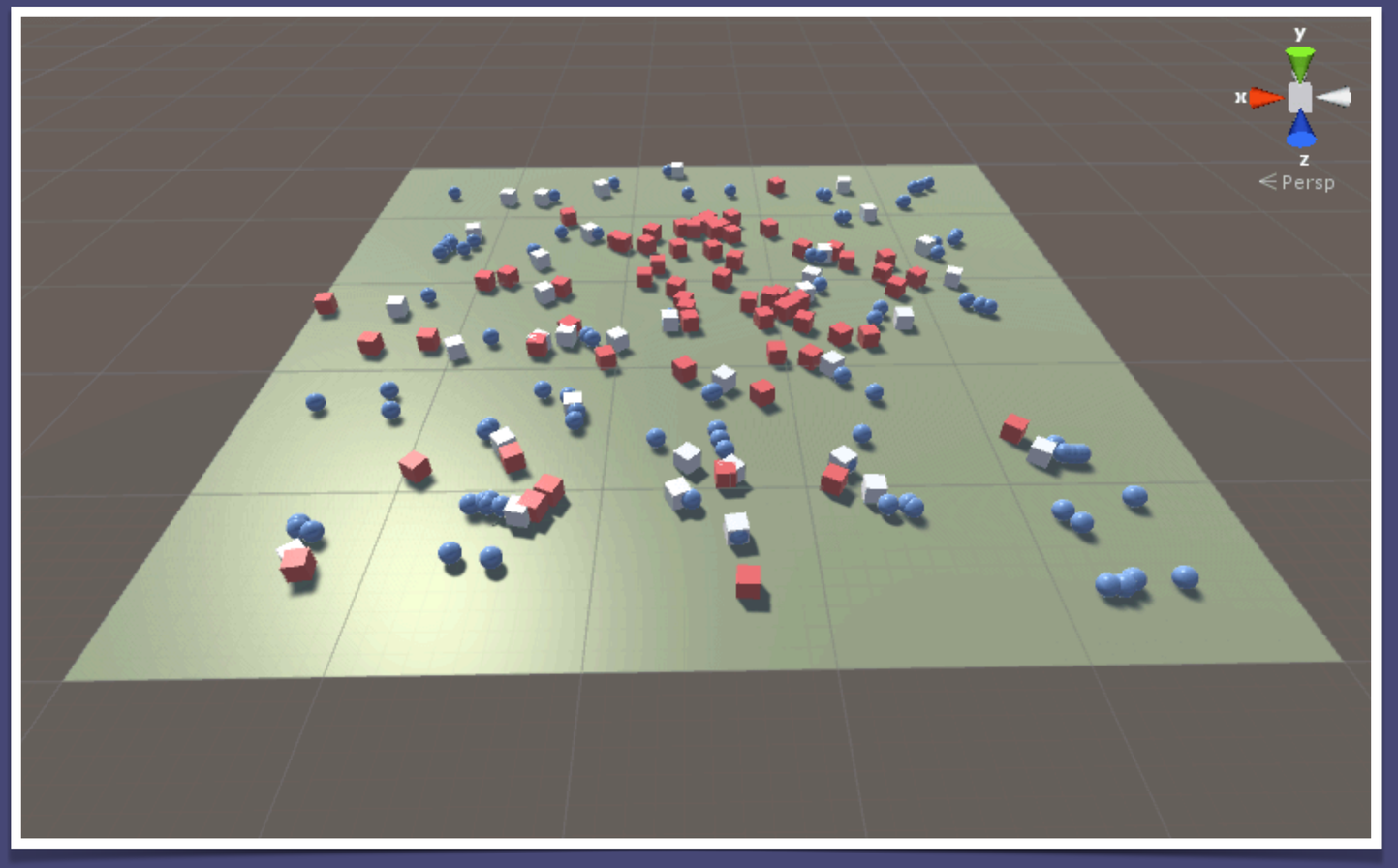
- Describe **the concept** of Design Patterns
- Understand some of the classic **'Gang of Four' Design Patterns**
- Implement some of the most **common design** patterns

Design patterns are more about how to organise code rather than about writing the code itself

Game developer - Robert Nystrom

Example scenario - Spatial Pattern

When making a game, you will often have a need to find **enemies** that are **close** to the **player**.



Role of Design Patterns

Object orientated systems tend to exhibit recurring structures that promote:

- **Abstraction**
- **Flexibility**
- **Modularity**
- **Elegance**

Formalising relationships between **objects**

CLIENT

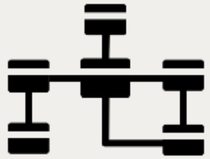
SUBJECT
creating, producing, changing,
observing or providing access

OBJECT

The Design Pattern - Categories

CREATIONAL
concerned with the process of managing object creation.

STRUCTURAL
dealing with the class and composition of objects.



BEHAVIOURAL
the different means by which objects can interact with each other.

The Design Pattern - Types

Creational	Structural	Behavioural
Abstract Factory	Adapter	Chain of Responsibility
Builder	Bridge	**Command**
Factory	Composite	Interpreter
Object	**Decorator**	Iterator
Pool	Facade	Mediator
Prototype	Flyweight	Memento
Singleton	Proxy	**Observer**
		State
		Strategy
		Template
		Visitor

Additional Game Design Pattern - Types

Double Buffer.

Game Loop.

Update Method.

Bytecode.

Subclass Sandbox

Type Object

Component.

Event Queue.

Service Locator.

Data Locality

Object Pool

Dirty Flag

Spatial Partition

http://habrador.com

http://gameprogrammingpatterns.com/

Exercise 1 - **Research a design pattern**

Define its key **properties**

- Design a poster in **MURAL** that explains the patterns key features:
  - Define the **purpose** and **use case** of the pattern
  - Explain its **principle classes, interfaces** and **implementations**
  - Use **code examples**

Exercise 2 - **Implement a design pattern**

- Using the design pattern you researched, implement an example that uses **Unity** and changes or alters game objects in some way.
- Demonstrate your approach with **well structured** and **commented code**.
- Be prepared to explain how this solves a **key problem**.
- You can apply it to your own **individual game project** if this gives you a problem that the pattern may solve.
- You can make use of **code samples** but please demonstrate how you customised the example to meet your own needs.