Unity Testing

* Challenges
  + External dependencies are not set up (some objects require complex initialization)
  + Test verifies a specific execution path that requires behavior from other classes
  + Calling methods of external might lead to irreversible changes in environment ( deleting record from database)
* Unit testing is about testing functionality in an isolated environment
  + All dependencies are mocked up
  + Only one execution path is possible
* Five test double patterns: Dummy Object, Test stub, Test spy, Mock, and Fake
* Dummy Object
  + A nominal object whose methods and properties are not intended to be used in this execution path
  + Merely a required parameter
  + Create manually
    - One dummy object per real object/interface

var functionalWeaponStub = Substitute.For<IWeapon>();

* Test Stub
  + If spaceship’s method should return some value and this value is taken from the ‘dummy object’, then the object is a **Test Stub**

// when Shoot() is called on weapon dummy object, new Shot(0,0,0) will be returned

var functionalWeaponStub = Substitute.For<IWeapon>();

functionalWeaponStub.Shoot().Returns(new[]{ new Shot(0, 0, 0) });

// returns 0, then 2, then 8 each time randomNumberService() is called

var randomNumberService = Substitue.For<IRandomNumberService>();

randomNumberService.Range(0,0).ReturnsForAnyArgs(0, 2, 8);

* + Can emulate a required sequence of random number events
* Test Spy
  + If test double has logging functionality, it is a **Test Spy**
  + Stub + logging functionality

int hitCount = 0;

var opponent = Substitute.For<ISpaceShip>();

opponent.AcceptIncomingShots( Arg.Do<IEnumerable<Shot>>(x => hitCount += x.Count());

* + NSubstitute uses special argument matcher Arg.Do to execute a delegate when argument to it passes to the substitute which means hitCount += x.Count() would be executed on each method call
* Mock Object
  + Test Spy with verification capability is called **Mock Object**
  + The test does not care about returning values of test doubles
  + It makes sure that certain methods with certain parameters were called
* Fake object
  + When a test double needs some logic
  + It is the only test double that contains logic and emulates a real system component
  + Ex: substitution of a real database with an in-memory database or using fake web service instead of a real one