

# WORKSHEET D: GRAPHICS & VFX

Version 1.0  
Computing  
COMP270

Brian McDonald, Kate Bergel

## Introduction

VFX in video games is a combination of art and programming practice, for which it is useful to have an understanding of the graphics pipeline and how to exploit its features to achieve interesting visual effects in an efficient way. In this worksheet, you will implement a VFX technique which will demonstrate your understanding of this process along with the calculations required to generate/manipulate 3D geometry and implement a shader.

Begin by **forking** the following git repository:

<https://github.com/Falmouth-Games-Academy/comp270-worksheet-D>

---

*"Today, with  
computer-generated visual  
effects, everything is possible.  
So we've seen everything. If it  
can be imagined, it can be  
put on screen."*

— Gabriel Campisi

---

**Complete** the task described below, remembering to **commit** your work regularly. To submit your work, open a **pull request** from your forked repository to the original repository.

This worksheet has only one task, with multiple components; you may approach the work and structure your code in whichever way you think is most appropriate.

## Task

For this worksheet, you should **implement** a basic VFX technique, which **must include** a procedural mesh element. You are advised to start by implementing a basic operation and iterating on it to increase the sophistication. You are free to choose any kind of effect, as long as it demonstrates your ability to manipulate data as it passes through the graphics pipeline in order to alter the appearance of an object, and you may implement your solution in either Unity or Unreal.

The following are some ideas for VFX techniques you could implement:

- Fireball
- Deflector Shield
- Liquid
- Flames
- Basic Animation (waving flags)



Image by [skeeze](#) from [Pixabay](#)

# Marking Rubric

All submissions and assessment criteria for this assignment are individual.

Criterion	Weight	Near Pass	Adequate	Competent	Very Good	Excellent	Outstanding
Basic competency threshold	30%	A reasonable attempt at the worksheet was not submitted by the formative deadline.	A reasonable attempt at the worksheet was submitted by the formative deadline. There is no evidence of academic misconduct.				
VFX Technique	40%	There is no VFX Technique implemented.	The VFX technique is very basic (e.g. some basic animation) and the procedural mesh is a basic primitive.	The VFX technique shows efforts towards sophistication, but the procedural mesh is a basic primitive.	The VFX technique is moderately sophisticated, with some effort to link it into a game context (e.g. Fireball is cast by a player character).	The VFX technique is sophisticated, well thought out, and is linked to a game action.	The VFX technique is extremely sophisticated, well thought out, and would fit well into a game.
Sophistication and Maintainability	30%	The code is only sporadically commented, if at all, or comments are unclear. Few identifier names are clear or inappropriate. Code formatting hinders readability.	The code is well commented. Some identifier names are descriptive and appropriate. An attempt has been made to adhere to a consistent formatting style. There is little obvious duplication of code or of literal values.	The code is reasonably well commented. Most identifier names are descriptive and appropriate. Most code adheres to a sensible formatting style. There is almost no obvious duplication of code or of literal values.	The code is reasonably well commented, with appropriate high-level documentation. Almost all identifier names are descriptive and appropriate. Almost all code adheres to a sensible formatting style. There is no obvious duplication of code or of literal values. Some literal values can be easily “tinkered”.	The code is very well commented, with comprehensive appropriate high-level documentation. All identifier names are descriptive and appropriate. All code adheres to a sensible formatting style. There is no obvious duplication of code or of literal values. Most literal values are, where appropriate, easily “tinkered”.	The code is commented extremely well, with comprehensive appropriate high-level documentation. All identifier names are descriptive and appropriate. All code adheres to a sensible formatting style. There is no duplication of code or of literal values. Nearly all literal values are, where appropriate, easily “tinkered”.