TDD Template

(You are not required to use all the fields; however, it is advised that you at least consider each option presented. The elements in red are the conventions used by myself—you are free to use mine, or use your own. The main goal of this document is to ensure that standards are consistent across all aspects of the project.)

Contents

[Variables: 2](#_Toc65922907)

[Public: 2](#_Toc65922908)

[Private: 2](#_Toc65922909)

[Methods: 2](#_Toc65922912)

[Public: 2](#_Toc65922913)

[Private: 2](#_Toc65922914)

[File Names: 3](#_Toc65922920)

[Scripts: 3](#_Toc65922921)

[Textures: 3](#_Toc65922923)

[Materials: 4](#_Toc65922924)

[Models: 4](#_Toc65922925)

[Animations: 4](#_Toc65922926)

[Prefabs: 4](#_Toc65922928)

[Flow Graphs 5](#_Toc65922930)

# Variables:

## Public:

Public variables will use rotating camel case where the first letter of the first word will have a lower case. All words following that word will include a capital letter to start the new word.

public var publicExample;

## Private:

Private variables will use the same system as publics. However, an underscore will be added to the front if they reference an Object or component.

private GameObject \_privateExample;

# Methods:

All methods will follow the naming convention that the first letter of the first word is capitalized, the first letter of each word following will also be capitalized. Methods do not need special names with the exception to those that return a bool, these should be formed in a question such as IsVisible or CanSee.

## Public:

Same as above, fill if needed.

## Private:

Same as above, fill if needed.

# File Names:

## Duplicate objects:

PineTree

PineTree (1)

PineTree (2)

## Scripts:

The first letter of each word will be capitalized. A base class will include the term Base in its name. For example:

ItemBase

EquipItem

WeaponEquip

MeleeWeapon

Classes that do not have children that inherit from them do no require the Base term.

Any scripts that do not fall into these categories are free to have any name as long as it makes sense and is easy to understand.

## Textures:

We have one texture for the Skybox

Skybox

## Materials:

The given name will be based on a fruit closest to the color of the Material and will be capitalized. Otherwise it will describe it’s property.

Lemon

## Models:

We imported one Model set:

Sci-Fi Warehouse Kit

## Animations:

The given name should reflect what this is an animation of, such as idle, run, etc. All words will be capitalized. For example:

RunForward

## Prefabs:

The given name should reflect what this is a prefab of, such as enemy, house, etc. It will specify if it is a pickup item. All words will be capitalized. For example:

BasicHealthPickup

# Flow Graphs:

Graphs will be laid out so that it is clear where one class starts, what inherits from it, and what variables and methods it will require. Graphs are not written in stone and will rarely include everything required by the end of a project. They act as a starting point to ensure baseline functionality.

Class Names: Blue

Public Elements: Green

Private Elements: Red

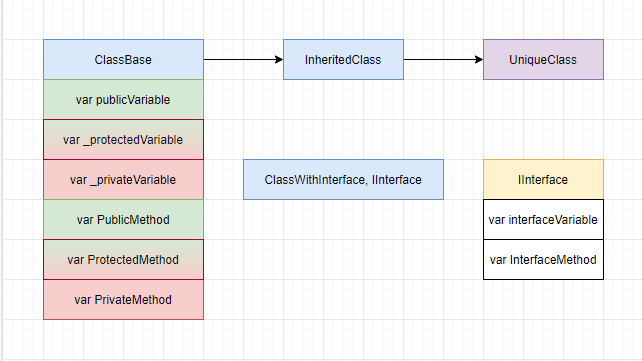
Protected Elements: Gradient of Green to Red

Unique Classes: Purple

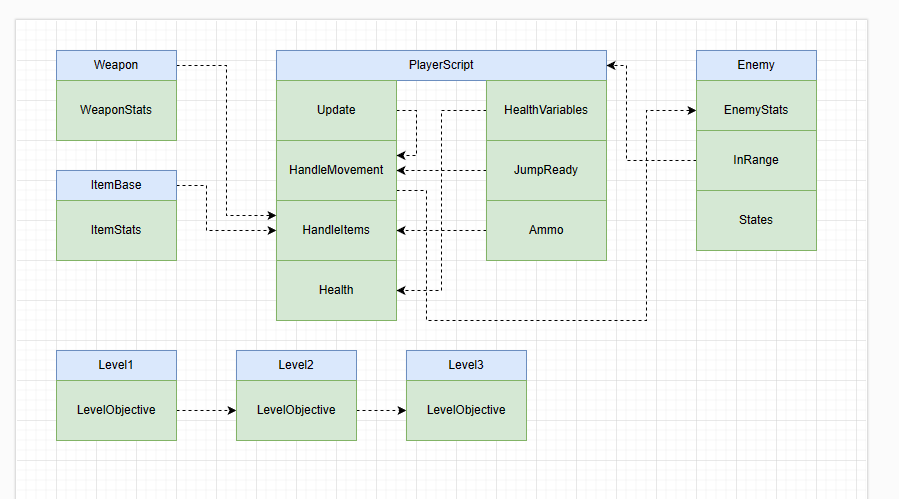
Interface Names: Yellow

Interface Variables: White

All previously defined naming conventions will be applied to the table, for example:



Lines will be used to help define the relationship between classes and methods. A solid line that goes from one class to another will indicate a parent:child relationship. A dotted line from one method to another will indicate the methods communicating to each other. These dotted lines should be read from top to bottom to understand the general layout of the code.



Reminder that the graphs are used to help plan code, but changes can still occur. The main purpose of flow graphs are so two members of a team can work on separate scripts, but ensure the scripts will work together in the end. The success of this will be determined from correct naming and protection levels, each of which is shown within the graphs.