***Team 2020 Coding Standards***

1. **Indentation**

Allman styling should be used. With indents being four (4) spaces for each level of scope. Every new curly bracket “{“ should be on a new line. Every method should have 1-2 lines of whitespace before/after it.

* ex:
  + void exampleMethod()
  + {
  + …
  + }
  + int exmapleToo()
  + {
  + …
  + }

1. **Naming**

**Variables**

* camelCase styling should be used. The variable names should start with a lowercase and every new word begins with an uppercase. Single worded variable names shall not be capitalized.
  + ex:
    - int speed;
    - string nameOfPerson;
* Avoid using single lettered variables unless it’s being used for iteration.

**Constants**

* Constants should be all capitalized and spaced with an underscore.
  + ex:
    - const int NUM\_OF\_FISH = 383;
    - const string LEVEL\_NAME = “Level 2”;

**Classes/Types**

* UpperCamelCase styling should be used. The class names should start with an uppercase and every new word begins with an uppercase as well.
  + ex:
    - Child\* child1;
    - Dog\* pug;
    - SceneManager\* sceneManager;

**Functions**

* camelCase styling should be used. The function names should start with a lowercase and every new word begins with an uppercase.
* Functions do actions, therefore, a verb should be included
* Parameters: include a space after a comma
  + ex:
    - void killTheEnemy();
    - int calculateHealth(GameObject Player, int damage);

1. **Commenting**

In general, include an inline (single line) comment on before most of the functions giving a brief description on the function’s action.

* ex:
  + // Makes the player jump when the spacebar is presses
  + void makePlayerJump()
  + {
  + …
  + }
* For bigger explanations use a multi-line comment:
  + ex:
    - /\*
    - \* This file is going to be in-charge of
    - \* moving the player and giving the
    - \* player the ability to pick up items.
    - \*/

1. **Error Handling**

When handling with errors use the assert() function

* Here is an example on how to use the assert() function:
  + void assert(expression);