BEST PRACTICE AND TIPS
THAT YOU CAN PRINT AND KEEP CLOSE BY ☺

A FEW THINGS TO REMEMBER WHEN YOU CREATE A SCRIPT (CHECKLIST)

As you create your first scripts in the next chapter, there will be, without a doubt, errors and possibly hair pulling :-). You see, when you start coding, you will, as for any new activity, make small mistakes, learn what they are, improve your coding, and ultimately get better at writing your scripts. As I have seen students learning scripting, there are some common errors that are usually made; these don't make you a bad programmer; on the contrary, it is part of the learning process.

> We all learn by trial and error, and making mistakes is part of the learning process.

So, as you create your first script, set any fear aside, try to experiment, be curious, and get to learn the language. It is like learning a new foreign language: when someone from a foreign country understands your first sentences, you feel so empowered! So, it will be the same with C#, and to ease the learning process, I have included a few tips and things to keep in mind when writing your scripts, so that you progress even faster. You don't need to know all of these by now (I will refer to these later on, in the next chapter), but just be aware of it and also use this list if any error occurs (this list is also available as a pdf file in the resource pack, so that you can print it and keep it close by). So, watch out for these :-).

- Each opening bracket has a corresponding closing bracket.
- All variables are written consistently (e.g., spelling and case). The name of each variable is case-sensitive; this means that if you declare a variable **myvariable** but then refer to it as **myVariable** later on in the code, this may trigger an error, as the variable **myVariable** and **myvariable**, because they have a different case (upper- or lower-case **V**), are seen as two different variables.
- All variables are declared (type and name) prior to being used (e.g., **int**).
- The type of the argument passed to a method is the type that is required by this method.
- The type of the argument returned by a method is the type that is required to be returned by this method.
- Built-in functions are spelt with the proper case (e.g., upper-case **U** for **Update**).
- Use **camel casing** (i.e., capitalize the first character of each word except for the first word) or **Pascal casing** (i.e., capitalize the first character of each word) consistently.

- All statements are ended with a semi-colon.
- For **if** statements the condition is within round brackets.
- For **if** statements the condition uses the syntax **"**==" rather than "=".
- When calling a method, the exact name of this method (i.e., case-sensitive) is used.
- When referring to a variable, it is done with regards to the access type of the variable (e.g., public or private).
- Local variables are declared and can be used within the same method.
- Global variables are declared outside methods and can be used anywhere within the class.

BEST PRACTICES

To ensure that your code is easy to understand and that it does not generate countless headaches when trying to modify it, there are a few good practices that you can start applying as your begin with coding; these should save you some time along the line.

Variable naming

- Use meaningful names that you can understand, especially after leaving your code for two weeks.

```
string myName = "Patrick";//GOOD
string b = "Patrick";//NOT SO GOOD
```

- Capitalize words within a name consistently (e.g., camel or Pascal casing).

```
bool testIfTheNameIsCorrect;// GOOD
bool testifthenameiscorrect; // NOT SO GOOD
```

Methods

- Check that all opening brackets have a corresponding closing bracket.
- Indent your code.
- Comment your code as much as possible to explain how it works.
- Use the **Start** method if something just needs to be done once at the start of the game.
- If something needs to be done repeatedly, then the method **Update** might be a better option.