This framework is built on the basis of game played on [www.choiceofgames.com/dragon](http://www.choiceofgames.com/dragon) . There are multiples scenarios and each scenario has multiple options which further open more scenarios. So I have created a class which starts the game on user’s command:

    class **Program**

    {

        static void **Main**(string[] args)

        {

            Console.**WriteLine**("do you want to start the game? (Y/N) :");

            char opinion = char.**Parse**(Console.**ReadLine**());

            if (opinion == keys.Y)

            {

                scenerio1 first = new scenerio1();

                first.**scenerio**();

            }

        }

    }

Then the controls is transferred to class scenerio1. Scenerio1 is extending a parent abstract class Scenerios:

    abstract class **scenerios**

    {

        public abstract void **scenerio**();

    }

Now we can create multiple classes, each for its own question and all of them will be required to include a function Scenerio().

The class scenerio1, extending class Scenerios, has a override function Scenerio(). This function includes the first scenario:

class **scenerio1** : scenerios

    {

        public override void **scenerio**()

        {

            Console.**Clear**();

            Console.**WriteLine**(" A knight charges up the slope at you. His horse pounds at the ground, carrying the heavily armored warrior as if he were a child's doll.\n The knight sets his lance to attack you. How do you defend yourself, O mighty dragon ? ");

            Console.**WriteLine**("\n A: I take to the air with a quick beat of my wings.");

            Console.**WriteLine**("\n B: I knock the knight from his horse with a slap of my tail.");

            Console.**WriteLine**("\n C: I rush into his charge and tear him to pieces with my claws.");

            Console.**WriteLine**("");

            Console.**WriteLine**("Enter your option: ");

            char option = char.**Parse**(Console.**ReadLine**());

            scenerio\_decider.**first**(option);

        }

    }

The user will be asked for his option and his option will be transferred to another class Scenerio\_decider.

    class **scenerio\_decider**

    {

        public static void **first**(int option)

        {

            if (option == keys.A)

            {

                scenerio1A second = new scenerio1A();

                second.**scenerio**();

            }

            if (option == keys.B)

            {

                scenerio1B second = new scenerio1B();

                second.**scenerio**();

            }

            if (option == keys.C)

            {

                scenerio1C second = new scenerio1C();

                second.**scenerio**();

            }

        }

        public static void **second**(int option)

        {

        }

    }

This class has a function **first**, which will decide the next scenario according to option received from scenario 1. Similarly, there is a function **second**, which will decide the next scenario according to option received from second scenario. Th function has named for ease of reader so that he can understand which function decides which scenario.

We also have a class keys. This class contains all static keys with their ASCII values. So when developer wants to compare the input character(option) with available characters, he just has to write **keys.(x)** , where x can be any alphabet.

The developer will also benefit from potential naming convention. For example, the scenerio1 has three options. Each option opens different scenario. So if option A was selected, the scenario\_decider will transfer the control to function scenerio1A, which indicates that this scenario is created due to A option of scenerio1. Likewise, function scenerio1B will be outcome of option B of scenerio1.

Now if developer wants to add new scenario, all he has to do is to create a new class, lets say, scenerio1AA. This will mean this new scenario will be result of option A of scenerio1A. the developer will inherit this class from parent abstract class scenerio. here he will write a new text and new options and then send the option to scenario\_decider. The last work will have to do is to tell function of senerio\_decider, what will be the next scenario according to received option.