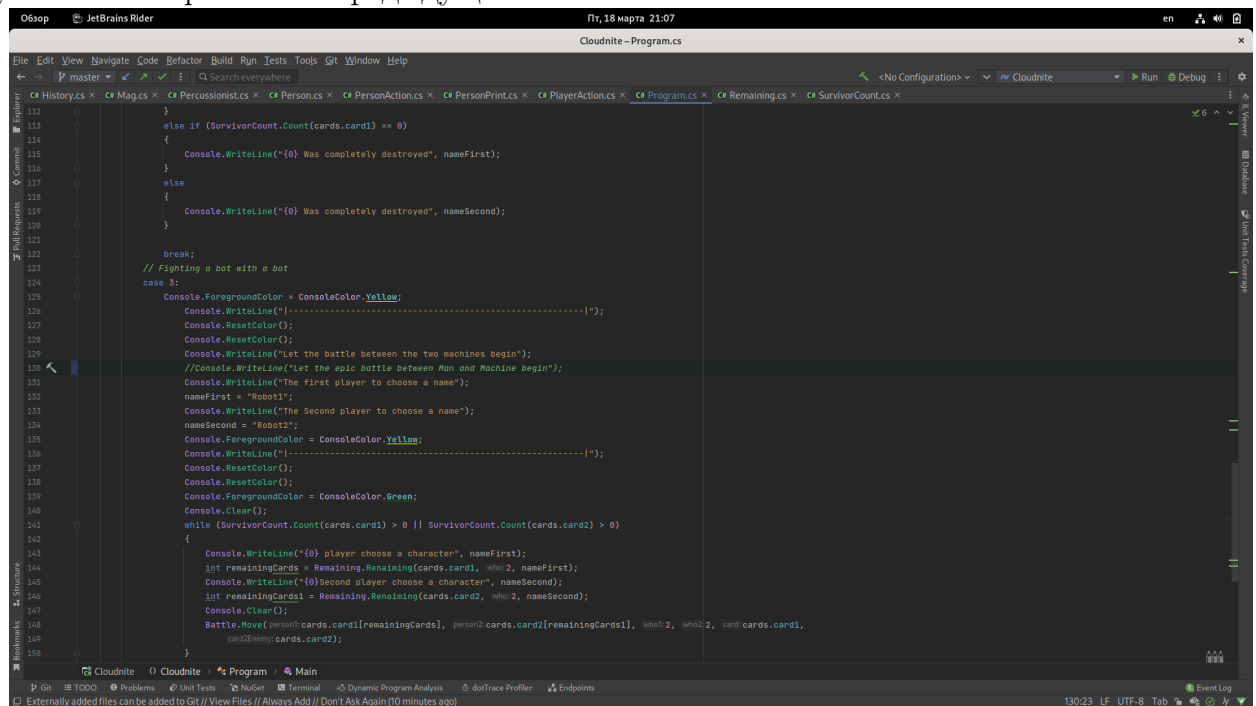


Review of Sergey Golovin by Dmitry Pilyuk

March 2022

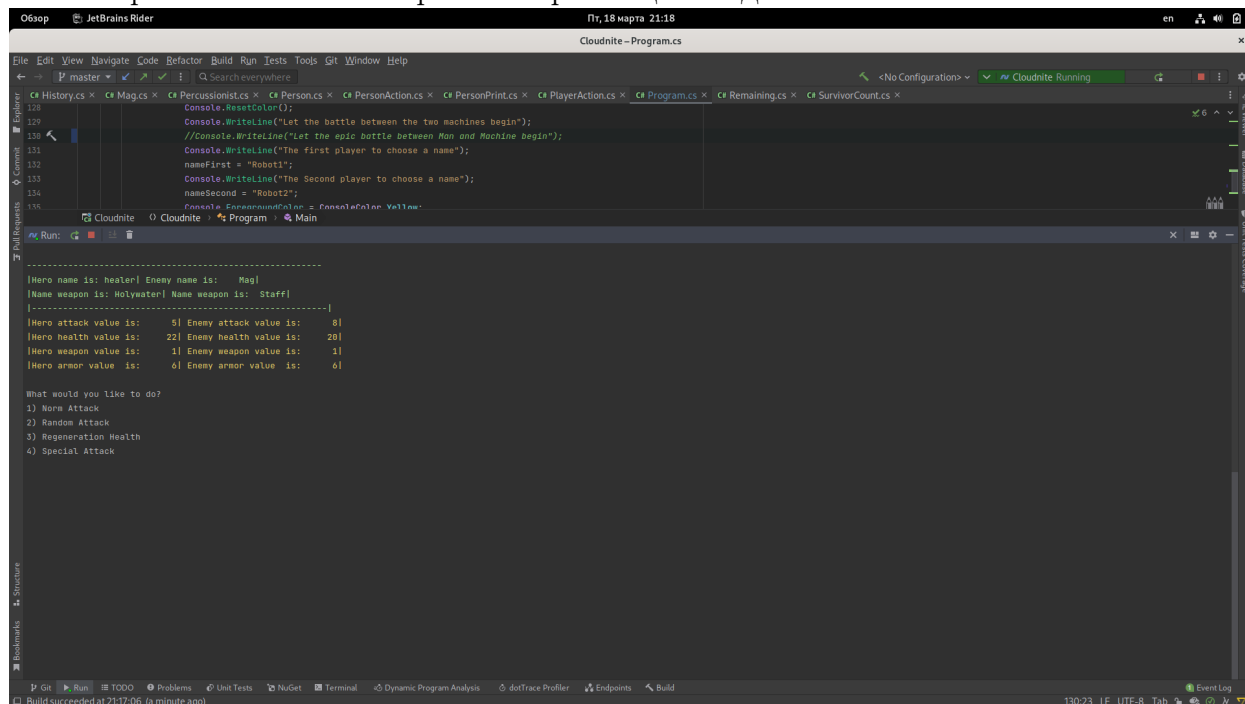
1 Недостатки

1) В классе Program метод Main строка 130 скорее всего лишняя и была случайно скопирована из предыдущих ветвлений switch.



```
112     }
113     else if (SurvivorCount.Count(cards.card1) == 0)
114     {
115         Console.WriteLine("{0} Was completely destroyed", nameFirst);
116     }
117     else
118     {
119         Console.WriteLine("{0} Was completely destroyed", nameSecond);
120     }
121
122     break;
123     // Fighting a bot with a bot
124     case 3:
125         Console.ForegroundColor = ConsoleColor.Yellow;
126         Console.WriteLine("-----");
127         Console.ResetColor();
128         Console.ResetColor();
129         Console.WriteLine("Let the battle between the two machines begin");
130         //Console.WriteLine("Let the epic battle between Man and Machine begin");
131         Console.WriteLine("The first player to choose a name");
132         nameFirst = "Robot1";
133         Console.WriteLine("The second player to choose a name");
134         nameSecond = "Robot2";
135         Console.ForegroundColor = ConsoleColor.Yellow;
136         Console.WriteLine("-----");
137         Console.ResetColor();
138         Console.ResetColor();
139         Console.ForegroundColor = ConsoleColor.Green;
140         Console.Clear();
141         while (SurvivorCount.Count(cards.card1) > 0 || SurvivorCount.Count(cards.card2) > 0)
142         {
143             Console.WriteLine("{0} player choose a character", nameFirst);
144             int remainingCards = Remaining.Remaining(cards.card1, who: 2, nameFirst);
145             Console.WriteLine("{0} second player choose a character", nameSecond);
146             int remainingCards1 = Remaining.Remaining(cards.card2, who: 2, nameSecond);
147             Console.Clear();
148             Battle.Move(person1: cards.card1[remainingCards], person2: cards.card2[remainingCards1], who1: 2, who2: 2, card: cards.card1,
149                 cardEnemy: cards.card2);
150         }
```

2)Порядок хода в классе Battle, метод Move определяется случайно, но в интерфейсе игры не пишет какому игроку предлагается сделать ход. Было бы хорошо писать имя игрока совершающего ход.

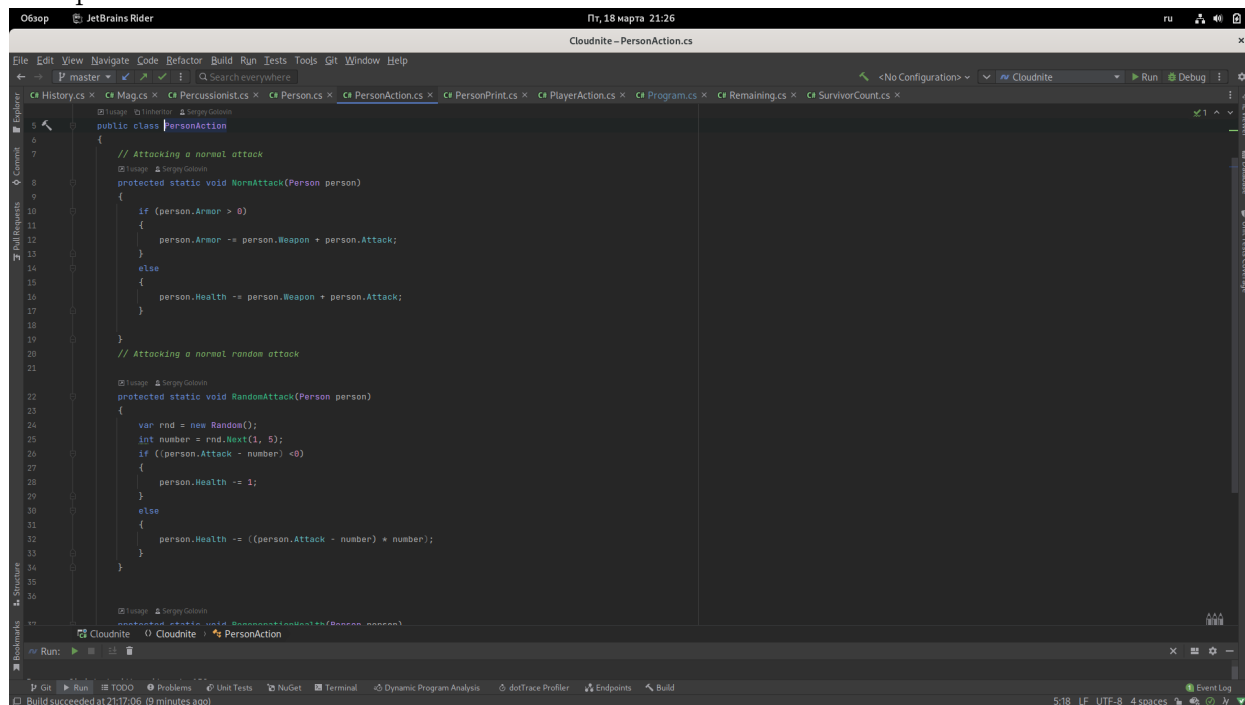


```
Console.ResetColor();
Console.WriteLine("Let the battle between the two machines begin");
//Console.WriteLine("Let the epic battle between Man and Machine begin");
Console.WriteLine("The first player to choose a name");
nameFirst = "Robot1";
Console.WriteLine("The second player to choose a name");
nameSecond = "Robot2";
Console.ForegroundColor = ConsoleColor.Yellow;
```

```
-----
Hero name is: healer| Enemy name is:  Mag|
Name weapon is: Holywater| Name weapon is: Staff|
-----
Hero attack value is:  9| Enemy attack value is:  8|
Hero health value is: 22| Enemy health value is: 20|
Hero weapon value is:  1| Enemy weapon value is:  1|
Hero armor value is:  6| Enemy armor value is:  6|

What would you like to do?
1) Norm Attack
2) Random Attack
3) Regeneration Health
4) Special Attack
```

3) В классе PersonActions атака и случайная атака наносит урон самому себе, т.к. оперируем с объектом класса person и имеется строка `person.Health -= person.Weapon + person.Attack;`, т.е. из здоровья person вычитаем урон person. Методы RegenerationHealth и Special Attack работают нормально.



```
public class PersonAction
{
    // Attacking a normal attack
    protected static void NormAttack(Person person)
    {
        if (person.Armor > 0)
        {
            person.Armor -= person.Weapon + person.Attack;
        }
        else
        {
            person.Health -= person.Weapon + person.Attack;
        }
    }

    // Attacking a normal random attack
    protected static void RandomAttack(Person person)
    {
        var rnd = new Random();
        int number = rnd.Next(1, 5);
        if ((person.Attack - number) < 0)
        {
            person.Health -= 1;
        }
        else
        {
            person.Health -= ((person.Attack - number) * number);
        }
    }
}
```

2 Предложения

1) В классе Remaining метод Remaining можно не проверять случай когда $flag = 0$ (срока 29), т.к. наличие карт в руке проверяется при каждом вхождении в циклы, в методе Main.

2) Добавить контроль вводных данных, для переменных которые используются в ветвлениях switch или if.

3) В методе Main в строках 45-70, 94-120 и 141-164, описан процесс боя, можно вынести это в отдельный метод, что бы не повторять 3 раза.

4) Возможно имеет смысл немного визуализировать бой компьютера с компьютером, и расставить паузы после ходов, с ожиданием действия от наблюдателя, например при помощи Console.ReadKey().

3 Достоинства

В программе соблюдается кодстайл, приятное оформление меню игры в консоли, с использованием разных цветов текста и условной псевдографики с помощью различных символов. Интересная идея игрового процесса.