Program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace BackendTest
{
    enum choice
    {
        rock = 1, paper, scissos, potion
    }
     class Program
    {
        static void Main(string[] args)
        {
            bool valid;
            string name;

            //remove the temp player for program simplicity lang po

            Console.WriteLine("Welcome to Game");
            do
            {
                Console.Write("Enter your Name: ");
                name = Console.ReadLine();
                valid = name.All(char.IsLetter);

                if (!valid)
                {
                    Console.WriteLine("Please enter alphabetical letters only.");
                }

            } while (!valid);

            Game game = new Game(name);
            game.Play();
        }
    }
}
```

Player.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace BackendTest
{
     class Player
    {
        public string Name { get; set; }
        public int Health { get; private set; }
        public int Potion { get; private set; } = 1;

        //removed/commented this part because I will be using the other method so I
could use the playername.
        /*public Player()
        {

            Health = 6;
        }*/

        //adjusted the health since up to 3 lang daw
        public Player(string name)
        {
            Name = name;
            Health = 3;
        }

        public void TakeDamage()
        {
            Health -= 1;
            if (Health < 0) Health = 0;
        }

        public void PotionHeal()
        {
            if(Health > 0 && Potion == 1)
            {
                Console.WriteLine($"\n{Name} used a potion and healed 1 health
point!");
                Health += 1;
                Potion--;
            }
            else if (Health <= 0 && Potion ==1) //if player is dead and has a potion
left
            {
                Random random = new Random();
                int heal = random.Next(1,3);//random generated
                if(heal == 1)
                {
                    Health = 1; //revive player with 1 health point
                    Potion--;
                    Console.WriteLine($"\n{Name} has been revived with 1 health
point!");
                }
```

```
                else
                {
                    Console.WriteLine($"\n{Name} failed to revive.");
                }
            }
            else
            {
                Console.WriteLine($"\n{Name} has no potions left to use.");
            }
        }
    }
}
```

Game.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace BackendTest
{
    class Game
    {
        private Player player;
        private Player computer;
        private Random  random;

        /* Deleted/commented this part because I will be using the other method so I
could use the playername.
        public Game()
        {
            player = new Player();
            computer = new Player("Computer");
            random = new Random();
        }*/
        public Game(string playerName)
        {
            player = new Player(playerName);
            computer = new Player("Computer");
            random = new Random();
        }
        public void Play()
        {
            //added variables for input validation
            bool valid;
            int value;

            while (player.Health>0 && computer.Health > 0)
            {
                Console.WriteLine("\nChoose your move:");
                Console.WriteLine("1. Rock (Press '1')");
                Console.WriteLine("2. Paper (Press '2')");
                Console.WriteLine("3. Scissors (Press '3')");
                Console.WriteLine("4. Use Potion (Press '4')");
```

```
                //fixed the datatypes & added input validation
                do
                {
                    ConsoleKeyInfo playerinput = Console.ReadKey();
                    valid = (int.TryParse(playerinput.KeyChar.ToString(), out value)
&& value > 0 && value < 5);

                    if (!valid)
                    {
                        Console.WriteLine("\nInvalid choice. Please press 1, 2, 3,
or 4.");
                    }

                } while (!valid);


                int computerinput = random.Next(1,4);//random generated


                int result = checkwinner(value, computerinput);

                //revive logic
                if (player.Health <= 0 && player.Potion == 1)
                {
                    player.PotionHeal();
                    if (player.Health > 0)
                    {
                        Console.WriteLine($"{player.Name} was revived! The game
continues...");
                        continue; //tuloy pag nabuhay ang player
                    }
                }

                if (result == 1)
                {

                    Console.WriteLine($"\n{player.Name} won this turn!");

                }
                else if (result == -1)
                {

                    Console.WriteLine($"\ncomputer won this turn!");

                }else if (result == 0)
                {
                    Console.WriteLine("\nIt's a tie! No damage taken.");
                }
                else if (result == 4)
                {

                    Console.WriteLine($"\n");
                }

                Console.WriteLine($"Remaining Health - {player.Name}:
{player.Health}, Computer: {computer.Health}");
                Console.WriteLine($"Remaining Potions - {player.Name}:
{player.Potion}");
```

```
        }

        if (player.Health <= 0)
        {
            Console.WriteLine("\nComputer Wins! Game over");
        }
        else
        {
            Console.WriteLine($"\n{player.Name} Wins! Congratulations!");
        }

    }
    private int checkwinner(int playerChoice, int cominput)
    {
        //1 rock
        //2 paper
        //3 scissors
        int p = playerChoice;
        int c = cominput;

        //added a condition for potion
        if (p == 4)
        {
            player.PotionHeal();
            return 4; //return 4 if player used potion
        }
        else
        {
            ShowPicked(p, player.Name);
            ShowPicked(c, computer.Name);
        }

        //player wins
        if ((p == 1 && c == 3) || (p == 2 && c == 1) || (p == 3 && c == 2))
        {
            computer.TakeDamage();
            return 1;
        }
        // computer wins
        else if ((c == 1 && p == 3) || (c == 2 && p == 1) || (c == 3 && p == 2))
        {
            player.TakeDamage();
            return -1;
        }
        // tie
        else if (p == c)
        {
            return 0;
        }
        return -1;
    }

    //added a method to show the chocie
    public static void ShowPicked(int choice, string name)
    {
        if(choice == 1)
        {
```

```
                Console.WriteLine($"\n{name} picked ROCK!");
            } else if(choice == 2)
            {
                Console.WriteLine($"\n{name} picked PAPER!");
            }else if(choice == 3)
            {
                Console.WriteLine($"\n{name} picked SCISSORS!");
            }else if(choice == 4){
                Console.WriteLine($"\n{name} picked POTION!");
            }
        }
    }
}
```

OUTPUTS

Microsoft Visual Studio Debug Console

```
Welcome to Game
Enter your Name: rich

Choose your move:
1. Rock (Press '1')
2. Paper (Press '2')
3. Scissors (Press '3')
4. Use Potion (Press '4')
3
rich picked SCISSORS!

Computer picked PAPER!

rich won this turn!
Remaining Health - rich: 3, Computer: 2
Remaining Potions - rich: 1

Choose your move:
1. Rock (Press '1')
2. Paper (Press '2')
3. Scissors (Press '3')
4. Use Potion (Press '4')
1
rich picked ROCK!

Computer picked SCISSORS!

rich won this turn!
Remaining Health - rich: 3, Computer: 1
Remaining Potions - rich: 1

Choose your move:
1. Rock (Press '1')
2. Paper (Press '2')
3. Scissors (Press '3')
4. Use Potion (Press '4')
1
rich picked ROCK!

Computer picked SCISSORS!

rich won this turn!
Remaining Health - rich: 3, Computer: 0
Remaining Potions - rich: 1

rich Wins! Congratulations!

D:\NICO\WPH\Assignments\BackendTest\BackendTest\bin\Debug\BackendTest.exe (process 2080) exited with code 0 (0x0).
Press any key to close this window . . .
```

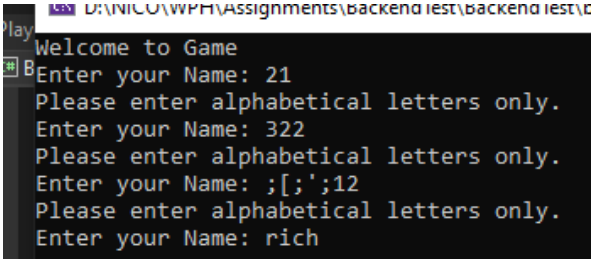
```
Welcome to Game
Enter your Name: RICH

Choose your move:
1. Rock (Press '1')
2. Paper (Press '2')
3. Scissors (Press '3')
4. Use Potion (Press '4')
3
RICH picked SCISSORS!

Computer picked SCISSORS!

It's a tie! No damage taken.
Remaining Health - RICH: 3, Computer: 3
Remaining Potions - RICH: 1

Choose your move:
1. Rock (Press '1')
2. Paper (Press '2')
3. Scissors (Press '3')
4. Use Potion (Press '4')
3
RICH picked SCISSORS!

Computer picked ROCK!

computer won this turn!
Remaining Health - RICH: 2, Computer: 3
Remaining Potions - RICH: 1

Choose your move:
1. Rock (Press '1')
2. Paper (Press '2')
3. Scissors (Press '3')
4. Use Potion (Press '4')
3
RICH picked SCISSORS!

Computer picked ROCK!

computer won this turn!
Remaining Health - RICH: 1, Computer: 3
Remaining Potions - RICH: 1

Choose your move:
1. Rock (Press '1')
2. Paper (Press '2')
3. Scissors (Press '3')
4. Use Potion (Press '4')
3
RICH picked SCISSORS!

Computer picked ROCK!

computer won this turn!
Remaining Health - RICH: 0, Computer: 3
Remaining Potions - RICH: 1
RICH failed to revive.

D:\NICO\WPH\Assignments\BackendTest\BackendTe
Press any key to close this window . . .
```

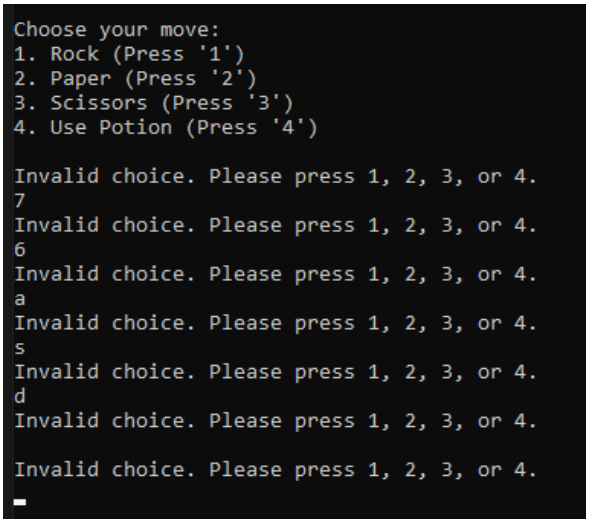
```
Remaining Health - Rich: 1, Computer: 3
Remaining Potions - Rich: 1

Choose your move:
1. Rock (Press '1')
2. Paper (Press '2')
3. Scissors (Press '3')
4. Use Potion (Press '4')
4
Rich used a potion and healed 1 health point!


Remaining Health - Rich: 2, Computer: 3
Remaining Potions - Rich: 0

Choose your move:
1. Rock (Press '1')
2. Paper (Press '2')
3. Scissors (Press '3')
4. Use Potion (Press '4')
1
Rich picked ROCK!

Computer picked PAPER!

computer won this turn!
Remaining Health - Rich: 1, Computer: 3
Remaining Potions - Rich: 0

Choose your move:
1. Rock (Press '1')
2. Paper (Press '2')
3. Scissors (Press '3')
4. Use Potion (Press '4')
2
Rich picked PAPER!

Computer picked SCISSORS!

computer won this turn!
Remaining Health - Rich: 0, Computer: 3
Remaining Potions - Rich: 0

Computer Wins! Game over

D:\NICO\WPH\Assignments\BackendTest\BackendTest
Press any key to close this window . . .
```

REVIVE LOGIC

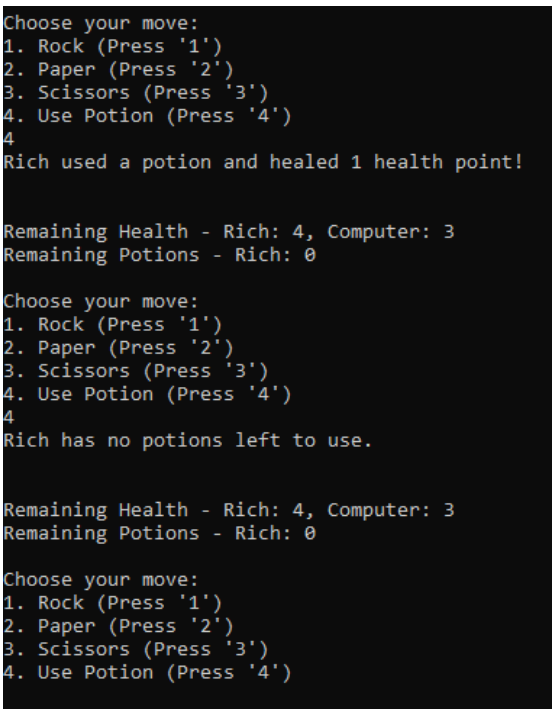
```
Choose your move:
1. Rock (Press '1')
2. Paper (Press '2')
3. Scissors (Press '3')
4. Use Potion (Press '4')
1
Rich picked ROCK!

Computer picked PAPER!

computer won this turn!
Remaining Health - Rich: 1, Computer: 1
Remaining Potions - Rich: 1

Choose your move:
1. Rock (Press '1')
2. Paper (Press '2')
3. Scissors (Press '3')
4. Use Potion (Press '4')
1
Rich picked ROCK!

Computer picked PAPER!

Rich has been revived with 1 health point!
Rich was revived! The game continues...

Choose your move:
1. Rock (Press '1')
2. Paper (Press '2')
3. Scissors (Press '3')
4. Use Potion (Press '4')

Invalid choice. Please press 1, 2, 3, or 4.
4
Rich has no potions left to use.


Remaining Health - Rich: 1, Computer: 1
Remaining Potions - Rich: 0

Choose your move:
1. Rock (Press '1')
2. Paper (Press '2')
3. Scissors (Press '3')
4. Use Potion (Press '4')
1
Rich picked ROCK!

Computer picked SCISSORS!

Rich won this turn!
Remaining Health - Rich: 1, Computer: 0
Remaining Potions - Rich: 0

Rich Wins! Congratulations!
```

VALIDATIONS





POTION USE



REFLECTIONS & EXPLANATIONS

For this project, we were asked to debug a Rock-Paper-Scissors game using C#. I made several changes to the code to make sure it followed the given requirements and worked as expected.

1. Changed How the Player Is Created
- At first, the Player class had a constructor with no arguments, but I wasn't really using it in the program. So I removed it and just used the constructor that takes a name. This helped clean up the code and made things simpler in Program.cs.
2. Fixed the Health Initialization
- The original code gave the player 6 health points, but the instructions said it should only be 3. I updated the Player class so that both the player and computer start with 3 health points instead.
3. Added Input Validation
- I added checks to make sure the player can only enter valid choices (1 to 4). If they enter something else, the program tells them it's invalid and asks again. This helps prevent errors during the game. Also added a feature that only accepts valid names, no characters or number.
4. Implemented the Potion Feature
- I added the ability for the player to use a potion by pressing 4. It heals 1 health point and only works once per game. When the potion is used, the computer skips its move for that turn.
5. Added Auto-Revive Mechanic
- If the player's health drops to 0 and they still have a potion left, there's a 50% chance they'll automatically revive with 1 health point. I made sure this happens inside the game loop so the game keeps going if the player revives.
6. Fixed Some Logic in the Game
- I corrected some of the conditions in the checkwinner() method to make sure the game properly decides who wins, loses, or ties based on the choices.
7. Cleaned Up the Output
- I improved the way the game shows messages after each round, including what choices were made, how much health each player has, and how many potions are left.

Overall, I focused on cleaning up the code, making sure it works based on the requirements, and adding the bonus potion features. The game now runs properly and is easier to understand and play.