

```
//resource from https://gist.github.com/edokeh/7580064
```

[illegible]

```
using System;
using System.IOEnumeration;
using System.Runtime.InteropServices;
```

```
namespace APCSP
{
```

```

internal class Program
{
    struct Player
    {
        public string looksLike;
        public string name;
        public int atk;
        public int atkedHp;
        public int hp;
        public int recentHp;
        public int criticalRate;
        public int xPosition;
        public int yPosition;
        public string weapon = "Basketball";

        public Player(string looksLike, string name, int atk, int hp, int criticalRate, int
xPosition, int yPosition)
        {
            this.looksLike = looksLike;
            this.name = name;
            this.atk = atk;
            this.hp = hp;
            this.recentHp = hp;
            this.criticalRate = criticalRate;
            this.xPosition = xPosition;
            this.yPosition = yPosition;
        }

        public Player(string looksLike, string name, int atk, int hp, int criticalRate, int
xPosition)
        {

```

```

        this.looksLike = looksLike;
        this.name = name;
        this.atk = atk;
        this.hp = hp;
        this.recentHp = hp;
        this.criticalRate = criticalRate;
        this.xPosition = xPosition;
        this.yPosition = 0;
    }

    public int RandomGenerateXPosition()
    {
        Random r = new Random();
        int xPosition = r.Next(2, 149);
        while (xPosition % 2 != 0)
        {
            xPosition = r.Next(2, 149);
        }
        return xPosition;
    }

    public int RandomGenerateYPosition(int xPosition, int[] xPositions, int[] yPositions,
    Player kunKun, Player monster)
    {
        Random r = new Random();
        int yPosition = r.Next(1, 34);
        bool continueLoop = true;

        while (continueLoop)
        {
            if (xPosition == kunKun.xPosition && yPosition == kunKun.yPosition)

```

```

        {
            yPosition = r.Next(1, 34);
            continue;
        }

        for (int i = 0; i < xPositions.Length; i++)
        {
            if (xPosition == xPositions[i] && yPosition == yPositions[i])
            {
                yPosition = r.Next(1, 34);
                break;
            }
            continueLoop = false;
        }
    }
    return yPosition;
}

public void Show() //print looksLike
{
    Console.SetCursorPosition(xPosition, yPosition);
    Console.Write(looksLike);
}

}

static bool Move(int[] importantXPositions, int[] importantYPositions, ref Player kunKun, ref
Player monster, ref Player boss, System.Media.SoundPlayer sound)
{
    Random r = new Random();
    while (true)
    {

```

```

    kunKun.recentHp = kunKun.hp;
    ResetProfile(ref kunKun);
    ClearConsole();
    Console.SetCursorPosition(kunKun.xPosition, kunKun.yPosition);
    switch (Console.ReadKey(true).KeyChar)
    {
        case 'w':
            if (CheckRepeatPosition(kunKun.xPosition, kunKun.yPosition - 1,
importantXPositions, importantYPositions))
            {
                CoverLastPosition(kunKun.xPosition, kunKun.yPosition);
                kunKun.yPosition -= 1;
                kunKun.Show();
            }
            break;
        case 's':
            if (CheckRepeatPosition(kunKun.xPosition, kunKun.yPosition + 1,
importantXPositions, importantYPositions))
            {
                CoverLastPosition(kunKun.xPosition, kunKun.yPosition);
                kunKun.yPosition += 1;
                kunKun.Show();
            }
            break;
        case 'a':
            if (CheckRepeatPosition(kunKun.xPosition - 2, kunKun.yPosition,
importantXPositions, importantYPositions))
            {
                CoverLastPosition(kunKun.xPosition, kunKun.yPosition);
                kunKun.xPosition -= 2;
                kunKun.Show();
            }

```

```

        }
        break;
    case 'd':
        if (CheckRepeatPosition(kunKun.xPosition + 2, kunKun.yPosition,
importantXPositions, importantYPositions))
        {
            CoverLastPosition(kunKun.xPosition, kunKun.yPosition);
            kunKun.xPosition += 2;
            kunKun.Show();
        }
        break;
    case 'j': //attack
        switch (IsValidAttack(kunKun.xPosition, kunKun.yPosition, importantXPositions,
importantYPositions))
        {
            case 0: //question
                if (Question(ref kunKun))
                {
                    kunKun.weapon = "Physics Excalibur";
                    kunKun.atk += 999;
                    kunKun.hp += 9999;
                    ResetProfile(ref kunKun);
                    ClearConsole();
                    Console.WriteLine("Congradulations! You got Weapon: Physics
Excalibur");

                    Console.SetCursorPosition(2, 37);
                    Console.WriteLine("HP + 9999, atk + 999");
                    Console.SetCursorPosition(2, 38);
                    Console.WriteLine("Press any key to continue.");
                    Console.ReadKey(true);
                }

```

```

        break;
    case 1: //boss
        if (Fight(ref kunKun, ref boss)) //win
        {
            sound = new System.Media.SoundPlayer(Properties.Resources.Win);
            sound.Play();
            return true;
        }
        else //lose
        {
            sound = new
System.Media.SoundPlayer(Properties.Resources.WhyYouDoThat);
            sound.Play();
            return false;
        }
    case 2: //monster
        if (Fight(ref kunKun, ref monster)) //win, regenerate monster
        {
            System.Media.SoundPlayer player = new
System.Media.SoundPlayer(Properties.Resources.Win);
            player.Play();
            kunKun.hp += 100;
            kunKun.atk += 10;
            ResetProfile(ref kunKun);
            CoverLastPosition(monster.xPosition, monster.yPosition);
            monster = new Player("▲", "monster", r.Next(kunKun.atk -
(int)(kunKun.atk * 0.1), kunKun.atk + (int)(kunKun.atk * 0.1)), r.Next(kunKun.hp - (int)(kunKun.hp *
0.1), kunKun.hp + (int)(kunKun.hp * 0.1)), 10, monster.RandomGenerateXPosition());
            monster.yPosition =
monster.RandomGenerateYPosition(monster.xPosition, importantXPositions, importantYPositions, kunKun,
monster);

```

```

        importantXPositions[2] = monster.xPosition;
        importantYPositions[2] = monster.yPosition;
        monster.Show();
    }
    else
    {
        sound = new
System.Media.SoundPlayer(Properties.Resources.WhyYouDoThat);
        sound.Play();
        return false;
    }
    break;
}
break;
case '1': //special gift
    if (Console.ReadKey(true).KeyChar == '1' ? Console.ReadKey(true).KeyChar == '4'
? Console.ReadKey(true).KeyChar == '5' ? Console.ReadKey(true).KeyChar == '1' ?
Console.ReadKey(true).KeyChar == '4' ? true : false : false : false : false : false)
    {
        kunKun.weapon = "Lawyer's Letter ";
        kunKun.atk += 250;
        kunKun.hp += 1919;
        ResetProfile(ref kunKun);
        ClearConsole();
        Console.Write("Congradulations! You got Weapon: Lawyer's letter");
        Console.SetCursorPosition(2, 37);
        Console.Write("HP + 1919, atk + 250");
        Console.SetCursorPosition(2, 38);
        Console.Write("Press any key to continue.");
        Console.ReadKey(true);
    }
}

```



```

        break;
    }
}

static void ResetHpProfile(Player kunKun)
{
    Console.ForegroundColor = ConsoleColor.Yellow;
    Console.SetCursorPosition(152, 3);
    Console.Write("                ");
    Console.SetCursorPosition(152, 3);
    if (kunKun.recentHp < 0)
    {
        kunKun.recentHp = 0;
    }
    Console.Write("HP: {0}/{1}", kunKun.recentHp, kunKun.hp);
}

static void ResetProfile(ref Player kunKun)
{
    kunKun.recentHp = kunKun.hp;
    Console.ForegroundColor = ConsoleColor.Yellow;
    Console.SetCursorPosition(152, 3);
    Console.Write("HP: {0}/{1}", kunKun.recentHp, kunKun.hp);
    Console.SetCursorPosition(152, 4);
    Console.Write("ATK: {0}", kunKun.atk);
    Console.SetCursorPosition(152, 5);
    Console.Write("Weapon: {0}", kunKun.weapon);
}

static bool Question(ref Player kunKun) //3 questions

```

```
{
    ClearConsole();
    Console.Write("How many laws do Newton's Laws have?");
    if (Console.ReadKey(true).KeyChar == '3')
    {
        ClearConsole();
        Console.Write("You are right! HP + 100, atk + 10");
        Console.SetCursorPosition(2, 37);
        Console.Write("Press any key to continue.");
        Console.ReadKey(true);
        kunKun.hp += 100;
        kunKun.atk += 10;
        ResetProfile(ref kunKun);
        ClearConsole();
        Console.Write("Is 0.9 repeat equals 1? Type 1 for yes, 2 for no");

        if (Console.ReadKey(true).KeyChar == '1')
        {
            ClearConsole();
            Console.Write("You are right! HP + 200, atk + 20");
            Console.SetCursorPosition(2, 37);
            Console.Write("Press any key to continue.");
            Console.ReadKey(true);
            kunKun.hp += 200;
            kunKun.atk += 20;
            ResetProfile(ref kunKun);
            ClearConsole();
            Console.Write("What is Love?");
            Console.CursorVisible = true;
            Console.SetCursorPosition(2, 37);
        }
    }
}
```

```

        if (Console.ReadLine().ToLower().Trim() == "baby don't hurt me")
        {
            Console.CursorVisible = false;
            ClearConsole();
            Console.Write("You are right! HP + 300, atk + 30");
            Console.SetCursorPosition(2, 37);
            Console.Write("Press any key to continue.");
            Console.ReadKey(true);
            kunKun.hp += 300;
            kunKun.atk += 30;
            ResetProfile(ref kunKun);
            return true;
        }
    }

    Console.CursorVisible = false;
    ClearConsole();
    Console.Write("That's incorrect");
    Console.SetCursorPosition(2, 37);
    Console.Write("Press any key to continue.");
    Console.ReadKey(true);
    ClearConsole();
    return false;
}

static bool Fight(ref Player kunKun, ref Player nonKunKun)
{
    ClearConsole();
    Console.Write("Start fight with {0}!", nonKunKun.name);
    Console.SetCursorPosition(2, 37);
    Console.Write("Press any key to continue.");

```

```

while (kunKun.recentHp >= 0)
{
    Console.ReadKey(true);
    if (JAttack(ref kunKun, ref nonKunKun)) //if one of their's hp <= 0
    {
        ClearConsole();
        if (nonKunKun.recentHp <= 0)
        {
            Console.WriteLine("You hit {0} {1} hp, {0} still have {2} hp", nonKunKun.name,
kunKun.atkedHp, nonKunKun.recentHp);
            Console.SetCursorPosition(2, 38);
            Console.WriteLine("Congradulations! You beat the {0}!", nonKunKun.name);
            Console.SetCursorPosition(2, 39);
            Console.WriteLine("Press any key to continue.");
            ResetHpProfile(kunKun);
            Console.ReadKey(true);
            return true;
        }
        else
        {
            Console.WriteLine("You hit {0} {1} hp, {0} still have {2} hp", nonKunKun.name,
kunKun.atkedHp, nonKunKun.recentHp);
            Console.SetCursorPosition(2, 37);
            Console.WriteLine("{0} hit you {1} hp, you still have {2} hp", nonKunKun.name,
nonKunKun.atkedHp, kunKun.recentHp);
            Console.SetCursorPosition(2, 39);
            Console.WriteLine("You lost");
            Console.SetCursorPosition(2, 40);
            Console.WriteLine("Press any key to continue.");
            ResetHpProfile(kunKun);
            Console.ReadKey(true);
        }
    }
}

```

```

        return false;
    }
}
ClearConsole();
Console.WriteLine("You hit {0} {1} hp, {0} still have {2} hp", nonKunKun.name,
kunKun.atkedHp, nonKunKun.recentHp);
Console.SetCursorPosition(2, 37);
Console.WriteLine("{0} hit you {1} hp, you still have {2} hp", nonKunKun.name,
nonKunKun.atkedHp, kunKun.recentHp);
Console.SetCursorPosition(2, 38);
Console.WriteLine("Press any key to continue.");
ResetHpProfile(kunKun);
}
return true;
}

static bool JAttack(ref Player kunKun, ref Player nonKunKun)
{
    Random r = new Random();
    kunKun.atkedHp = kunKun.atk + (r.Next(0, 101) <= kunKun.criticalRate ? kunKun.atk : 0);
    nonKunKun.atkedHp = nonKunKun.atk + (r.Next(0, 101) <= nonKunKun.criticalRate ?
nonKunKun.atk : 0);
    if (nonKunKun.recentHp - kunKun.atkedHp <= 0)
    {
        nonKunKun.recentHp = 0;
        return true;
    }
    else if (kunKun.recentHp - nonKunKun.atkedHp <= 0)
    {
        nonKunKun.recentHp -= kunKun.atkedHp;
        kunKun.recentHp = 0;
    }
}

```

```

        return true;
    }
    nonKunKun.recentHp -= kunKun.atkedHp;
    kunKun.recentHp -= nonKunKun.atkedHp;
    return false;
}

static void CoverLastPosition(int xPosition, int yPosition)
{
    Console.SetCursorPosition(xPosition, yPosition);
    Console.Write(" ");
}

static void ClearConsole()
{
    for (int i = 0; i < 10; i++)
    {
        Console.SetCursorPosition(2, 36 + i);
        Console.Write("
");
    }
    Console.SetCursorPosition(2, 36);
}

static int IsValidAttack(int xPosition, int yPosition, int[] importantXPositions, int[]
importantYPositions)
{
    for (int i = 0; i < importantXPositions.Length; i++)
    {
        if (importantXPositions[i]==xPosition+2 || importantXPositions[i]==xPosition-2)
        {

```

```

        if (importantYPositions[i] == yPosition)
        {
            return i;
        }
    }
    else if (importantYPositions[i] == yPosition + 1 || importantYPositions[i] == yPosition
- 1)
    {
        if (importantXPositions[i] == xPosition)
        {
            return i;
        }
    }
}
return 114514;
}

```

```

static bool CheckRepeatPosition(int xPosition, int yPosition, int[] importantXPositions, int[]
importantYPositions) //for kunKun
{
    if (xPosition > 148 || xPosition < 2 || yPosition > 34 || yPosition < 1) //don't cross red
blocks
    {
        return false;
    }

    for (int i = 0; i < importantXPositions.Length; i++)
    {
        if (xPosition == importantXPositions[i] && yPosition == importantYPositions[i])
        {
            return false;
        }
    }
}

```

```

        }
    }

    return true;
}

static void CreateObjects(int[] importantXPositions, int[] importantYPositions, ref Player
kunKun, ref Player monster, ref Player boss)
{
    Random r = new Random();
    kunKun = new Player("答", "KunKun", 10, 100, 10, 74, 18);
    monster = new Player("▲", "monster", r.Next(kunKun.atk - (int)(kunKun.atk * 0.1),
kunKun.atk + (int)(kunKun.atk * 0.1)), r.Next(kunKun.hp - (int)(kunKun.hp * 0.1), kunKun.hp +
(int)(kunKun.hp * 0.1)), 10, monster.RandomGenerateXPosition());
    monster.yPosition = monster.RandomGenerateYPosition(monster.xPosition, importantXPositions,
importantYPositions, kunKun, monster);
    boss = new Player("首", "boss", 114, 5141, 91, 98, 10); //hen, hen, hen, aaaaaaaaaaaaaaaaaa

    importantXPositions[2] = monster.xPosition; //1st question, 2nd BOSS, 3rd monster
    importantYPositions[2] = monster.yPosition;

    Console.SetCursorPosition(148, 34);
    Console.Write(" ? ");
}

static void StartGamePage()
{
    RedBlocks();

    Console.ForegroundColor = ConsoleColor.White;
    Console.SetCursorPosition(71, 5);
}

```



```

    Console.WriteLine("KunKun Fight");

    Console.ForegroundColor = ConsoleColor.Red;
    Console.SetCursorPosition(71, 14);
    Console.WriteLine("Start Game");

    Console.ForegroundColor = ConsoleColor.White;
    Console.SetCursorPosition(72, 18);
    Console.WriteLine("Quit Game");

    Console.ForegroundColor = ConsoleColor.Yellow;
    Console.SetCursorPosition(157, 37);
    Console.WriteLine("Press W & S to");
    Console.SetCursorPosition(156, 38);
    Console.WriteLine("change selection");
    Console.SetCursorPosition(159, 41);
    Console.WriteLine("Press J to");
    Console.SetCursorPosition(156, 42);
    Console.WriteLine("choose selection");
}

static void EndGamePage(bool winOrLose)
{
    string title = winOrLose ? "You Won!" : "You Lost";
    Console.Clear();
    RedBlocks();

    Console.ForegroundColor = ConsoleColor.White;
    Console.SetCursorPosition(73, 5);
    Console.WriteLine(title);
}

```

```
    Console.ForegroundColor = ConsoleColor.Red;
    Console.SetCursorPosition(71, 14);
    Console.Write("Replay Game");

    Console.ForegroundColor = ConsoleColor.White;
    Console.SetCursorPosition(72, 18);
    Console.Write("Quit Game");

    Console.ForegroundColor = ConsoleColor.Yellow;
    Console.SetCursorPosition(157, 37);
    Console.Write("Press W & S to");
    Console.SetCursorPosition(156, 38);
    Console.Write("change selection");
    Console.SetCursorPosition(159, 41);
    Console.Write("Press J to");
    Console.SetCursorPosition(156, 42);
    Console.Write("choose selection");
}

static void InGameGuide()
{
    Console.ForegroundColor = ConsoleColor.Yellow;
    Console.SetCursorPosition(158, 31);
    Console.Write(" ? = question");
    Console.SetCursorPosition(158, 32);
    Console.Write("答 = yourself");
    Console.SetCursorPosition(159, 33);
    Console.Write("▲ = monster");
    Console.SetCursorPosition(157, 34);
    Console.Write("首 = final boss");
}
```

```
        Console.SetCursorPosition(156, 38);
        Console.Write("Press WASD to move");
        Console.SetCursorPosition(159, 41);
        Console.Write("Press J to");
        Console.SetCursorPosition(157, 42);
        Console.Write("interact others");
    }

    static void RedBlocks()
    {
        Console.ForegroundColor = ConsoleColor.Red;
        Console.SetCursorPosition(0, 0);
        for (int i = 0; i < 90; i++)
        {
            Console.Write("■" + " ");
        }

        Console.SetCursorPosition(0, 35);
        for (int i = 0; i < 90; i++)
        {
            Console.Write("■" + " ");
        }

        Console.SetCursorPosition(0, 46);
        for (int i = 0; i < 90; i++)
        {
            Console.Write("■" + " ");
        }

        Console.SetCursorPosition(0, 0);
        for (int i = 0; i < 46; i++)
```

```

    {
        Console.Write("■");
        Console.SetCursorPosition(0, Console.CursorTop + 1);
    }

    Console.SetCursorPosition(150, 0);
    for (int i = 0; i < 46; i++)
    {
        Console.Write("■");
        Console.SetCursorPosition(150, Console.CursorTop + 1);
    }

    Console.SetCursorPosition(178, 0);
    for (int i = 0; i < 46; i++)
    {
        Console.Write("■");
        Console.SetCursorPosition(178, Console.CursorTop + 1);
    }
}

static bool StartGameOrNot(bool startOrEnd)
{
    string title = startOrEnd ? "Start  Game" : "Replay Game";
    bool start = true;
    bool endLoop = false;
    while (!endLoop)
    {
        switch (Console.ReadKey(true).KeyChar)
        {
            case 'w':
                Console.ForegroundColor = ConsoleColor.White;

```

```

        Console.SetCursorPosition(72, 18);
        Console.Write("Quit Game");
        Console.ForegroundColor = ConsoleColor.Red;
        Console.SetCursorPosition(71, 14);
        Console.Write(title);
        start = true;
        break;

    case 's':
        Console.ForegroundColor = ConsoleColor.White;
        Console.SetCursorPosition(71, 14);
        Console.Write(title);
        Console.ForegroundColor = ConsoleColor.Red;
        Console.SetCursorPosition(72, 18);
        Console.Write("Quit Game");
        start = false;
        break;
    case 'j':
        endLoop = true;
        return start;
    }
}
return start;
}

static void Main(string[] args)
{
    //1st question, 2nd BOSS, 3rd monster
    int[] importantXPositions = { 148, 98, 0 }; //arrays without monster's position
    int[] importantYPositions = { 34, 10, 0 };
    Console.SetWindowSize(200, 100);

```

```
    Console.SetBufferSize(210, 110);
    int recentStage = 0;
    Random r = new Random();
    Console.CursorVisible = false;

    Console.WriteLine("Please use Windows && screen size >= 15.6 inches to run this
program:)");
    Console.Write("Full screen, then press any key to start.");
    Console.ReadKey(true);
    Console.Clear();

    StartGamePage();
    while (true)
    {
        switch (recentStage)
        {
            case 0: //start page
                recentStage = StartGameOrNot(true) ? 1 : 4;
                break;
            case 1: //in game
                Console.Clear();
                RedBlocks();
                InGameGuide();

                Player kunKun = new Player();
                Player monster = new Player();
                Player boss = new Player();
                CreateObjects(importantXPositions, importantYPositions, ref kunKun, ref
monster, ref boss);

                kunKun.Show();
```

```

        monster.Show();
        Console.ForegroundColor = ConsoleColor.Red;
        boss.Show();
        ResetProfile(ref kunKun);

        //source from online
        System.Media.SoundPlayer sound = new
System.Media.SoundPlayer(Properties.Resources.JNTM);
        sound.Play();

        recentStage = Move(importantXPositions, importantYPositions, ref kunKun, ref
monster, ref boss, sound) ? 2 : 3;

        break;
    case 2: //win game page
        EndGamePage(true);
        recentStage = StartGameOrNot(false) ? 1 : 4;
        break;
    case 3: //lose game page
        EndGamePage(false);
        recentStage = StartGameOrNot(false) ? 1 : 4;
        break;
    case 4: //end game
        Console.Clear();
        Console.ForegroundColor = ConsoleColor.White;
        Environment.Exit(0);
        break;
    }
}
}

```

