

Matheus Maracajá

Turn based rpg Analyses

First of all, I analyzed the Program script directly and looked for some errors.

The code was broken when I was running. On grid script newBox was used before it was created. So I just organized it.

First problem as running: There's no indicator of position

There was some errors in allocation of players, this function that should do a random for player and enemy position was not being used.

```
int GetRandomInt(int min, int max)
{
    var rand = new Random();
    int index = rand.Next(min, max);
    return index;
}
```

An empty grid was being created and used as parameter, so I just changed for a useful one

```
GridBox currentgrid = new GridBox();
if (currentgrid.ocupied)
{
    //if()
    Console.Write("[X]\t");
}
else
{
    Console.Write($"[ ]\t");
}
```

I did a foreach to look at the grid list and found the ones that was occupied.

```
foreach (GridBox current_grid in grids)
{
    if(current_grid.xIndex == i && current_grid.yIndex == j)
    {
        if (current_grid.ocupied)
        {
            Console.Write("[X]\t");
        }

        else if (!current_grid.ocupied)
        {
            Console.Write($"[ ]\t");
        }
    }
}
```

Second problem as running: Movement and near check

Then, the move function was working kind of strange, so I checked it and made de if for each check. The directions was wrong too so I made they look right.

```
if ((battlefield.grids.Exists(x => x.Index == currentBox.Index - 1)))
{
    currentBox.ocupied = false;
    battlefield.grids[currentBox.Index] = currentBox;
    currentBox = (battlefield.grids.Find(x => x.Index == currentBox.Index - 1));
    currentBox.ocupied = true;
    battlefield.grids[currentBox.Index] = currentBox;
    Console.WriteLine($"Player {PlayerIndex} walked up\n");
    battlefield.drawBattlefield(5, 5);

    return;
}
```

Some variables of EnemyCharacter creation was not being set, just player stats was being modified

```

void CreateEnemyCharacter()
{
    //randomly choose the enemy class and set up vital variables
    var rand = new Random();
    int randomInteger = rand.Next(1, 4);
    CharacterClass enemyClass = (CharacterClass)randomInteger;
    Console.WriteLine($"Enemy Class Choice: {enemyClass}");
    EnemyCharacter = new Character(enemyClass);
    EnemyCharacter.Health = 100;
    PlayerCharacter.BaseDamage = 20;
    PlayerCharacter.PlayerIndex = 1;
    StartGame();
}

```

On turn manager, there some inaccessible codes, like this

```

} else if(currentBox.xIndex < Target.currentBox.xIndex)
{
    currentBox.ocupied = false;
    battlefield.grids[currentBox.Index] = currentBox;
    currentBox = (battlefield.grids.Find(x => x.Index == currentBox.Index + 1));
    currentBox.ocupied = true;
    return;
    battlefield.grids[currentBox.Index] = currentBox;
    Console.WriteLine($"Player {PlayerIndex} walked right\n");
    battlefield.drawBattlefield(5, 5, currentBox, Target.currentBox);
}

```

And some codes in wrong order, like the drawBattleField function should run by last.

```

if (this.currentBox.yIndex > Target.currentBox.yIndex)
{
    battlefield.drawBattlefield(5, 5, currentBox);
    this.currentBox.ocupied = false;
    battlefield.grids[currentBox.Index] = currentBox;
    this.currentBox = (battlefield.grids.Find(x => x.Index == currentBox.Index - battlefield.xLenght));
    this.currentBox.ocupied = true;
    battlefield.grids[currentBox.Index] = currentBox;
    Console.WriteLine($"Player {PlayerIndex} walked up\n");
    return;
}

```

Here was || and not &

```

if (left || right || up || down)
{
    return true;
}

```

Third problem as running: Combat not working

So right now, everything was working fine for movement and start the fight. But the combat was wrong too

1 - The damage was wrong. It was using BaseDamage and not the damage parameter. I changed BaseDamage for amount

```
1 referência
public bool TakeDamage(float amount)
{
    if((Health -= BaseDamage) <= 0)
    {
        Die();
        return true;
    }
    return false;
}
```

2 - The enemy damage was not set. So i change from player to enemy

```
void CreateEnemyCharacter()
{
    //randomly choose the enemy class and set up vital variables
    var rand = new Random();
    int randomInteger = rand.Next(1, 4);
    CharacterClass enemyClass = (CharacterClass)randomInteger;
    Console.WriteLine($"Enemy Class Choice: {enemyClass}");
    EnemyCharacter = new Character(enemyClass);
    EnemyCharacter.Health = 100;
    PlayerCharacter.BaseDamage = 20;
    PlayerCharacter.PlayerIndex = 1;
    StartGame();
}
```

3 - There's no change with class and damage multiplier. So when the player chooses the class I made each one have different attributes, so I created a function to make this. Setting each class health, damage and multiplier (I used it as a crit damage).

```
void Setup_class_info(CharacterClass player_class, CharacterClass enemy_class)
{
    switch (player_class)
    {
        case CharacterClass.Paladin:
            PlayerCharacter.Health = 120;
            PlayerCharacter.BaseDamage = 10;
            PlayerCharacter.DamageMultiplier = 1.5f;
            break;
        case CharacterClass.Warrior:
            PlayerCharacter.Health = 100;
            PlayerCharacter.BaseDamage = 20;
            PlayerCharacter.DamageMultiplier = 1.6f;
            break;
        case CharacterClass.Cleric:
            PlayerCharacter.Health = 150;
            PlayerCharacter.BaseDamage = 5;
            PlayerCharacter.DamageMultiplier = 1.1f;
            break;
        case CharacterClass.Archer:
            PlayerCharacter.Health = 70;
            PlayerCharacter.BaseDamage = 20;
            PlayerCharacter.DamageMultiplier = 2;
            break;
    }
}
```

Since the enemy class had no variable created I created and setted it for him. The player class was not being used too.

```
void CreatePlayerCharacter(int classIndex)
{
    CharacterClass characterClass = (CharacterClass)classIndex;
    Console.WriteLine($"Player Class Choice: {characterClass}");
    PlayerCharacter = new Character(characterClass);
    PlayerCharacter.PlayerIndex = 0;

    CreateEnemyCharacter();
}
```

I made some visual feedback to better the match. For this i got the reference from player and enemy character

```
1 referência
public void Attack (Character target, Character player, Character enemy)
{
    var rand = new Random();
    target.TakeDamage(BaseDamage);
    Console.WriteLine($"Player {PlayerIndex} is attacking the player {Target.PlayerIndex} and did {BaseDamage} damage\n");
    Console.WriteLine("Player health " + player.Health);
    Console.WriteLine("Enemy health " + enemy.Health);
}
```

At last I did the Crit and stun case. I implemented a random rate and used it for everytime this number hit 9 or 10 it stunted the opponent for 2 turns.

```
1 referência
public void Attack (Character target, Character player, Character enemy)
{
    Random random_rate = new Random();
    int num = random_rate.Next(0, 10);

    if (num >= 9)
    {
        Console.WriteLine($"Player {PlayerIndex} stunned player {Target.PlayerIndex} and did {BaseDamage} damage\n");
        target.TakeDamage(BaseDamage * DamageMultiplier);
        target.rounds_stunned = 2;
    }
}
```