Esse documento tem a única e exclusiva finalidade de mostrar o progresso de criação e alteração de código envolvendo o projeto AUTO BATTLE. Para consulta do GDD do projeto, acesse o link abaixo:

https://docs.google.com/document/d/15OolJKZ_w-G3_eCipQ59GLVsxFXyGatnxMR1QsjsyS8/edit?usp=sharing

Versão 1.2.0 (Final)

Melhorias de Versão

- Agora personagens podem tomar Knockback de forma aleatória e mudar de posição no campo;
- Corrigido erro de personagens poderem atacar mesmo sem vida durante os turnos.

Feature - Ataques podem afastar o personagem de forma aleatória

---

Classe Character / Método *StartTurn()*

- Durante a comparação, se o personagem está perto do inimigo, é também comparado se a vida do personagem e do inimigo estão maiores que zero para que o código execute o ataque e o knockback.
- Se a chance sorteada obedecer a condição, o personagem atacante é empurrado para outra posição no campo. O código final ficou assim:

```
55          if (CheckCloseTargets(battlefield))
56          {
57              // Se o personagem tem vida e seu inimigo também, o ataque é realizado
58              if (health > 0 && target.health > 0)//alterado
59              {
60                  Attack(target);
61
62                  // Chance aleatória do personagem mudar de posição no campo após realizar ataque
63                  var rand = new Random();
64                  int chanceToKnockback = rand.Next(0, 100);
65
66                  if (chanceToKnockback > 85)
67                  {
68                      Console.WriteLine($"Player {name} tomou um empurrão do adversário!");
69                      int knockbackChar = rand.Next(0, gridBoxesTotal);
70                      currentBox.ocupied = false;
71                      battlefield.grids[currentBox.index] = currentBox;
72                      currentBox = battlefield.grids.Find(x => x.index == knockbackChar);
73                      currentBox.ocupied = true;
74                      currentBox.charType = playerType;
75                      battlefield.grids[currentBox.index] = currentBox;
76                      battlefield.DrawBattlefield();
77                  }
78              }
```

Refatoração - Correção do Script Character.cs

Classe Character / Método *EndGame()*

- Foi adicionado o comando *Environment.Exit(0)* para o programa ser totalmente parado depois de um vencedor, evitando qualquer possível execução de comandos após o jogo acabar.

```
47      public void EndGame()
48      {
49          Console.WriteLine($"Player {name} venceu o {target.name}!");
50          Environment.Exit(0);
51      }
```

Refatoração - Correção do Script Program.cs

Classe Program / Método *StartTurn()*

- Cada personagem recebe agora o número de boxes definidos inicialmente para serem escolhidas na hora de mudarem de posição aleatoriamente no método *StartTurn()*.

```
205                 foreach (Character character in AllPlayers)
206                 {
207                     character.gridBoxesTotal = grid.grids.Count;
208                     character.StartTurn(grid);
209                 }
```