**Candidate: Felipe Zarattini Miranda**

This entry is a refactor of a past attempt at the test. I analyzed what I believe I did wrong the first time and made corrections.

**Character.cs**

* Added a Title variable to differentiate between player and enemy.
* Added an event to be raised when the character dies.
* Removed Linq statements from CheckCloseTargets for performance.
* Maintained Linq statements on HandleMovement for legibility since they are called less than the ones in CheckCloseTargets.
* Changed movement to be able to walk diagonally.
* Added DisplayDirection method to show the player where the characters moved on the current round.
* Added CheckSpecialAbilityRemainingTurns for special abilities that have a set amount of turns to be active.
* Added several Console.WriteLines to show the player what is happening with the character during the rounds.
* Added TrySpecialAbility method to roll a random chance to use special abilities.
* Added ExecuteSpecialAbility method to perform the special ability when TrySpecialAbility returns true.

**Types.cs**

* Added CharacterClassSpecific struct to hold modifiers and a reference to the class special ability.
* Added Special Ability struct, which is constructed through a switch statement based on the character class passed as parameter.
* Added the variable OccupyingCharacter to the GridBox.

**Utilities.cs**

* Created this class to hold generic methods such as random generation and string validation methods.

**Grid.cs**

* Fixed the xLength and yLength relationships with lines and columns.
* Added the initial letter of the character class to the gridbox they are occupying.

**Program.cs**

* Added the initialization of the CharacterClassSpecifics, Special abilities and the classes dictionary to link the characterClass enums to their respective CharacterClassSpecifics.
* Added string validation to the grid size player choice.
* Fixed the random values for the enemy character because it was only choosing between 3 classes.
* Added a ShufflePlayerList method so the character that plays first in the game is always random.
* Changed a lot of Console.WriteLine and to make the game better to read and added a separation for rounds. Left a commentary line to clear the console as a possibility.
* Added the method HandleEndGame which hails the player as the winner or as the loser.
* Subscribed onCharacterDied event to HandleEndGame method.
* Added restart game functionality