

Design Description

Program name : Project4_Design_Description_Schaefer_Kristin.pdf
Author : Kristin Schaefer
Date : 05-26-2019

Main Function

Description: The Main function creates a Game object and calls all of the Game functions necessary to play the Fantasy Combat game. Particularly the game setup function and the play tournament function.

Process:

1. Create Game object
2. Use do-while loop to call Game functions
3. Check if user wants to continue playing with boolean value
4. Exit loop once the user selects to exit the program

Character Class

Description: The Character class is the parent class of Barbarian, Vampire, Blue Men, Medusa and Harry Potter. It is an abstract class and does not defines only basic member functions such as the getter and setter functions, as well as the recovery function. Compared to project 3, the Character class is redefined to have the Character object act as a node for the doubly-linked list objects Team and Loser.

Key data members:

1. Character *next
 - A pointer to a Character object node that is next in the doubly-linked list
 - Set to nullptr in the header file
2. 1. Character *prev
 - A pointer to a Character object node that is previos in the doubly-linked list
 - Set to nullptr in the header file
3. string playerName
 - Holds the user-specified player name

Key functions:

1. recovery()
 - Uses the player's dice roll to determine if the damage inflicted is reduced to 75%, 50% or 25%.
 - returns an int value
2. getPrevPtr() / getNextPtr()
 - Returns a pointer to a Character object indicating the next or previous Character object node in the list
3. setPrevPtr() / setNextPtr()
 - Takes a pointer to a Character object as a parameter
 - Sets the previous or next pointer to the parameter passed to the function

Game Class

Description: The Game class creates the team lineups and allocates two character objects to represent the Fantasy combat players. The Game class controls the attack and defense rounds representing the combat game, and also outputs the round and final game results to the user.

Game class contains all of the functions controlling the simulation of the tournament. The class contains two Team objects representing the lineup of each team as a doubly-linked list of nodes of Character objects. The class also has one Loser (also DLL) containing the Character object nodes that have lost a battle round. The class has member functions to control the game setup, the creation of new players for the team lineups, the simulation of the tournament, as well as support functions to control each combat round, print the round results and print the final game results.

Key data members:

1. Team team1
2. Team team2
3. Loser loserPile
4. int numPlayers

Key functions:

1. gameSetup()
 - Calls menu function to get the users choice if they want to play the game
 - Calls menu function to get the users choice for the team size

Design Description

Program name : Project4_Design_Description_Schaefer_Kristin.pdf
Author : Kristin Schaefer
Date : 05-26-2019

- Calls menu function to get the users to enter a player name for each individual player
- 2. playTournament()
 - Gets the head object in both team lineups.
 - Calls a function for the attack and defense functions for two attack rounds.
 - Checks which player has incurred more damage or if there was a tie.
 - If there was a tie, the player lineup does not change
 - If there was not a tie, the winner player is moved to the back of its team lineup and the loser player is moved to the loser pile.
 - Checks if either of the team lineups is empty
 - If one of the teams is empty, then a bool value is set and the game loop ends
 - Other wise another round is played.
 - Prints the current round information.
 - Prints the game information after the tournament is over.
 - Asks user if they want to print the loser pile
- 3. printRound()
 - Prints the round results.
- 4. printGame()
 - Prints the game results.

Menu function

Description: The Menu function asks the user if they want to play the Fantasy combat game, which players they want to combat each other and if they want to play the game again.

controls the start menu of the tournament, asking the user for the number of team players per team, asking the user to enter the name of each player, asking the user if they want to print the loser pile and asking the user if they would like to play the tournament again.

Key functions:

1. menuStart()
 - Ask the user if they want to play the game
 - 1. They want to play, start the game
 - 2. They do not want to play, exit the game
2. menuTeamSize()
 - Gets the users choice for the number of players per team
 - Validates values
 - Acceptable range of 1-20 players
3. menuCharacters()
 - Asks the select characters for each player on both teams
 - User can select multiple of the same character type
 - Needs to use the team size input to know how many characters to set
4. menuPrintLoser()
 - Asks the user if they want print the loser pile at the end of the game
 - 1. They want to print the loser pile
 - 2. Don't want to print the loser pile
4. playAgain()
 - Ask the player if they want to play the game again
 - 1. They want to play again, restart the game
 - 2. They do not want to play again, exit the game

Design Description

Program name : Project4_Design_Description_Schaefer_Kristin.pdf
Author : Kristin Schaefer
Date : 05-26-2019

