Project 4 Reflection

Design Description:

Main.cpp:

Call playGame function

functions.cpp

void titleDisplay

void playMenu

void retrieveNumbers

void retrieveTeams

void createTeam1

void createTeam2

void fightMain

void showWinners

void showLosers

void playAgain

character.cpp

Character class:

Protected:

string type;

    double strength;

    double maxStrength;

    int armor;

    int damage;

    int lives;

    string alias;

    int points;

    int \*mobDamage = &damage;

virtual int rollAttackDice

virtual int rollDefenseDice

public:

string getType

    void setArmor

    int getArmor

    void setStrength

    double getStrength

    int getDamage

    void setLives

    int getLives

    void setName

    string getName

    void setPoints

    int getPoints

    Character

    virtual ~Character

    void recover();

    void attack

    int defense();

    void applyDamage(double damage);

    bool isDead();

dice.cpp – Dice Class

protected:

int numSides;

public:

Dice()

Dice(int numSides)

Int rollDie

Static int rollOne

Static int rollTwo

Static int rollThree

Queue.cpp – Queue class

Public:

Struct Node:

Node \*next

Node \*prev

Character \*opponent

Int points

Node \*tail

Node \*head

Queue()

~Queue()

void setPoints

int getPoints

bool isEmpty const

void addHeadNode

void add TailNode

Character\* getFront

Void removeFront

Void moveToLastNode

Vampire.cpp – Vampire class

Vampire()

Dice attack1

Dice defense1

Virtual int rollAttackDice()

Virtual int rollDefenseDice

Bool fiftyFifty

Barbarian.cpp – Barbarian class

Barbarian()

Dice attack1

Dice attack2

Dice defense1

Dice defense2

Virtual int rollAttackDice()

Virtual int rollDefenseDice()

Bluemen.cpp – Bluemen class

Bluemen()

Dice attack1

Dice attack2

Dice defense1

Dice defense2

Dice defense3

Virtual int rollAttackDice()

Virtual int rollDefenseDice()

Medusa.cpp – Medusa class

Medusa()

Dice attack1

Dice attack2

Dice defense1

Virtual int rollAttackDice()

Virtual int rollDefenseDice()

HarryPotter.cpp – HarryPotter class

HarryPotter()

Dice attack2

Dice defense1

Dice defense2

Dice defense3

Virtual int rollAttackDice()

Virtual int rollDefenseDice()

Reflection:

This program was very intense. It was honestly the toughest program I have ever had to code. I became much better with classes and understand inheritance much more. I got much more practice with linked lists and nodes. I got ample practice with pointers and feel like I finally have solid grasp on them. I am honestly impressed by how well I am getting at coding.

I really struggled with getting the Character objects to go into the nodes and then moving them around. That was by far the hardest part of the program for me. It took tons of trial and error. I was probably stuck on it for ~5 hours. I feel like I understand it better but I could definitely use more practice with linked lists. The pointers in linked lists are still confusing to me.

I had to change several functions in the function.cpp file. I had to include the Queue objects for the linked lists which required me to rewrite much of the code in these functions. I had to add functions in character.cpp to track the points for each character object.

The hardest part of the program was getting the linked list to work properly with the character objects.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case** | **Input Values** | **Driver Functions** | **Expected Outcomes** | **Observed Outcomes** |
| If fighter[i] = 1…5 | i | createTeam1 and createTeam2 | Creates correct derived character object | Creates correct derived character object |
| If player1 is dead | Points1 | setPoints | Reduces 1 point from player1’s total points | Reduces 1 point from player1’s total points |
| If player1 is dead | Points2 | setPoints | Increases player2’s total points by 2 | Increases player2’s total points by 2 |
| If player1 is dead | maxStrength | Recover() | Player2 recovers half of maxStrength | Player2 recovers half of maxStrength |
| If player1 is dead | teamPoints2 | setPoints(teamPoints2) | Team2’s total points increase by 2 | Team2’s total points increase by 2 |
| If player1 is dead | teamPoints1 | setPoints(teamPoints1) | Team1’s total points is reduced by 1 | Team1’s total points is reduced by 1 |
| If player1 is dead | Player1 | addHeadNode(player1) | Adds player1 to losers linkedlist | Adds player1 to losers linkedlist |
| If team1 points > team2 points | points | getPoints() | Team 1 wins and displays the winners team and points | Team 1 wins and displays the winners team and points |
| If player1 is dead | Team1 | removeFront() | Removes the character object from team1 linkedlist | Removes the character object from team1 linkedlist |
| If player1 is dead | Team2 | moveToLastNode() | Moves character object from team1 from front of linked list to the back | Moves character object from team1 from front of linked list to the back |
|  |  |  |  |  |