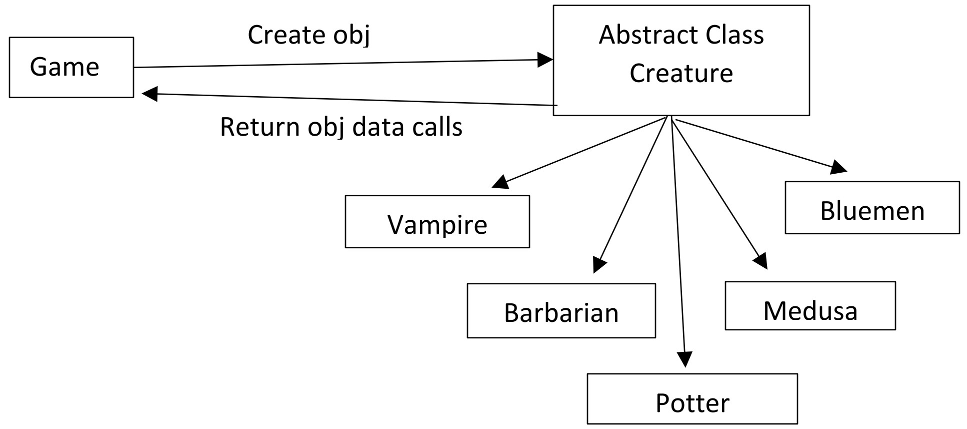
Stephen Townsend

CS 162 – Assignment 4

Design

For the design of Assignment 4 I kept what I had for Assignment 3. For the Tournament rules, I decided to recuse my lab 6 as the assignment stated. I had to do little modification to make it work. I added just a few more functions that would allow me to output what I had in either the Stack or Queue

Assignment 3 Hierarchy



Drivers

Main

menu

1. Vampire
2. Barbarian
3. Blue Men
4. Medusa
5. Harry Potter

Player1.loadCreature();
Player2.loadCreature();
play(player1, plaer2)

Game

Play()
Display()
LoadCreature()

Class

Virtual void attack() = 0;
Virtual void defend() = 0;
Virtual void diceRoll() = 0;

Test Plan

Number of creatures	Team 1	Team 1	Outcome
<Blank>			A default value of 1 will be used.
2			Continue and choose Team names
2a			Please input an integer
3	Red	Blue	The program will use these for names

Reflection

There was not a testing that I needed to perform. My assignment 3 worked as I wanted and made this Assignment a lot easier to finish. All I needed to do was figure a way to implement my lab 6 for the Queue and Stack.

I decided to deal with the Queue and Stack in my main. I did this because This is where I had created the objects. It would be long to have to point them into my game.cpp file. As a result, I did not need to modify much besides my main.cpp. The issues I had was figuring a way to move a creature from the Queue to Stack and also how to push a creature into the back of the Queue after a battle.

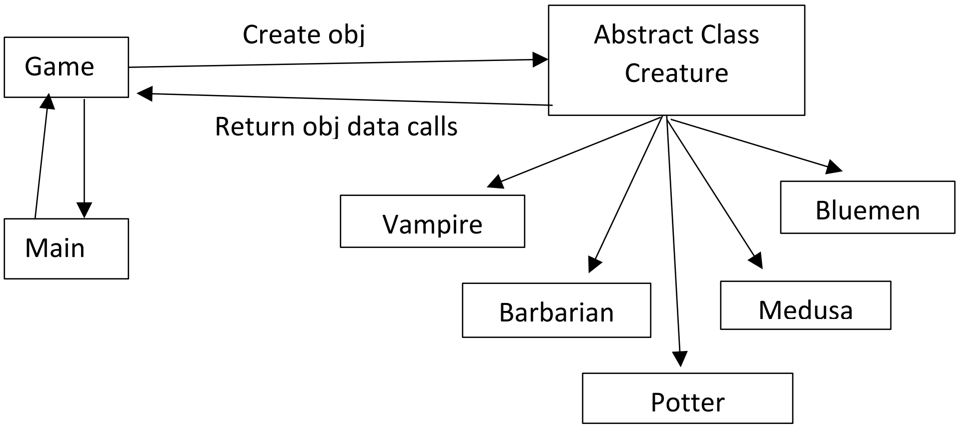
To achieve these two issues, I decided that I could just create two Stack objects; as I had for the Queue. This allowed each player to have their own loser pile. From there I could just push the corresponding looser into the stack using an if statement. If player1 or player 2 is dead, then push into stack and delete from Queue. To place the winning creatures into the back of the queue, I decided the easiest option is to remove the node that held the Creature and then push them back in with their current stats and strength in tacked.

Stephen Townsend

Overall, for the program, I kept the overall general layout of my design and Assignment 3 as described above.

Final

Hierarchy



Drivers

**Changes from original (added) are in blue

```
Main
Enter lineup total

          menu
        1. Vampire
        2. Barbarian
        3. Blue Men
        4. Medusa
        5. Harry Potter
choice = inputVal(choice)
add to queue - Loop
Player1.loadCreature();
Player2.loadCreature();
dead = play(player1, plaer2)
add looser to stack
play again loop
Output looing Stack
Output Winner and winner
loosing pile.
```

```
Game
        Creature *loadCreature(string creature);
        void storyLine(string);
        void play(Creature *player1, Creature *player2);
        void getTombStone(Creature *player);
        void display(Creature*, Creature*);
        string inputVal(string);
        void closeGameData(Creature *. Creature *):
```

```
Class
        virtual string getCreature() = 0;
        virtual int attack() = 0;
        virtual int defense() = 0;
        virtual int getArmor() = 0;
        virtual void isAlive() = 0;
        virtual STATUS getState() = 0;
        virtual int getStrength() = 0;
        virtual int damage(int) = 0;
        virtual int rollDice(int, int, int) = 0;
        virtual int combatRestHealth() = 0;
        virtual SPECIAL special() = 0;
        Creature(); //Default constructor
        Creature(string, int, int, int, int, int, int);
```