David Mednikov

CS 162 W17

onid: mednikod

<p style="text-align:center">Project 3 Reflection</p>

<u>Design</u>

I already had my program backbone from Project 3, so I decided to just make changes to that code than to start from scratch. Obviously, I had to make some changes. First, instead of letting the user decide one creature for each player, I had to let the user decide how many creatures each team would have, and then use a loop to add creatures to each player's team. I decided that I would use vectors for my teams because they already have stack capabilities (like push and pop) built in to them. I didn't see anywhere on Piazza or the assignment that vectors wouldn't be allowed, so for the sake of convenience and my familiarity with them, I decided to implement my teams and dead pile using vectors.

The program also called for being able to set the creature's name, so I had to create overloaded constructors for the base class and all subclasses that would set the name of the creature per the user's input. Because I used string name to store the type of creature in program 3, I had to add a string variable called type to store the type of creature, and then the creature's name could be saved in the name variable. The program also called for a function that restores some health after a creature wins a round, so I created a pure virtual function in the base class and implemented a restoreHealth() function in the subclasses. Depending on what type of creature it was, they would either gain 50% or 100% of their health back. For the sake of fairness, I restored 50% of stronger creatures' health (Vampire, Blue Men, Harry Potter) and for weaker creatures (Medusa and Barbarian) I restored 100% of their health.

I also had to give the user several options, such as letting them decide if they want to see the current score at the end of each round, and letting them decide if they want to see the dead pile at the end of the match. I used y/n input validation that I had used in earlier assignments for this. I also had to implement a way to keep score. I just used the assignment's suggestion (win = +2, loss = -1), and made sure that a score that was negative was automatically changed to 0.

The most noticeable change for me was adding code after a creature's health went down to 0. Before, I would just end the match as soon as that happened and print the result to the screen. Now, I had to print out the results of the round, increment the round counter, update player 1's and player 2's scores, show the current score (if the user wanted to see it after each round), as well as move the creatures either to the dead pile or to the back of their vector. I created a pointer to the dead creature, pushed it to the dead pile, and then removed the [0] index of the team vector. Then, I called

restoreHealth() for the creature that won, created a temp pointer to it, erased it from the [0] index, and pushed it to the back. After doing all this, I checked the size of each team's vector to see if one was out of creatures. If the size of the vector was 0, that meant that all of the team's creatures were in the dead pile and that the match was over. Then I had to find the winner (or see if it was a tie) and print the result to the screen. Last I would give the user the option to play again or not. Either way, I deallocated all of the memory from the tournament before ending the program or starting a new one.

My implementation did not differ from my design at all. I had a vision for how to complete the assignment and implemented it exactly how I had envisioned it.

<u>Test Plan</u>

Expected Outcomes:

From Project 3:

- Vampire will win often thanks to Charm and high strength points
- Blue Men will win the most thanks to a high attack, defense, and armor. Mob doesn't weaken them that much until they are down to 4 strength points
- Harry Potter will win often thanks to Hogwarts ability and high strength points
- Medusa will win if they get lucky (Glare) but their low strength points can cause them to lose despite their high defense and armor
- Barbarian is the weakest one and will lose most often

Project 4:

- A team with stronger creatures will almost always win
- For example, a team with multiple Harry Potters will pretty much never lose since they have the Hogwarts power AND have the restoreHealth() function
- Likewise, a team with multiple Blue Men will also never lose because they have the highest attack and defense of all the characters, despite the Mob weakness. The restoreHealth() function normalizes the Mob effect and makes Blue Men very powerful.
- A team with multiple Barbarians will usually lose, unless they are teamed up with strong creatures. They have lower attack, defense, armor, and strength points, and have no special powers.
- Medusas will usually lose as well. Glare happens pretty infrequently, and certainly not frequently enough for multiple Medusas on the same team to roll a 12.
- A team with multiple vampires will usually do well because of charm. They can dodge a lot of attacks and slowly chip away at the opponent's health points. However, going up against Blue Men and Harry Potters, they will still usually lose.
- A team that has all 5 creatures in the same order (Vampire, Barbarian, Blue Men, Medusa, Harry Potter) will essentially have a 50/50 outcome. However, if one of the teams has a reversed order (HP, Medusa, BM, Barbarian, Vampire) the reverse team will win more often because Harry Potter will get to play more than one round.
- The larger a team is, the stronger characters it needs to win. Small teams can get lucky, but it's hard to get lucky when 3 or more characters need to die before a team can win.

<u>Test Plan</u>

- For the sake of saving space and clutter I will not be including Project 3 test plan

Test Case	Team 1	Team 2	Expected Outcomes	Observed Outcomes
1	Vampire, HP, BM, Vampire	Medusa, Vampire, Barbarian, Vampire	Team 1 wins easily	Team 1 won 7-1 (4 rounds won, 1 round lost) The only round won by team 2 was the 4th round, Vampire vs. Vampire
2	Vampire, Barbarian, BM, Medusa, HP	Vampire, Barbarian, BM, Medusa, HP	A close match, around 5-5 final	Team 1 was winning easily (5-1 at some point) but Team 2 still had blue men and harry potter and ended up winning 9-1. What a comeback!
3	Vampire, Barbarian, BM, Medusa, HP	HP, Medusa, BM, Barbarian, Vampire	Team 2 wins	Team 1 won 9-1. Again, Team 2 had an early lead, but once Team 1 was down to just stronger characters, they won easily
4	BM, Medusa, BM	Medusa, BM, Medusa	Team 1 wins easily	Team 1 won 6-0, 3 rounds and it was over. Multiple Blue Men is a huge advantage
5	Barbarian, HP, Barbarian, HP	Vampire, Medusa, Vampire, Medusa	Team 1 wins easily	Team 1 won 8-2. Multiple Harry Potters is also a huge advantage, and Medusas are weak
6	BM, Medusa, Vampire	HP, Vampire, Medusa	Close match that Team 2 wins in the end	Team 1 won easily, 3 rounds and it was over (6-0).

<u>Test Results</u>

My test results did not exactly match my expectations. Mainly because the more creatures there are, the more possibilities there are.

- Test Case 1: My expected outcome happened, because Harry Potter and Blue Men made Team 1 stronger.
- Test Case 2: My expected outcome was off from what actually happened, both teams were the same but as the match wore on Team 2 was able to win some rounds in a row and got the win.
- Test Case 3: My expected outcome was completely different from the actual outcome. Team 1 ended up winning easily, mainly because after the Barbarian died, most of Team 1's creatures were powerful.
- Test Case 4: My expected outcome happened, the team with more Blue Men won easily.
- Test Case 5: My expected outcome happened, the team with a strong character (HP) won easily.
- Test Case 6: My expected outcome differed from the results. I expected Harry Potter to beat Blue Men and Team 2 to win out. But Blue Men defeated Harry Potter and rolled to the W.