Werewolves vs Vampires Game

Μιχάλης Μιχόπουλος 111520170091
Βασίλης Μεντζίνης 111520800116

Commands in order to run the programm:

cd src
make clean
make
./game <**dimension1**> <**dimension2**> <**Avatar team**>

dimension1 : map dimension
dimension2: map dimension
Avatar team : W or V (W for werewolves V for Vampires).

e.x) ./game 20 20 W , ./game 15 15 V , ./game 22 22 W . e.t.c

Basic info.

1) The game is turned based . When the player moves every creature is doing their move concurrently.

2) The player can do the moves described below.

w + Enter : move avatar upwards.
s + Enter : move avatar downwards.
a + Enter : move avatar to the left.
d + Enter : move avatar to the right.
Space + Enter : pause/resume the game (also displayings current stats)
h + Enter : avatar heals his team.

3) Map Display :

Trees are represented by '^'.
Water is represented by '~'.
Land by this ' '.
Potion available for avatar is represented by 'P'.
Vampires are represented by 'B'.

Werewolves are represented by 'L'.

Avatar is represented by 'W'/'V' (based on his team)

Implementation:

The core functionality of the game is being delivered by the classes listed below.

1) The ***Game*** class implemented in src/game.cpp.
2) The ***Grid*** class implemented in src/grid.cpp.
3) The ***Creature*** class implemented in src/creature.cpp.

Game:

The game class implements the basic functionality of the game.
It's the enginee of the game having every Creature and the Map in it's attributes.

In the list below will be covered the most important functions of the class.

1)InitializeGame:

This function implements the game set-up.It generates the map.
It creates every objective and Creature and set's them on the grid of the map properly

2)BeingsEngagement:

This function implements every creature interaction(from game's perspective) at the current game-state and positioning of the creatures. These interactions include attacking , healing , escaping from the enemy etc.

3) PrintStats:
This function prints the game's current stats when the player pauses or the game ends.

4) GamePlay:

This function implements the “playing loop” of the game.Catches every move of the player and the game responds accordingly.When the player moves the game is changing state and the gamePlay function calls the beingEngament function discussed above.Prints stats when the player wants to by calling PrintStats and it is also responsible for Checking the ending condition of the game and the changing the day/night state of the game.

Grid:

The Grid class implements everything that has to do with the map.
Implements the the grid of the map and it’s functionality.

In the list below will be covered the most important functions of the class.

1) Grid (x , y)

Initializes a grid of tiles (x*y dimensions) that will be used as the map of the game.A **tile** is how we implemented the pieces of the map. It serves the purpose of keeping the basic info of every position(i,j) on the map.

2)SetBeing

This function simply puts a creature on legal position on the map.

3)display

This function displays the map on the user .

Creature:

The creature class implements the basic functionality of every being of the game .It is the base class that the derived classes **Avatar , Vampire, Werewolfe** inherit their main functionality from.

It keeps the main info tha every being has on this game . Such as the current position of the being , stats and current state of health etc.

It implements the accessors and mutators of a Creature’s vital info for the game like health, potions, shield and power. (And the decrementation or incrementation of those if needed.

It is also responsible for the acts of attacking an enemy or healing an ally respectively or evading from the enemy.

Functions like move up/down/left/right needed in every creature are also implemented here.These are functions used by the derived classes inside the **pure virtual function move** which is implemented differently in every type of being because of it's different nature of moving on the map.So the move function is a pure virtual declared in Creature Class.

Virtual Functions implemented in Creature Class are the functions

1) heal:
   we decided for heal to be a virtual function because of the different nature of healing in the implementation of avatar in contrast with the other creatures because the avatar creatures heals all his allies. (so the heal function must be overridden)
2) is_legal_move

   we decided this because the creatures have a different way of moving so
   the calculations of moves that are legal is also different and in some cases the base class calculations of legal loves must be overridden from a specific creature.