Foreword

Play the game, I recommend playing against the Hoard, Have fun. Afterward if you are wondering why I did something, below are my general reasoning for most of my choices and my general design. Honestly I don't think you need to read it but I wanted to lay out my ideas anyway.

Why Hexagons-

First please watch this to understand - https://www.youtube.com/watch?v=thOifuHs6eY.
Now you know Hexagons are the Bestagons. Besides that point I do feel like diagonal maneuvering is much more fluid within hexagons and makes much more sense. This results in what I feel like is a more accurate battlefield map. The number one difficulty I have seen from a user design perspective for most games is that this type of hexagonal movement cannot be mapped easily to arrow keys. This is not a difficulty with the strategy genre in general and has been done wonderfully in Civ series with their hex tile map. The difficulty for me was coding all of this up. Sadly as I shall go over in a bit there is no support for hexagon anything and so everything was done custom by me.

Game Flow-

I have played quite a few RTS(Real time strategy games) and while I love them I always felt that mechanical skill outranks strategy in almost all the popular RTS. I did not want this and more specifically I did not want this RTS game to be extremely micro-management heavy. Yet the difficulty was that I also wanted tile based unit production. Thus originally I decided everything would be nice and slow and that would cause micro not to be an issue. Then I played it and it was very boring, then boring, then very micro heavy, you just waited around a lot early game till later the amount of castles and units explodes and there is a ton of micro. I honestly don't know what the answer is. Perhaps micro is needed in RTSs so that the early game is not so boring. Anyway a lot of the pacing and scaling had to be adjusted along with all the AI(sadly time consuming) to make the beginning only slightly boring and the late game a little less micro heavy.

Rendering-

THERE IS NO LIBRARY TO DRAW NICE ALIGNED HEXAGONS :(. Thus I had to just draw my own by simply drawing the correct lines(alot of math not done well elsewhere).

Foreword on the AI -

This game took a long time to make(certainly more than it should have) and I had a lot of ambition for a lot of extra functionality so I wrote a lot of frameworks that are in the game such as dynamic terrain that simply was not included because I simply did not have time to test it against all the different AI. But look through the code and you will find everything is built to be extended and is ready for so much more functionality. Also there were alot of small bits of logic added everywhere such as health regen and building level affecting damage taken and beyond the big sections mentioned below. I find all these small bits of logic and quality of life improvements add up to alot.

General AI Design

The number one principle in my AI design was - create abstract functions that did cool things. Then test these functions and then make simple AI functionality that could use them and then build bigger AI on top of this. This allowed for the Onion design discussed in class, so that I could always add functionality and revert to the last working functionality if I ran out of time. In the end while I did not get all the functionality I wanted I was able to get everything promised in the proposal working and a lot of extra systems working that can really easily be extended.

Unit Pathing -

Units needed to be able to go up and fight each other and needed to be able to navigate the hexagonal tile map. The way I resolved this was to redo the waypoint system such that they will only attack a target within the same hexagon. The second was for each unit to hold its own path and when it is not fighting follow this path. The path finding was done using A* using a distance as the heuristic. The difficulty is that the neighbors on not constant and there is an offset based on the y value thus the A* had to be completely custom. This A* system can be used to avoid dynamic terrain but this system is untested in relation to the personality AI.

AI Personality Difficulty System-

I feel difficulty affecting in-game scaling is just a bad approach. Because in the end comradery over difficulty is what I think builds community in the absence of a gripping story in an offline setting and if everyone has a completely different experience I don't feel like this is possible. I wanted to reflect this thus instead of difficulty determining some in game property difficulty is the AI personalities.

Personality Basic Functionality -

The personality AI is designed with an onion approach and thus has multiple levels of abstract classes and working functionality. The most basic version had a lot of functionality. This included first a strategic overlay of the game that every tick went through and evaluated the board gathering all information needed for the AI. Second a decaying queue based system that evaluated every single tile for defensive and offensive value. The evaluation was an abstract method that would be overridden by most AI that would use it to decide where to move or where to build or just to do quick analysis of the whole board. Third all AI had had access to and used a dynamic functionality to output voice lines to the user. On top of this a very large amount of functions also existed in this abstract class(persona) that allowed for many different ways to interact and change the board. These were used in the high level decision trees implemented by every personality. The Personalities with only these basic utilities are the ZergRush, Fat Bastard and Fat God. I also think the voice lines are fun, read through a few of them if you get the chance. I do think just a bit of personality in the AI in all aspects goes a long way towards player enjoyment.

Personality Mindset -

The second onion level for the personalities was the mindset personality exemplified within its own abstract class. These personalities had existential mindsets and functioned less like decision trees and more like finite state machines with the two primary states being offense and defense. This allowed for a lot of really cool human-like interaction with my personal favorite, the retreat, if the player actually gets their forces together on time. It really allows for just much more engaging AI.

Personality Extendability-

As almost all the functionality is abstracted in abstract classes so creating new personalities is a breeze.

Lessons Learned-

My greatest problem was that I designed around what I wanted for the AI not based on the existing code and that is what killed me. In this I have learned great respect for the middleware companies that have to deal with existing code bases and grand AI ideas at the same time. Though I really do feel like this was all a great learning experience and I do think it is a good game all things considered.

Final comment-

I hope you enjoyed playing and thank you for a great semester.