

Harry Nystad
Neuton
Game: Frank Darkhawks Maze

I have been working all around the project, but one of the things i did single handedly was the menu. What i worked on the was first and foremost string formating. For instance with the IP address and number of players, and i got to work a little with sprite animation, animating the spinning planet in the background of the menu. In addition I implemented all the sound i the game, although this simply was to convert the files to a format that SFML could read and then write one line of code to load in the file, and one line to start or stop the sound.

I liked the SFML music library uses its own thread, so after hitting play i didn't have to worry about the sound blocking the rest of the program.

other parts of the project i worked on was the characters. We started of making one character object and then having player and non-player character inherent form the character class put we got some problems with the factory model we whent for, for storing characters. To fix this problem we scraped the player and non-player class and started to work on on single character type that used different types of controllers. the non-player controller got the pathfinding, and player controller got an inputhandler that told the character class what to do.

The AI controller was originally design the same way as the player controller just to take the inputs from some sort of list, but was afterwards redone to fit better when the a-star pathfinding got plase.

the way we design the flow of the program we got some problems when trying to thread the update loop. I used a lot of time reading on threads to make the for loop parallel and thread safe. this was difficult because all the information was stored inside objects and I didn't have access to everything. And after several tries, ending up with locking almost the whole for loop we ended up with removing the threading from this part of the code. One of the thing i learn by talking to people who knew more about this and by reading on the interwebs, was that object oriented programing is not the best way to do multi threading

looking back at world::update it wasn't the best idea. the action limiter is an example of this. To limit the time between each movement. this ended up with restricting the input from keyboard to only register every 100 milliseconds. This is not a problem if you are holding the keys down, but if you are doing sort quick pushes on the keys, occasionally the key won't register. it would have been nice to have the characters know their position in the world, and have a separate input handler (outside the update loop). but this was pushed back and that was needed to do everything that was "wrong" with the update function would have been so time consuming so it was never dealt with as we where is a so late part of the project at this time.

I believe we all have at one point we all have had one version on placeCharater in world.cpp. It is not really something that have been taken out of the game it have always build upon the old version, but it have been interesting to see how this function have evolved when we added new

content to the game. for instance adding 2 players we no longer could place the character in the top left corner. when mazes was introduced we had to take walls in consideration, and so one.

just to mention some minor things i have worked on:

Point: system, adding subtracting points

Healt: adding and removing hitpoints.

the Goal flag

world resetting: when you want to start a new game without exiting the game first

some other things i have learned is:

Some more insight into the world of licensing and copyright by working with Thomas who is very straight with this.

And i have learn more about version control, manly using git