Grace Mao, Amanda Zheng, and George Zhou Team GAG
1/2/20
Pd 10

Final Project Proposal

Description

For our final project, we will be creating a multiplayer Mafia game in the terminal. Up to 12 people can play, but once a game has started, no new players may join. At the start, roles (mafia, doctor, police, etc.) will be distributed randomly, and all other players will remain citizens. The number of mafias will be determined by how many players there are. At the beginning of each round called the “night” phase, the mafia will decide who to kill, the doctor will decide who to heal, and the police can find out one player’s role (might be more roles)... After that, dead players will only be able to view the game, not participate. After that phase of the game, there will be a minute for a debate of who people think is part of the mafia. Then, each player votes for a player that they think are the mafia for a minute. If any person gets the majority of the votes, the voting goes further: There will be a minute for players to vote (or abstain) if they think the person should be executed. If that player gets a majority of yeses, then he shall be executed. Its role will not be displayed. The game ends when either all mafia die or mafia outnumber the citizens.

User Experience

A user will either join a server or open a server. After the minimum amount of people are on the server, the host can choose to start game or wait for more players. Then everyone enters a username and gets a role displayed only to themselves. Next, the night round commences and one person of each special role vote for a person of their choice (to kill, to heal, or check identity depending on the role of the person). Then, in the morning, the results are announced. Every user can debate in a forum-like chat. There will be time warnings of how long they have to chat (1 minute, 30 seconds, time’s up!). Then everyone will vote for someone to be executed (ties means no one dies). Another night round continues and the round keeps going until all mafia members are dead or mafia outnumbers the innocent people. The people who died before the game ends can choose to spectate the game or leave the game.

Example:

```
\Mafia$ Waiting for players to Join…
\Mafia$ 5 players in game, start? (y/n)
\Mafia$ Waiting for other players to Join
\Mafia$ Enter Username : dan_
Your Username is: dandymandy
\Mafia$ Waiting for other players...
In game: horhay63, maobao123, dandymandy (you), mabelline, dwdwdwdw, apples111
\Mafia$ Generating Role…
         Your Role: Mafia
Night Round Begins!
Mafia Please Vote for Your Victim
------------------------------------------------//shown only to this special role
Other Mafias: horhay63
\Mafia$ maybell_
You voted for maybelline!
Next Mafia to vote: horhay63
\Mafia$ Waiting for Doctor to select someone to save…
\Mafia$ Waiting for Police to select someone to save…
\Mafia$ Night Ended!
\Mafia$ maybelline was killed! Discuss who you think is the mafia! You have 60 seconds.
Horhay63: okay guys who do u think it is
Maobao123:the most sus guy is the one that talks first
\Mafia$: But what if you a_
Horhay63: okay guys who do u think it is
Maobao123: the most sus guy is the one that talks first
You: But what if you are using reverse psychology?
Times Up!
Vote for who you think is the mafia!
\Mafia$ maobao123 died :( The next night is beginning…
```

Technical Design

Topics Implemented:

1. Signaling
2. Semaphores
3. Networking
4. Pipes
5. Forking

SIGNALING

We will use signaling to stop and start the forking for the server in order to begin the game. A signal will be sent to the other computers to start the game and the signal will be sent to the other computers to stop the game.

SEMAPHORES

We will use semaphores so that during the “night” phase, the mafias and each player will have to take turns to perform their action. Also, all players with similar roles have to choose a victim at the same time, so we will need semaphores for that.

NETWORKING

We will need networking because this is a multiplayer game (not pass and play version) so everyone will get their computer and join into a game like a server. This requires networking to coordinate all the computers together with the game.

PIPES

We will need pipes for the chat box. The input text that one user types in should show up on the other computers and should be printed out on all the terminals that are in this game.

FORK

We will need forking so that we can fork a server-client off of the main client so that we have a more organized code. A forked child will get all the data from the main parent and if it messes up, we can make the child die and fork again and revert to the original data.

Development Stages

I. Modular Setup and Data Structures
    A. Create any necessary files, including makefiles, header files etc
    B. Code in #includes and other necessary definitions
    C. Decide on function headers and required functionality methods
    D. Initialize data structures needed, can be added onto later

II. Game Shell
    A. Open shell every time game is run, ask to start after 5 players have joined
    B. Semaphore will only allow up to 12 players to join, nobody will be able to join after the game begins
    C. Writing in-game shell before prompted should have no effect

III. Initial game setup
    A. Implement shell functionality of organizing the game and simulating the different rounds
    B. Some text-only visible for certain people
    C. Voting system should work with a tie system etc

IV. Networking
    A. Multiple people should be able to access the server and play together

V. Extra Features (with more time)
    A. Mafia will be able to communicate with each other during the Night Round in order to decide who they will kill
    B. With a tie, players will be able to revote between the ones voted on in the tie

Data Structures and Algorithms

Structures:

- *char[] roles
    - The list of roles at the beginning of the game.
- *char[] players
    - The list of nicknames. The indices correspond to the roles in the arrays
- int[] votes
    - Gives a tally of the number of votes each person has (you can't change your vote by the way)
- int num_players
    - At the beginning of the game, keeps track of how many players are on the server.
- int[] status
    - List of booleans that tell which person is alive or dead
- int timer
    - Is used for situations in which time is involved.
- int days
    - The number of days that have passed
- *char
    - A String that will distinguish what time of day it currently is; it cycles from debate, voting, yes/no voting, execution, doctor healing, mafia debate, and mafia voting/killing
- *char message
    - Helps for messaging, which is described below in algorithms
- int has_message
    - Boolean that says whether there is an impending message from anybody

Algorithms:

- Timer
    - We will need a timer to keep track of the amount of time that has elapsed based on a given point. It will be used to time debates and the amount of time the roles are given for the night.
- Chat Piping
    - We will need to interpret the information given by pipes. This allows secret conversations between people, whether they are in the mafia or not. It will create a pipe starting from the person who wants to initiate a *secret* and then the end at the person who it wants to be sent to. Then, it must be formatted correctly to be displayed in the recipient's terminal.

Timeline

JANUARY 2	Proposal Due
JANUARY 4	Modular Setup with Data Structures
JANUARY 6	Set up Mafia shell (complete with working semaphore)
JANUARY 8	Simulate Night Round of the game
JANUARY 10	Distribution of roles finished with correct number of Mafia by number of players
JANUARY 13	Create chatting functionality after night round using pipes
JANUARY 14	Voting system finished if not finished before
JANUARY 15	Finish networking and start final debugging
JANUARY 17	Final Project Due

Work Division

Legend	
H	Hard
M	Medium
E	Easy

George:

- Networking (H)
- Police Role (M)
- Sharing memory across computers (M)
- Username Controls (E)
- End Game (E)

Amanda:

- Mafia Role (H)
- Creating Server (M)
- Vote (M)
- Timer (E)
- Quit Game (E)

Grace:

- Chat Room (H)
- Doctor Role (M)
- Allowing people to join server (M)
- Terminal Updates (E)
- Spectate mode (E)