

Werewolf Game Server

...

Powered by
Eden Wu & Cecilia Cai



A Game Requires a good
GOD and LOTS OF PLAYERS

No one want to
be god!

What if we get rid of the GOD

And allow people to play it not face
to face

Or having a perfectly justice and fair
god by a programm?



KEEP
CALM
AND
KILL
GOD

Werewolf Game Server

Beta 1.0.1
New Features

- Werewolves, Prophet, Witch, Villagers
 - 4 - 10 players gaming together
 - Multi-terminal game
 - Completely trustworthy (no cheating allowed)
 - Fixed all bugs! Yeah, every single one of em.
-

Try our werewolf game server!

- Get our project at
<https://github.com/Cceciliaa/ICS-final-project>
- Clone to your computer
- Open the terminal, cd to the project folder
- Type: `python chat_cmdl_client.py -d 10.209.10.115`
- Wait for enough players to start the game!

Recorded Demo of Six players



Psudo Code

Here come the logical parts Yay...

#0 Players(Character)

Class Players is a subclass of Character

self.role ["villager", "wolf", "prophet", "witch"]

self.state["alive", "dead"]

self.poison[0, 1]

self.cure[0, 1]

self.gaming_state["action", "asleep",
"discussion"] (not allowed to talk during asleep)

#1 Role Assign

Type “g” + player name to enter a game lodge

(log the players into the Player(Character) class)

State = S_GAMING

Type “start”

If $4 \leq \text{player number} \leq 10$: Start game

Assign roles from roles(dictionary) to players

Player.state = “alive”

Player.gaming_state = “asleep”

Witch set poison and cure to 1

~~Go to~~ state “start”

#2 Wolves

“round” = “start”

Set wolves' gaming_state into “action”

Wolves get into the group_wolves

List all players that are “alive”

Members in group_wolves chat secretly together

Type ‘KILL’ + playerName to change player.state into “dead”

Set wolves' gaming_state into “asleep”

Send message to server “round” = “check”

#3 Prophet

“round” = “check”

Set prophet’s gaming_state into “action”

List all in game players

Type “CHECK” + playerName returns
player.role

Set prophet’s gaming_state into “asleep”

Send message to server “round” = “poison”

#4 Witch(Poison)

“round” = “poison”

Set witch’s gaming_state into “action”

List all in game players

Type “POISON” + playerName to set player.state to “dead”

Set poison to 0

Or type “SKIPP” to skip

Send message to server “round” = “cure”

#5 Witch(Cure)

“round” = “cure”

List the player that just dead

Type “CURE” + playerName to set player.state to “alive”

Set cure to 0

Or type “SKIPC” to skip

Send message to server “round” = “discussion”

#6 Discussion

“round” = “discussion”

Players are free to talk to each other

Type “FIN” to change the “round” to “poll”

#7 Poll

“round” = “poll”

self.poll = {playerName: 0 for player in
gamingplayers}

If message == playerName

self.poll[playerName] += 1

If everyone voted and there's no tied vote

self.judge _victory()

#8

Judge Victory

Loop through the gaming player list, find the players whose status is “alive”, add their names and their respective roles to a dictionary {[name: role]...}

Loop through the values in the dictionary:

If there is no wolf, villagers win;

If there is no villagers or no special characters, wolves win;

Otherwise, returns “continue”

#8

Judge Victory

Judge the result every time entering discussion and finishing poll

Pass the current gaming group in server to the `judge_result()` function in the `Players` class, which will return “continue” or the winning side

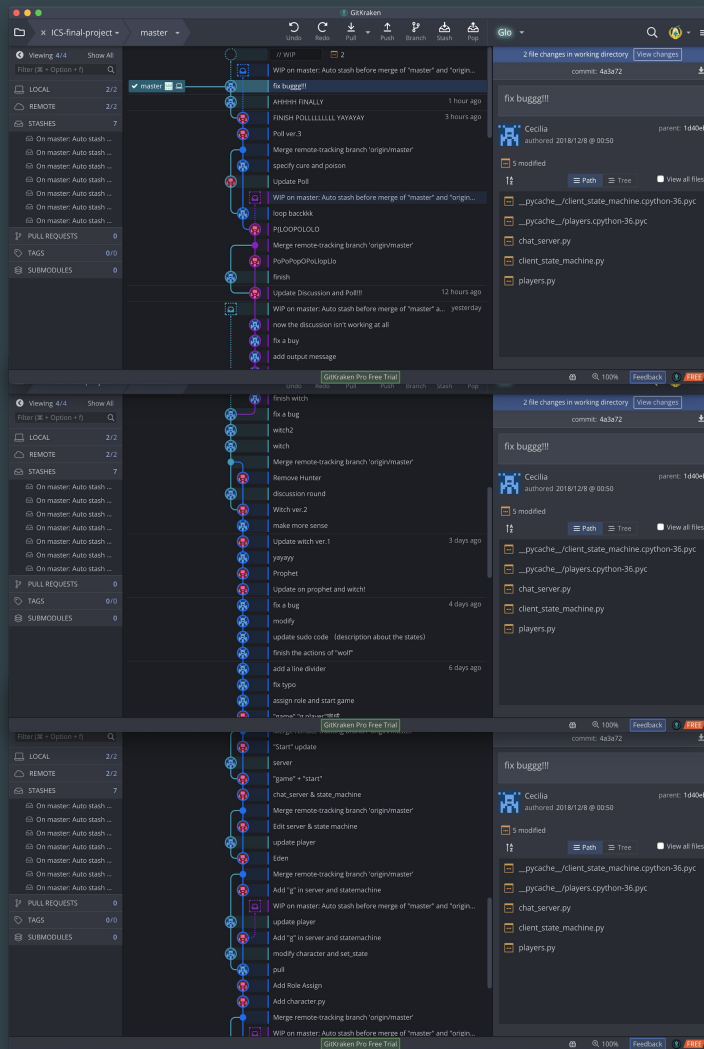
If returns “continue”, continue the game and move on to the next state

Elif receive the winning side, call the game to an end and print the winning side on each client’s window

1200+

Lines of total codes

Collaborate through Github



**“Even the seemingly
simplest game contains a
HELL lot of logic!!!”**

Thank You!!!

By Eden Wu & Cecilia Cai
Special thanks to Wen Yin for the helps

Introduction to Computer Science
December 13th