CSC307 PlanItpoker Project
Members: Adrian, Marco, Aidan
UML DIAGRAM: here

Stories

1. As a player, I want to create a new game room with a unique ID so that I can invite others to join and start a planning session
2. As a player, I want to join an existing room using a room ID, so that I can participate in the same session as my team
3. As a player, I want to send a card with my estimate, so that the team can use it for planning
4. As a player, I want to trigger a reveal of all players' cards, so that we could compare estimates
5. As a player, I want to view a summary of everyone's estimates, so that they could all be reviewed
6. As a player, I want to send and receive chat messages, so that I can communicate with other players during the session

Classes

class Main: Launches the program, parses arguments (room ID or name), and connects components (UI and MQTT)

class Repository: Holds game state data (player list, room ID, card submissions, game state, chat history), observed by GUI, updated by MQTT events, and sends changes via publisher.

class Publisher: Sending messages (estimates, reveals, chats)and wraps MQTT client for publishing under correct topics

class Subscriber: Listens for messages on MQTT

class MessageHandler: Decodes and dispatches MQTT messages

class GameRoom: holds details of the current room details like ID, player list, state, and estimates

class Player: name, estimate, ID

class Card: Face up/down to show estimate value and status, can we decorated using decorator for highlighted and selected

class CardDecorator^^

class ChatPanel: Chat UI for sending/receiving messages via publisher/ subscriber

class MainWindow: JFrame that holds all the panels and allows for switching views

class RoomPanel: Shows the players in the room and room ID

class CardPanel: Displays the cards for each player while observing repository

class ControlPanel: Holds all the game action buttons (Send estimate, reveal)

MAYBES
class ScoreBoardPanel: summary after revealing

class PlayerGroup: Handles groups of players using Composite design pattern