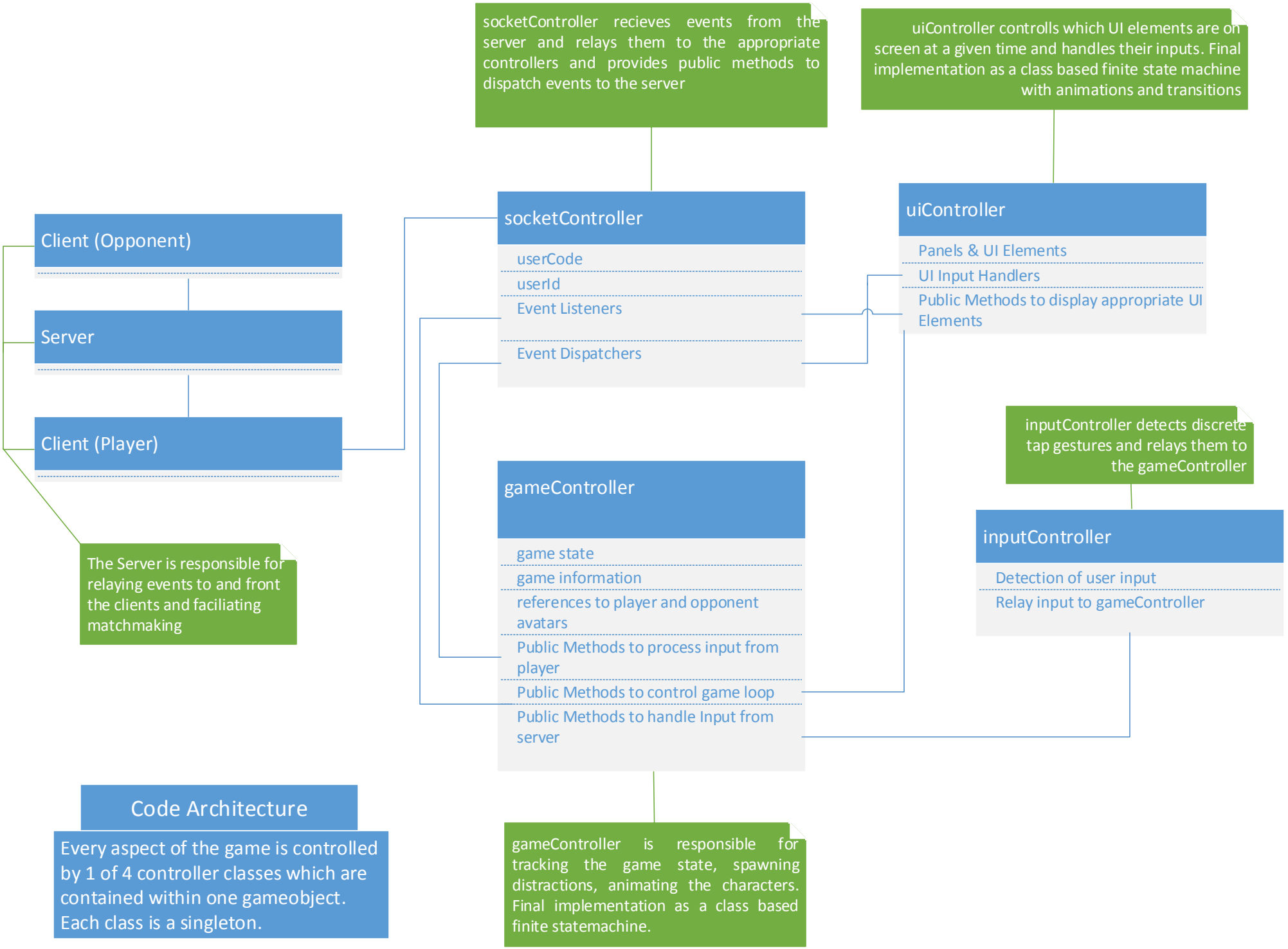
socketController recieves events from the server and relays them to the appropriate controllers and provides public methods to dispatch events to the server
uiController controlls which UI elements are on screen at a given time and handles their inputs. Final implementation as a class based finite state machine with animations and transitions
socketController
userCode
userId
Event Listeners
Event Dispatchers
uiController
Panels & UI Elements
UI Input Handlers
Public Methods to display appropriate UI Elements
Client (Opponent)
Server
Client (Player)
The Server is responsible for relaying events to and front the clients and faciliating matchmaking
gameController
game state
game information
references to player and opponent avatars
Public Methods to process input from player
Public Methods to control game loop
Public Methods to handle Input from server
inputController detects discrete tap gestures and relays them to the gameController
inputController
Detection of user input
Relay input to gameController
Code Architecture
Every aspect of the game is controlled by 1 of 4 controller classes which are contained within one gameobject. Each class is a singleton.
gameController is responsible for tracking the game state, spawning distractions, animating the characters. Final implementation as a class based finite statemachine.