

Application architecture overview

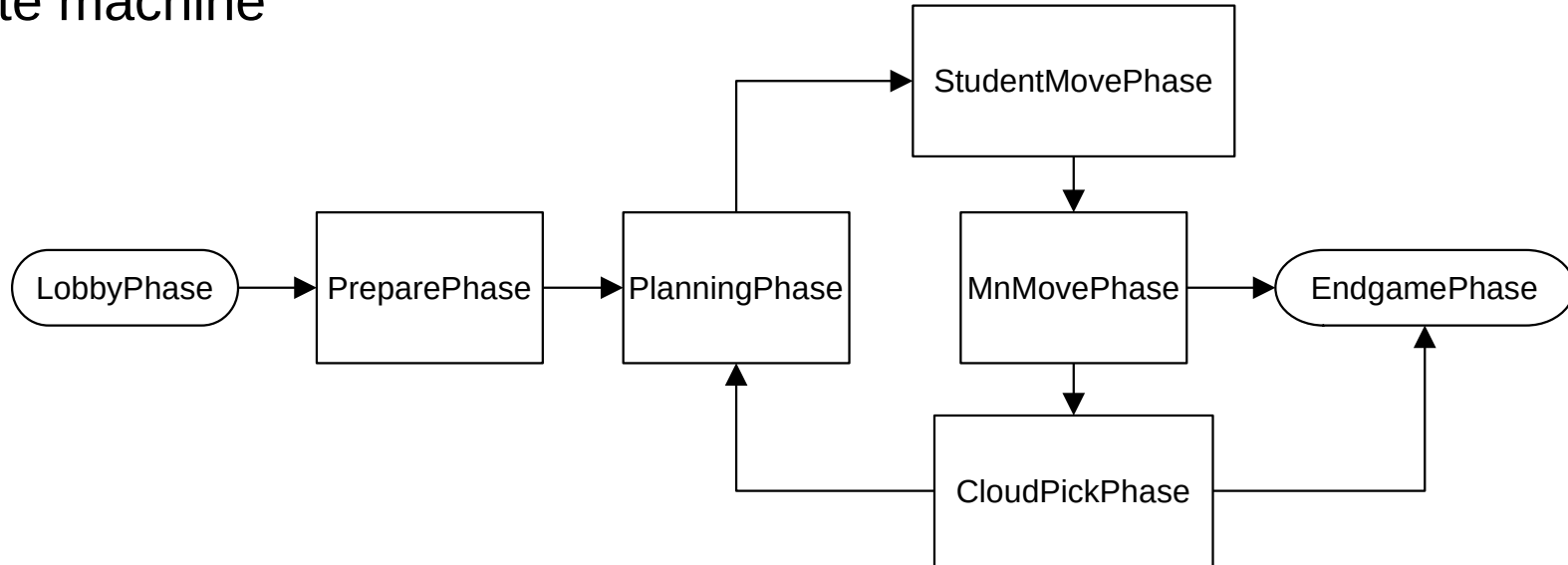
Group AM36: Bianconi, Bradatan, Busso

Package hierarchy

- MVC pattern with split controller
- `server`
 - `server.model`: game model
 - `server.controller`: server side controller
 - `server.controller.persistence`: persistence management
 - `server.net`: networking
- `client`
 - `client.control`: client side controller
 - `client.view`: views
- `functional, enums`: support utilities

server.model

- Immutable design
 - Simple multithreading
 - No partial update problems
- State machine

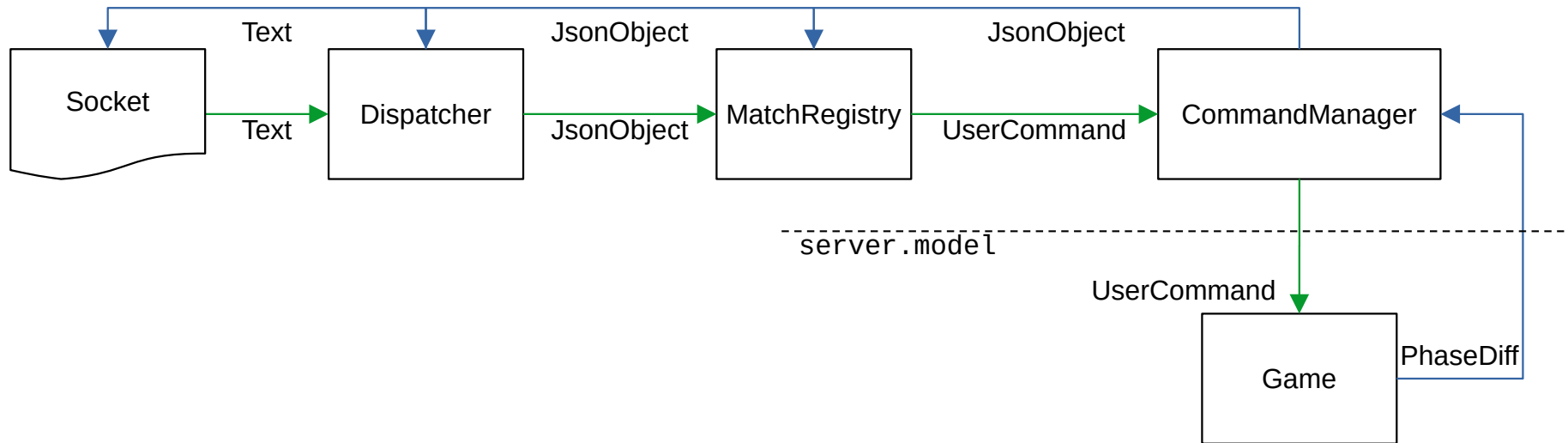


server.model

- Each phase exposes a specialized API in addition to the generic Phase one
 - For example, ActionPhase precursor of {StudentMove,MnMove,CloudPick}Phase exposes a generic state-modification API usable by other package members
- Characters are implemented as “plug-ins” to the Phase state machine
 - Easy implemetation of new characters:
 - extend Character
 - implement doEffect()
 - add an entry to ChracterFactory

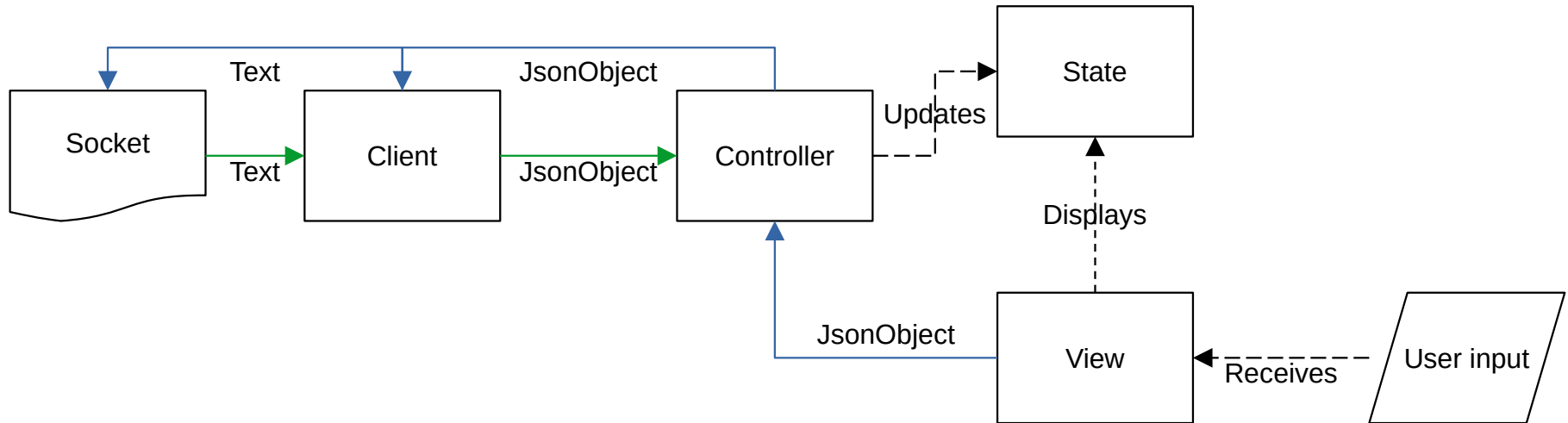
server.controller

- Pipeline-like design
- Multithreaded
 - Each match has its own thread and Command queue



client

- Partial controller interfaces with a generic View interface
 - Concrete implementations are specific to CLI and GUI
- Similar, but stripped down, design as the server counterpart



Persistence

- File-based database
 - Each record is a JSON dump of the model
 - Problem: requires special reflection tricks to correctly serialize/deserialize to/from JSON
- Each state commit is done after each successful user command
- Server restores files on startup and marks restored matches
 - Clients need to rejoin the match with the same id with the same usernames
 - If all clients rejoined, the match continues as normal

Communication protocol

- JSON UTF8 based
- Request-response based
 - Client sends command requests
 - Server responds with either ACKs or NACKs
 - Client waits for a server response before sending a new request
- Minimal number of message types
 - 1 message for each possible user action
- All types have a similar structure
 - Easy deserialization