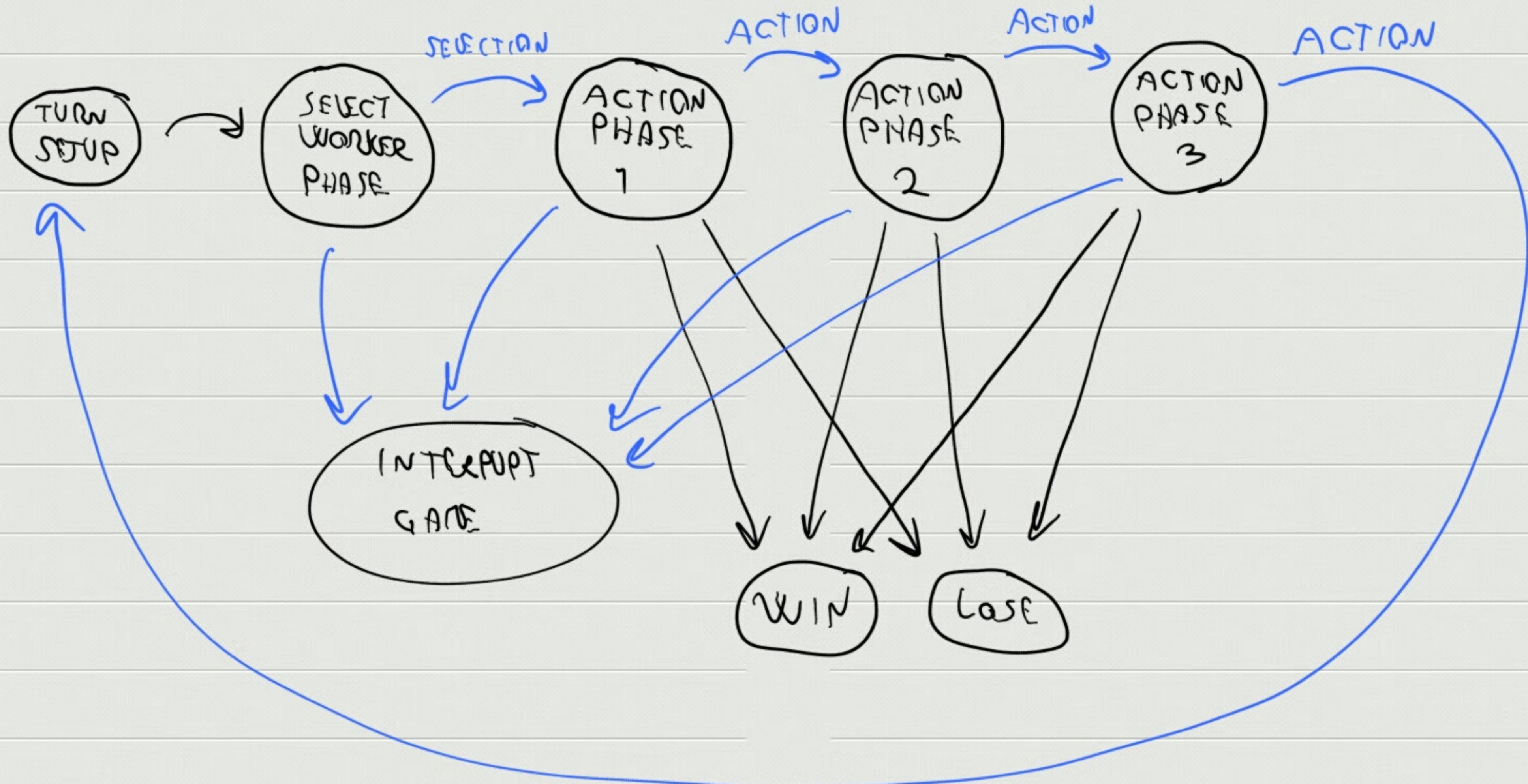
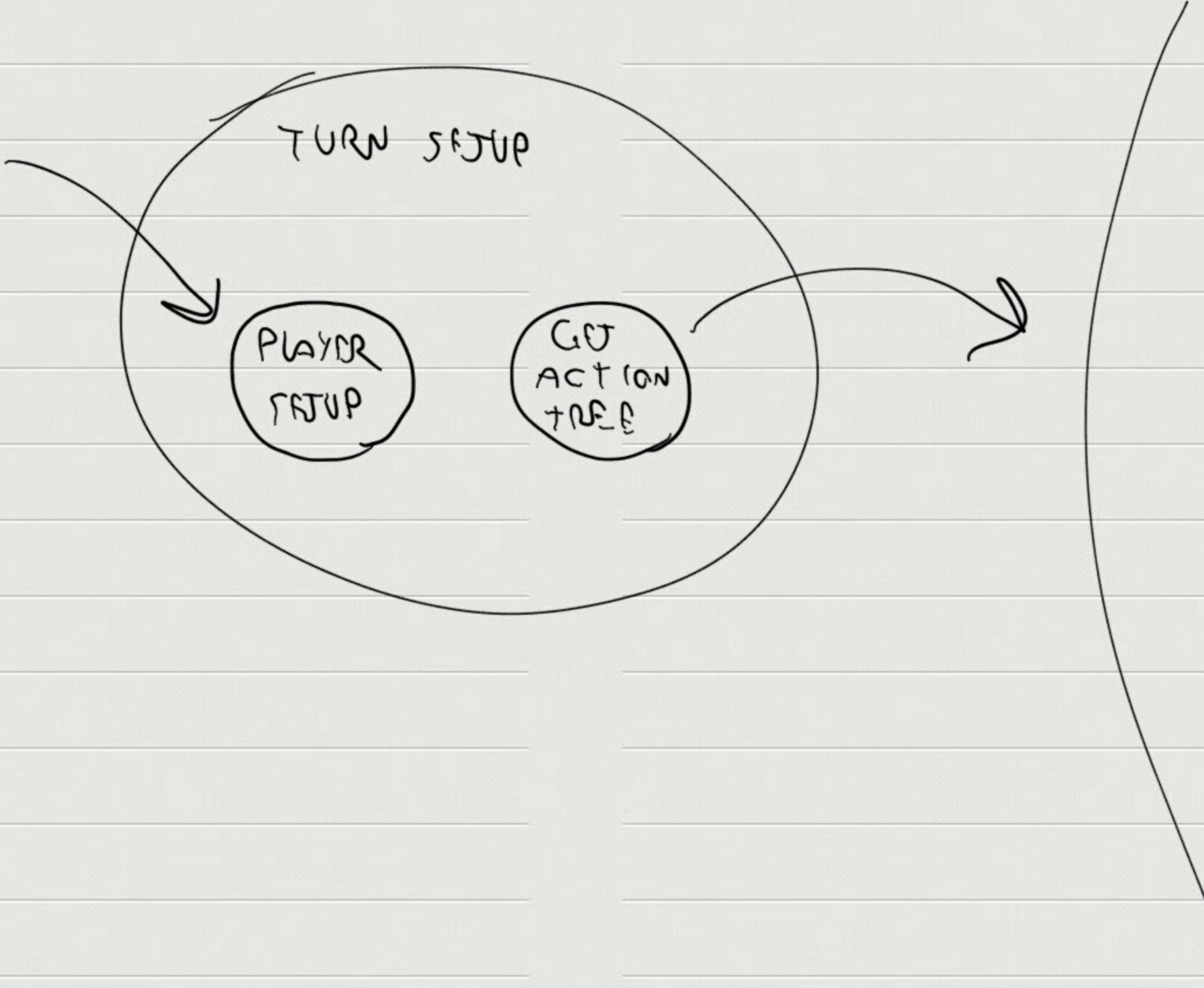


TURN FLOW



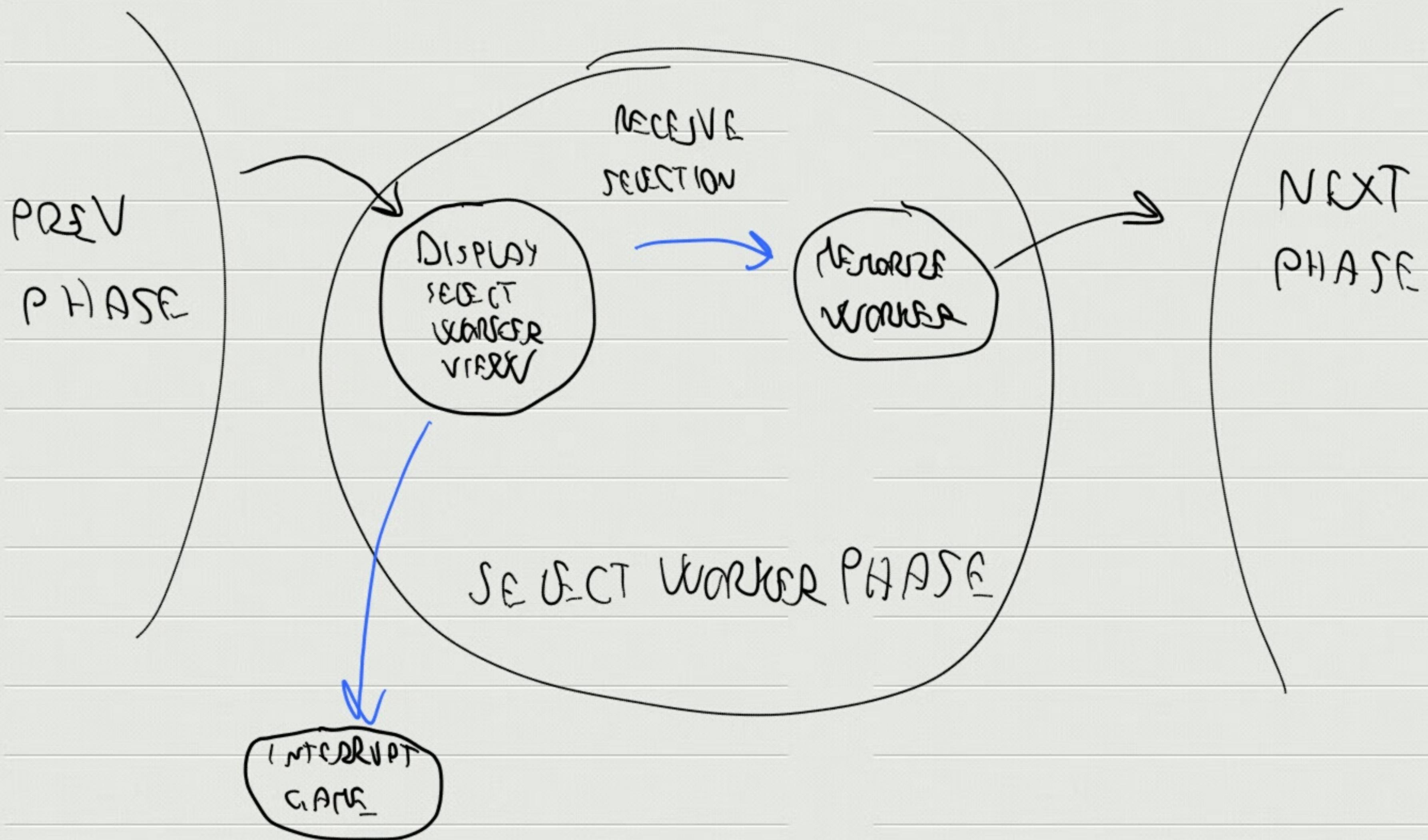


Note: PLAYER setup sopra account parametri che verranno usati nel turno.

Ad esempio seleziona il player corretto e la sua virtual view.

Inoltre recupera l'action tree per fornirlo poi alla view

• = USER INTERACTION \Rightarrow NOTIFICATION \Rightarrow UPDATE



NOTE: OGNI PHASE HA ALCUNO STATO DI ELABORAZIONE Dopo
LA RECEZIONE DI USER INPUT. PER SELECT, È SUFFICIENTE
MEMORIZZARE IL WORKER DA USARE NELL' ACTION.
DA OGNI VIEW È POSSIBILE QUITTARE (FA PERSISTENZA).
OGNI RECEZIONE USER INPUT È TRAMITE UPDATE \Rightarrow IL METODO DELLO STATO
"DISPLAY SELECT WORKER VIEW" TERMINA CON L'INVOCAZIONE DEL METODO
CHE MOSTRA LA VIEW. NON PUÒ PROSEGUIRE PERCHÉ POTREBBERE CONFLIGGERE
CON L'INVOCAZIONE DI UPDATE

ACTION PHASE

RECEIVING
ACTION

DISPLAY
ACTION
VIEW

UPDATE
VIEWS
...

UPDATE
MODEL

WIN

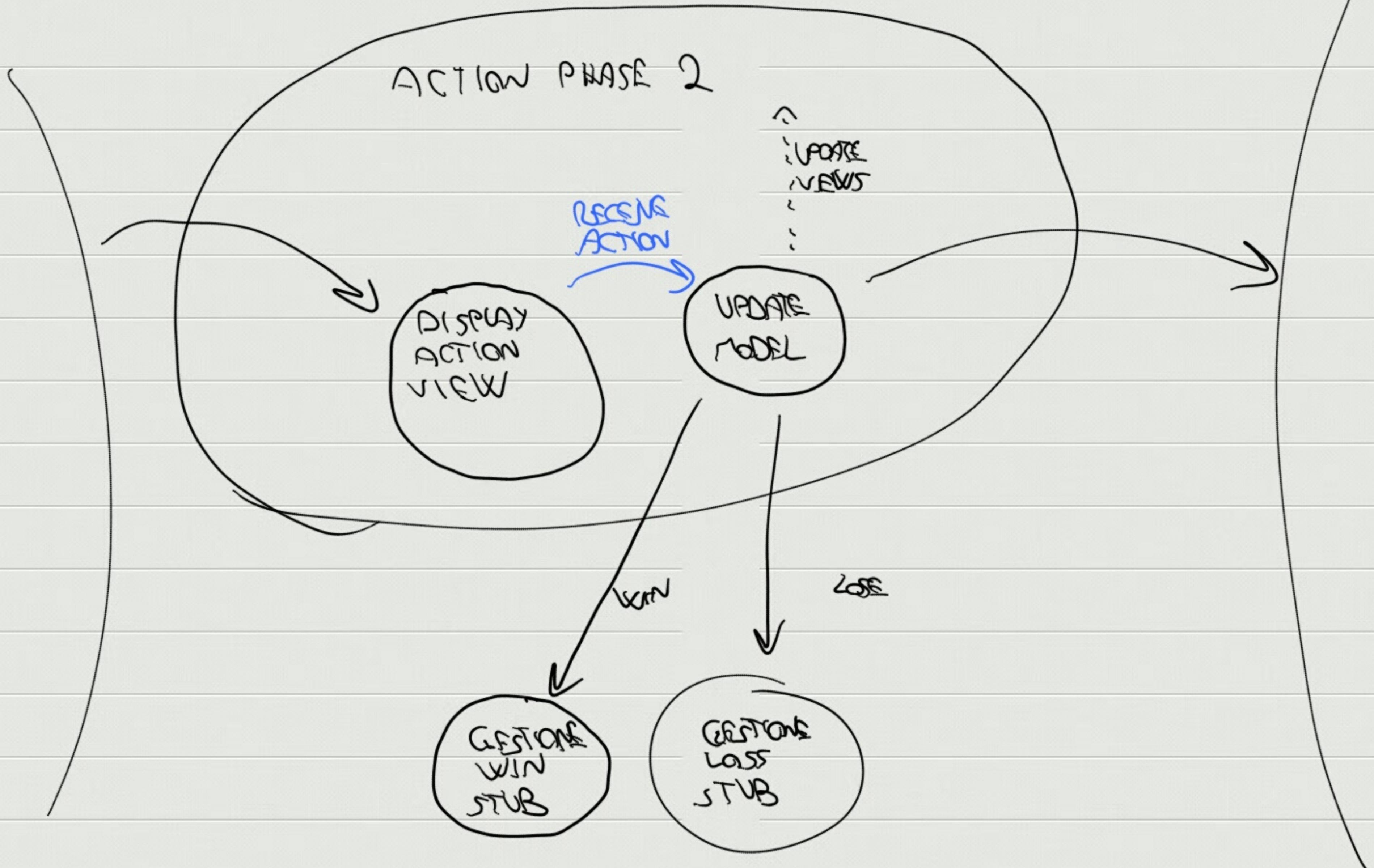
LOSE

GESTIONE
WIN
STUB

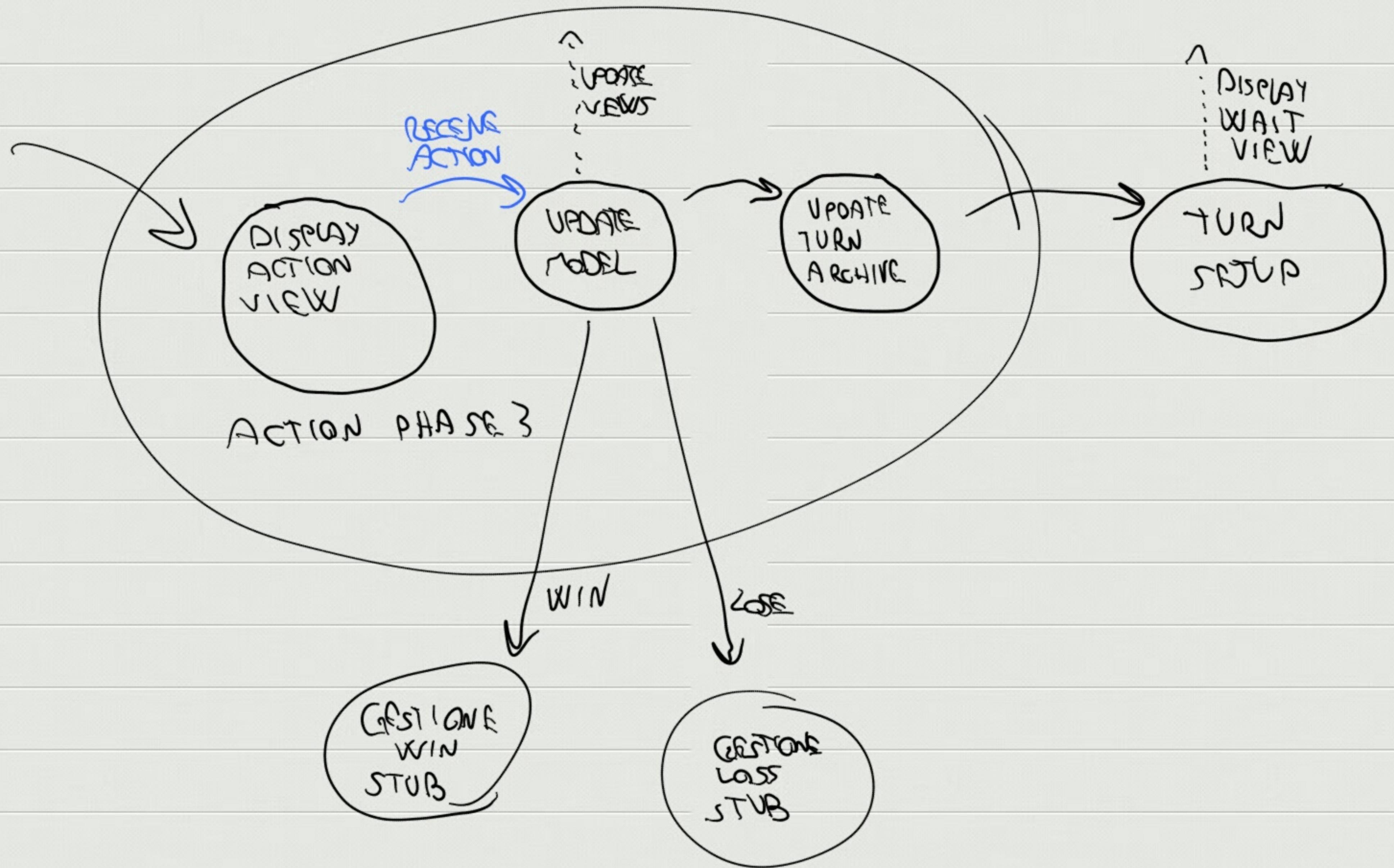
GESTIONE
LOSE
STUB

NOTE: "ACTION VIEW" USA L'ACTIONTYPE IN BASE ALLA ACTION
PRESENTI NEL PROTO LAYER PERMETTE ALL'UTENTE DI FARE MOVE, BUILD O
ENTRANCE. PER FARLO QUESTO DEVE SAPERE DI DOVER GUARDARE IL PRIMO
LAYER, QUINDI L'INVOCAZIONE DI ACTIONVIEW DEVE AVERE ANCHE UN
ARGOMENTO 1. IN ALTERNATIVA SI POTREBBE PASSARE ALL'ACTION
VIEW SOLTANTO IL LAYER DI ACTIONTYPE CHE GLI CONFERISCE
"UPDATE MODEL" PROPAGA UPDATE ALLA VIEW. INOLTRE CONTROLUA WINCONDITION
E LOSS CONDITION

ACTION PHASE 2



NOTE: QUASI IDENTICA ALLA FASE PRECEDENTE, MA CON VIE NUOVE
ARGOMENTO 2 INFINE CHE 1, IN MODO DA OSSERVARE SOLO
IL SECONDO VAYER DELL'ACTION TREE



NOTE: TURN STATE CHAIN WAITING WHILE SUL CURRENT PLAYER PRIMA DI
CAMBIARE LO STATO DJ TURNFLOW PER IL METODO SUCCESSIVO

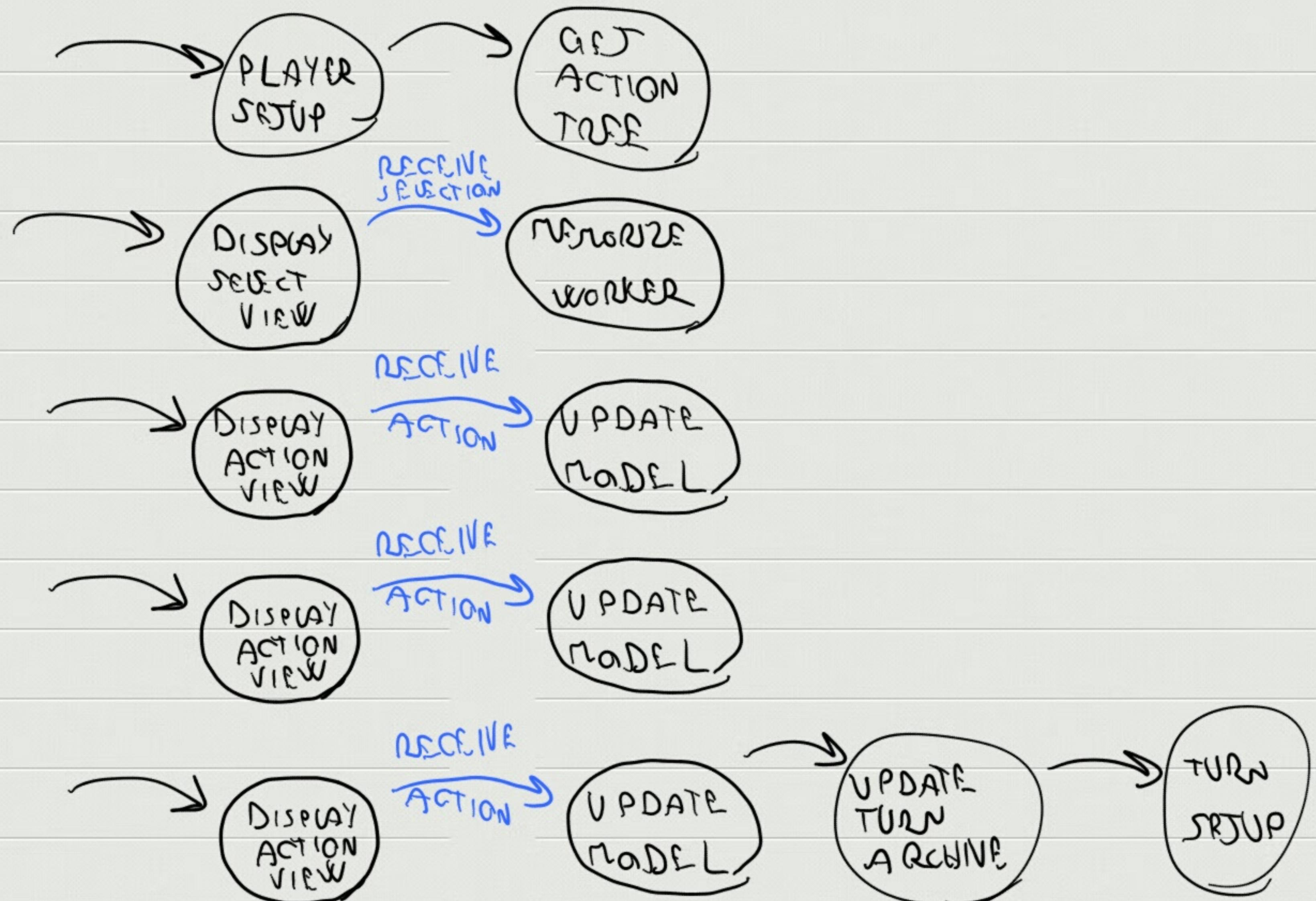
TURN
SETUP

SELECT
PHASE

ACTION
PHASE 1

ACTION
PHASE 2

ACTION
PHASE 3



TURN
SETUP

SELECT
PHASE

ACTION
PHASE 1

ACTION
PHASE 2

ACTION
PHASE 3

