Log All Player Actions

Problem: currently, we retain only 1 PlayerAction record per player/hand and update the 'active' status and bet/action values for the same row each time a player does something.

This results in a loss of hand history and means I can only get the latest action based on the updated_at column, which means sometimes a player will act within the same second as the previous player, and the code will not switch to the next player, meaning sometimes you get 2 actions for the price of one.

Requirement: update all logic related to player actions to create a new action rather than updating the existing one.

Then use the auto-increment primary DB key for the current hand/player-action to get the latest action.

Also update related unit tests to use a re-usable method for setting up hand scenarios based on player actions

Key

An existing step to be improved

A solution/improvement step for the problem

Undecided/TBC

Current Flow

PAC finds the

player's action and

updates it + calls

GamePlay->play()

updateSeatStatusOfLatestAction() and

updateAllOtherSeatsBasedOnLatestAction()

are called, when getting latest, it uses

updated_at

nextStep() is called, using lots of conditions to

check the status of each player - using

updated_at for the latest and they are also not

grouped by street ID, if that's needed at all?

All other steps

Request entry points

PlayerActionController /

HandController

Hand was completed

New street

HC creates a new

hand + calls

GamePlay->start()

initiateStreetActions()

is called, creating a

PlayerAction record for

each table seat.

setDealerAndBlindSeats() is called

and uses PlayerAction::find to see who

is current BB - not getting the latest

player action record

identifyTheNextDealerAndBlindSeats() is

called and uses table_seat_ids to determine

who the next dealer, sb & bb will be

When setting the new SB/BB on player actions, it does not currently look for the

latest action

getPlayerData() is called

getActionOn() is called

TableSeat::firstActivePlayer(\$this->handId)

is called, which triggers a DB query - need

to add group by player_id + order by PK

desc I think

allPlayerActionsAreNullSoANewSreetHasBeenSet() is called - this may continue to work, as when we create

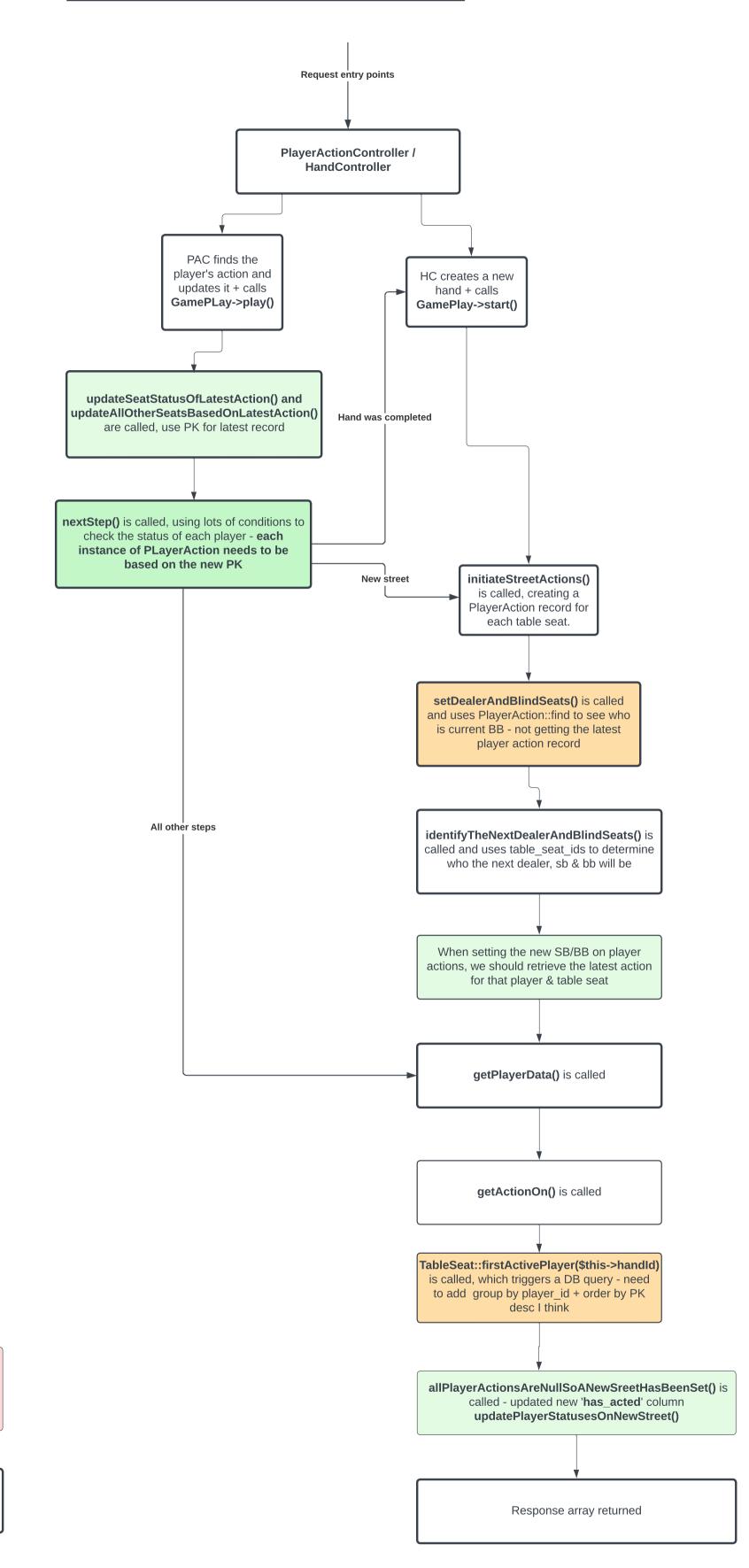
PlayerActions on a new street, they will all be null -

currently updating existing row in

updatePlayerStatusesOnNewStreet()

Response array returned

Proposed Flow



Notes/Conclusion

I've marked a couple boxes in orange. There are a lot of others like it where if we base it on a new row per action, we will also need to group by the latest set of actions for each player

Either I update each one to be based on the unique rows, OR keep the existing player_actions table and use it more like a 'player_status' table, so we can continue to keep track of who is active using the existing logic - it also doesn't make sense to have an 'active' column in a player_action_log, only active players can carry out an action.

We can then log a new record in **player_action_log** or similar and get the latest & hand histories that way.

Additionally, rather than changing all the individual calls to DB for this, they can be left and changed later to get the data from a **GameState** object, which I intend to populate with all or most of the data GamePlay needs to progress the game.

Proposed ERD of new log table with player_status modification to player_actions:

