GamePlay Class Extraction
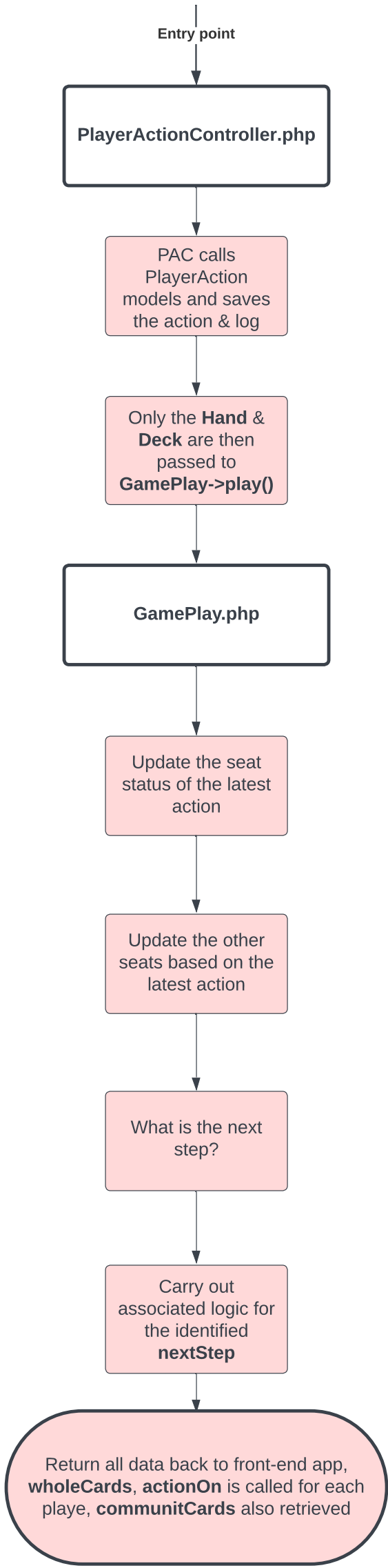
Problem: the GamePlay class is currently resposible for too much logic. It should be separated into respective classes, roughly 1 for handling actions, 1 for deciding the next step in the game, with a GameState object being passed around and storing all the information needed at each stage (reducing multiple DB queries to get the same data)
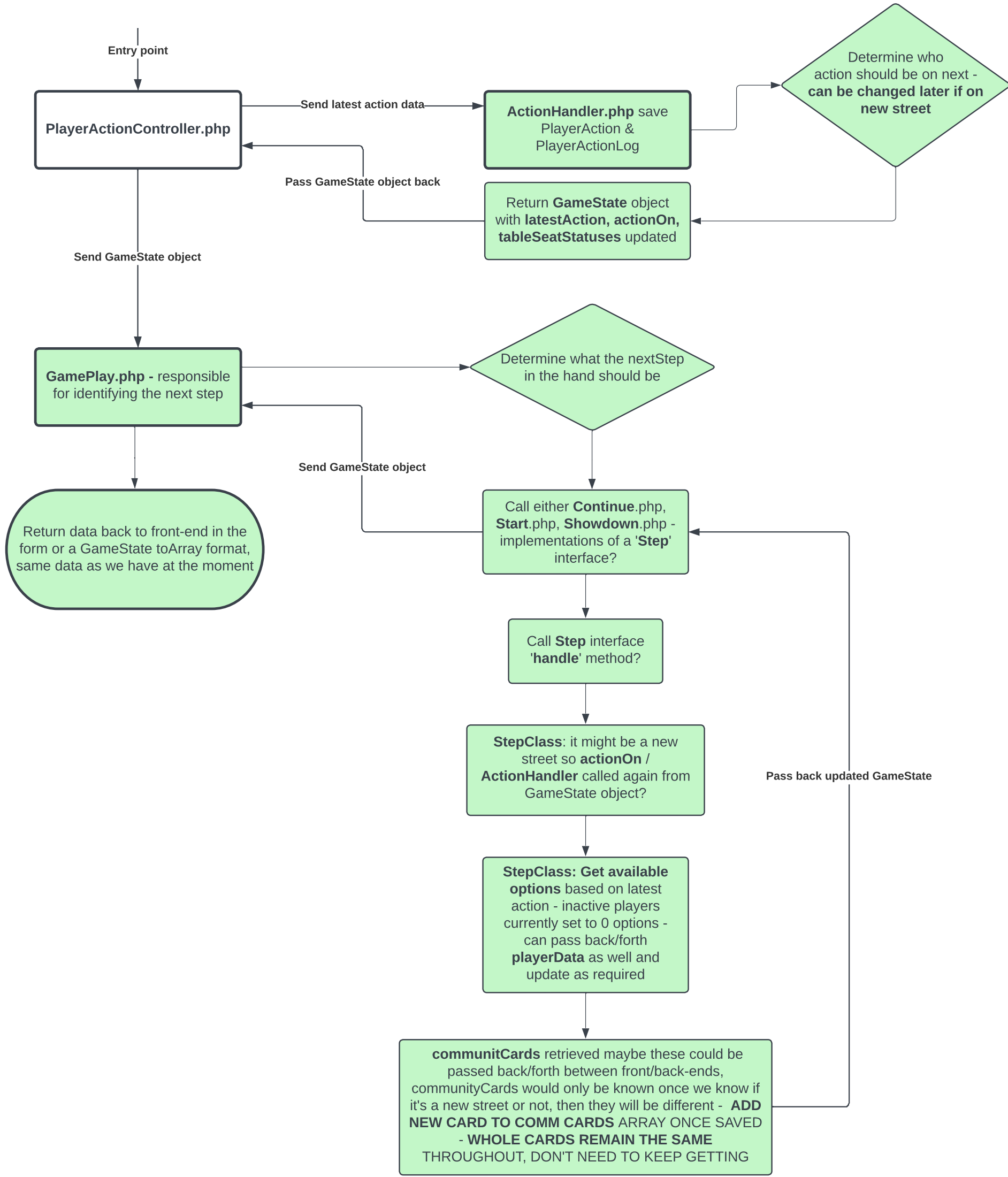
Key

- An existing step to be improved
- A solution/improvement step for the problem
- Undecided/TBC

Current Flow

Entry point

PlayerActionController.php

↓

PAC calls PlayerAction models and saves the action & log

↓

Only the **Hand** & **Deck** are then passed to **GamePlay->play()**

↓

GamePlay.php

↓

Update the seat status of the latest action

↓

Update the other seats based on the latest action

↓

What is the next step?

↓

Carry out associated logic for the identified **nextStep**

↓

Return all data back to front-end app, **wholeCards**, **actionOn** is called for each playe, **communitCards** also retrieved

Proposed Flow

Entry point

PlayerActionController.php —Send latest action data→ **ActionHandler.php** save PlayerAction & PlayerActionLog

→ Determine who action should be on next - **can be changed later if on new street**

Return **GameState** object with **latestAction, actionOn, tableSeatStatuses** updated

—Pass GameState object back→ PlayerActionController.php

↓ Send GameState object

GamePlay.php - responsible for identifying the next step

→ Determine what the nextStep in the hand should be

↓

Return data back to front-end in the form or a GameState toArray format, same data as we have at the moment

Call either **Continue**.php, **Start**.php, **Showdown**.php - implementations of a '**Step**' interface?

↑ Send GameState object

↓

Call **Step** interface '**handle**' method?

↓

StepClass: it might be a new street so **actionOn** / **ActionHandler** called again from GameState object?

↓

StepClass: Get available options based on latest action - inactive players currently set to 0 options - can pass back/forth **playerData** as well and update as required

↓

communitCards retrieved maybe these could be passed back/forth between front/back-ends, communityCards would only be known once we know if it's a new street or not, then they will be different - **ADD NEW CARD TO COMM CARDS** ARRAY ONCE SAVED **- WHOLE CARDS REMAIN THE SAME** THROUGHOUT, DON'T NEED TO KEEP GETTING

Pass back updated GameState →

Code Example

So **PlayerActionController** might look like this:

```
public function action()
{
    $requestBody = file_get_contents('php://input')
        ? json_decode(file_get_contents('php://input'), true)['body']
        : unserialize($_POST['body']);

    $gameState = $this->actionHandler($requestBody);

    return $this->gamePlay->handle($gameState);
}
```

GamePlay might look like this:

```
public function handle($gameState)
{
    return $this->nextStep($gameState);
}

private function nextStep()
{
    // logic to determine next step...

    return $this->start->handle($gameState),
        $this->continue->handle($gameState),
        $this->showdown->handle($gameState) accordingly
} // returning instance of StepInterface
```

Each **StepImplementation** might look like this:

```
public function handle($gameState)
{
    // specific step logic, updating GameState

    return $gameState;
} // so GameState can 'bubble' up the classes
```

Approach

Create GameState object
It should have properties for each current data set being given to front-end:

```
return [
    'deck'           => $this->dealer->getDeck(),
    'pot'            => $this->hand->pot()->amount,
    'communityCards' => $this->getCommunityCards(),
    'players'        => $this->getPlayerData(),
    'winner'         => $winner
];
```

With associated getter/setter methods that can be called
Other properties: latestAction, actionOn with getters/setters
Step classes don't need properties, only manipulating GameState?