**Determine First Player**

* Trigger:
  + Triggered by the round being updated in the GameStateManager.
* UI:
  + Both players see who the first player is. (via App.jsx -> GamePhaseModals.jsx -> FirstPlayerModal
  + Press ‘Continue’ to close.
    - This closes the popup and informs that the player is ready to proceed with the next turn.
    - When one player has acknowledged, but the other has not, the player who has acknowledged will be presented with the ‘Waiting for other Player’ modal (Need an action phase specific one of these)
* AI (if single player):
  + Automatically acknowledges.
* Processing:
  + SimultaneousActionProcessor updates that both players have completed their actions
  + GameFlowManager acknowledges that both players have completed their actions, and progresses to the next Phase.

**energyReset (automatic)**

* Trigger:
  + GameFlowManager moves to this phase.
* UI:
  + App.jsx -> GameHeader.jsx -> values are automatically updated.
  + No specific modal, game simply progresses.
* AI (if single player):
  + N/A - automatic
* Processing:
  + actionProcessor updates each players Energy state in GameStateManager and then informs SimultaneousActionProcessor that a player has completed their action.
  + SimultaneousActionProcessor updates the GameStateManager as to which player(s) have completed their action.
  + When both players have completed their actions, GameFlowManager progresses to the next Phase.

**madatoryDiscard (simultaneous)**

* Trigger:
  + GameFlowManager moves to this phase.
* UI:
  + If a player has no cards in hand, this is automatically passed by SimultaneousActionManager.
  + For any player who does need to discard:
    - They see the MandatoryActionModal
    - This takes them to the mandatory discard phase
    - They select cards and press confirm.
    - On confirm, the cards are discarded.
  + When one player has acknowledged, but the other has not, the player who has acknowledged will be presented with the ‘Waiting for other Player’ modal (Need an action phase specific one of these)
* AI (if Single Player):
  + AIPhaseProcessor passed to aiLogic to decides what cards to discard.
  + AIPhaseProcessor then informs actionProcessor of outcome.
* Processing:
  + actionProcessor updates the confirming players Hand / Discarded Card state in GameStateManager and then informs SimultaneousActionProcessor that a player has completed their action.
  + SimultaneousActionProcessor updates the GameStateManager as to which player(s) have completed their action.
  + When both players have completed their actions, GameFlowManager progresses to the next Phase.

**draw (automatic)**

* Trigger:
  + GameFlowManager moves to this phase.
* UI:
  + For both players, cards in hand are updated. Deck size is updated.
* AI (if single player):
  + N/A - automatic
* Processing -> Automatic:
  + actionProcessor automatically updates each players Hand and Deck states in GameStateManager and then informs SimultaneousActionProcessor that a player has completed their action.
  + SimultaneousActionProcessor updates the GameStateManager as to which player(s) have completed their action.
  + When both players have completed their actions, GameFlowManager progresses to the next Phase.

**AllocateShields (simultaneous)**

* Trigger:
  + GameFlowManager moves to this phase.
* UI:
  + If a player cannot allocate any shields (namely because all their shields are at max), this is automatically passed by SimultaneousActionManager.
  + For any player who does need to allocate shields:
    - App.jsx reacts and presents them with the RoundStartShieldAllocation menu
  + When one player has acknowledged, but the other has not, the player who has acknowledged will be presented with the ‘Waiting for other Player’ modal (Need an action phase specific one of these)
* AI:
  + AIPhaseProcessor passed to aiLogic to decides which shields to restore.
  + AIPhaseProcessor then informs actionProcessor of outcome.
* Processing -> On Confirm:
  + actionProcessor updates the new Shield states in GameStateManager and then informs SimultaneousActionProcessor that a player has completed their action.
  + SimultaneousActionProcessor updates the GameStateManager as to which player(s) have completed their action.
  + When both players have completed their actions, GameFlowManager progresses to the next Phase.

**madatoryDroneRemoval (simultaneous)**

* Trigger:
  + GameFlowManager moves to this phase.
* UI:
  + If a player is not over their deployment limit, this is automatically passed by SimultaneousActionManager.
  + For any player who does need to remove drones:
    - They see the MandatoryActionModal
    - This takes them to the mandatory drone removal process
    - They select drones to remove, and press confirm.
    - On confirm, the cards are removed.
  + When one player has acknowledged, but the other has not, the player who has acknowledged will be presented with the ‘Waiting for other Player’ modal (Need an action phase specific one of these)
* AI (if Single Player):
  + AIPhaseProcessor passed to aiLogic to decides what drones to remove.
  + AIPhaseProcessor then informs actionProcessor of outcome.
* Processing:
  + actionProcessor updates the confirming players removed Drones in GameStateManager and then informs SimultaneousActionProcessor that a player has completed their action.
  + SimultaneousActionProcessor updates the GameStateManager as to which player(s) have completed their action.
  + When both players have completed their actions, GameFlowManager progresses to the next Phase.