State Machine Design

Author: Dr. Wen Jiang

Objective

Define the states an aircraft could fall into on both incoming and outgoing stages. Design a state machine that updates the states and makes the transitions. Draw the state flow graph to help visualize the transitions.

State Machine

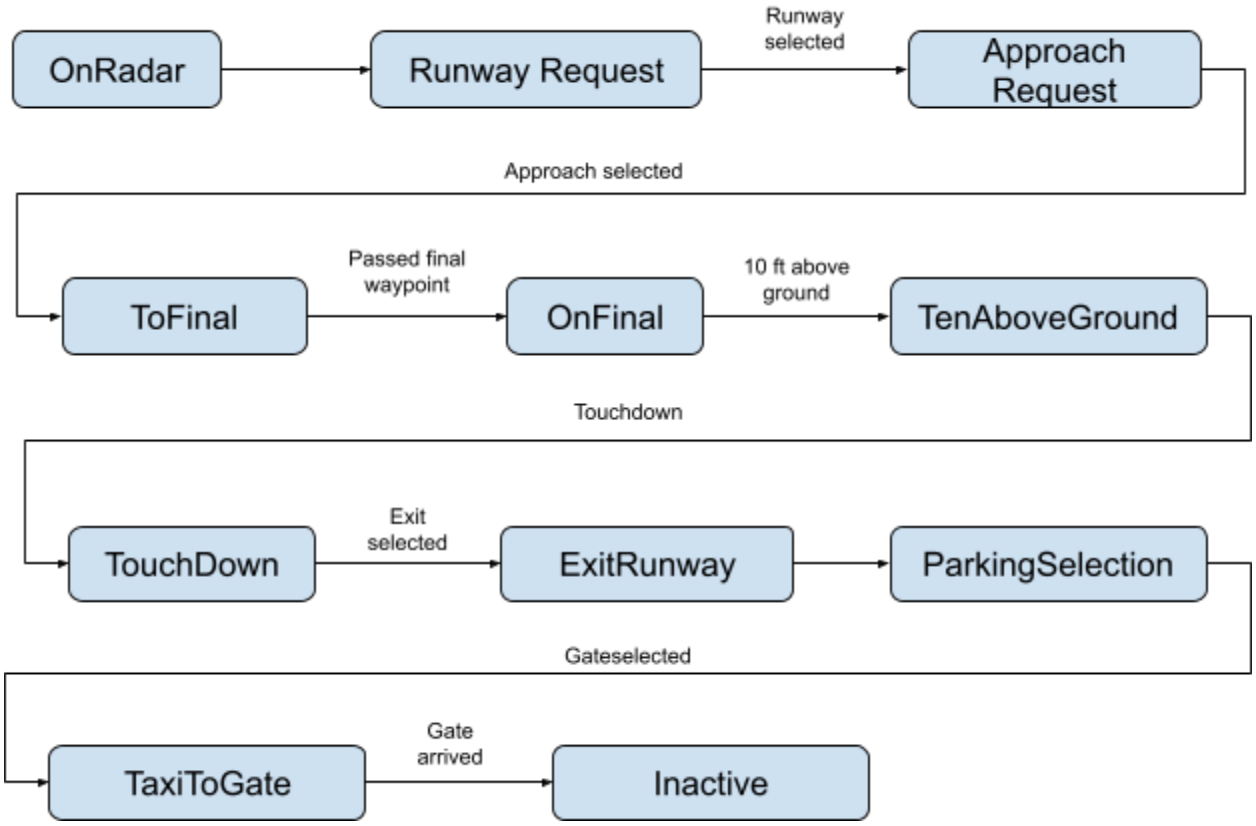
The aircraft object holds the StateMachine component. The StateMachine component holds a complete list of state instances. All the states inherit from the basic State class, which has entry, update and exit methods. The State class also has name(string) and id(int) members. The StateMachine class implements the transition method and passes it to all the states as delegates(callbacks).

Inbound States

1. OnRadar
   When the aircraft is within the radar detection range of the airport.
2. RunwayRequest
   Waiting for landing runway selection.
3. ApproachRequest
   Waiting for landing approach selection. The landing approach defines a series of waypoints guiding the aircraft to the final.
4. ToFinal
   Following waypoints to final.
5. OnFinal
   Passing the final waypoint and descending to the runway.
6. TenAboveGround
   10 ft above the runway and performing landing.
7. TouchDown
   The aircraft touches the runway. Triggering exit selection.
8. ExitRunway
   If the exit is selected, use that to exit the runway. If not selected, use the last exit to exit the runway.
9. HoldOnParkingSelection
   Waiting for parking gate selection.
10. TaxiToGate

Taxi to the selected gate.

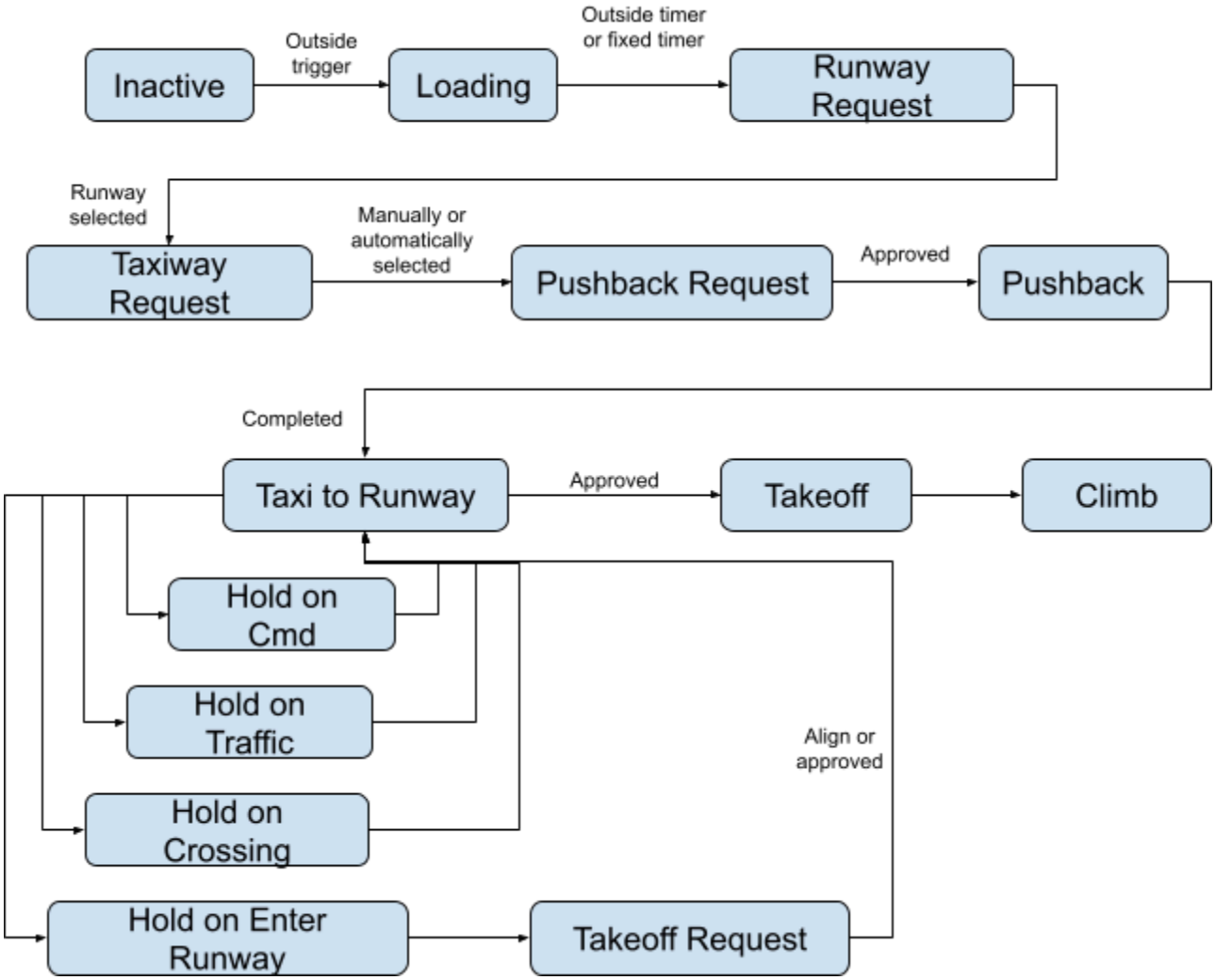
11. Inactive
    Once the aircraft reaches the gate, turn to the inactive state. This completes the inbound operation.



Outbound States

1. Inactive
   When receiving signal from outside, it can be transferred into loading.
   When exit, create a card in the outgoing panel.
2. Loading
   Loading has a timer which can be either from outside or a fixed number.
3. Runway request
   Call the airport object to get all possible runways for the user to choose. Activate the buttons for the runways.
4. Taxiway request
   Pop up manual or automatic buttons. If manual, the user needs to click waypoints, left click(+), right click(-).
5. Pushback request
   Pop up the pushback request button.
6. Pushback
   Start the engine and push back.

7. Taxi to runway
   Pop up the hold on command button.
8. Hold on command
   Pop up the continue button.
9. Hold on traffic
   Automatically continue when traffic is cleared.
10. Hold on sign
    Pop up different approval buttons based on the sign.



Future Work

1. Support go around
2. Support state transition fully automated rather than manually triggered