States:

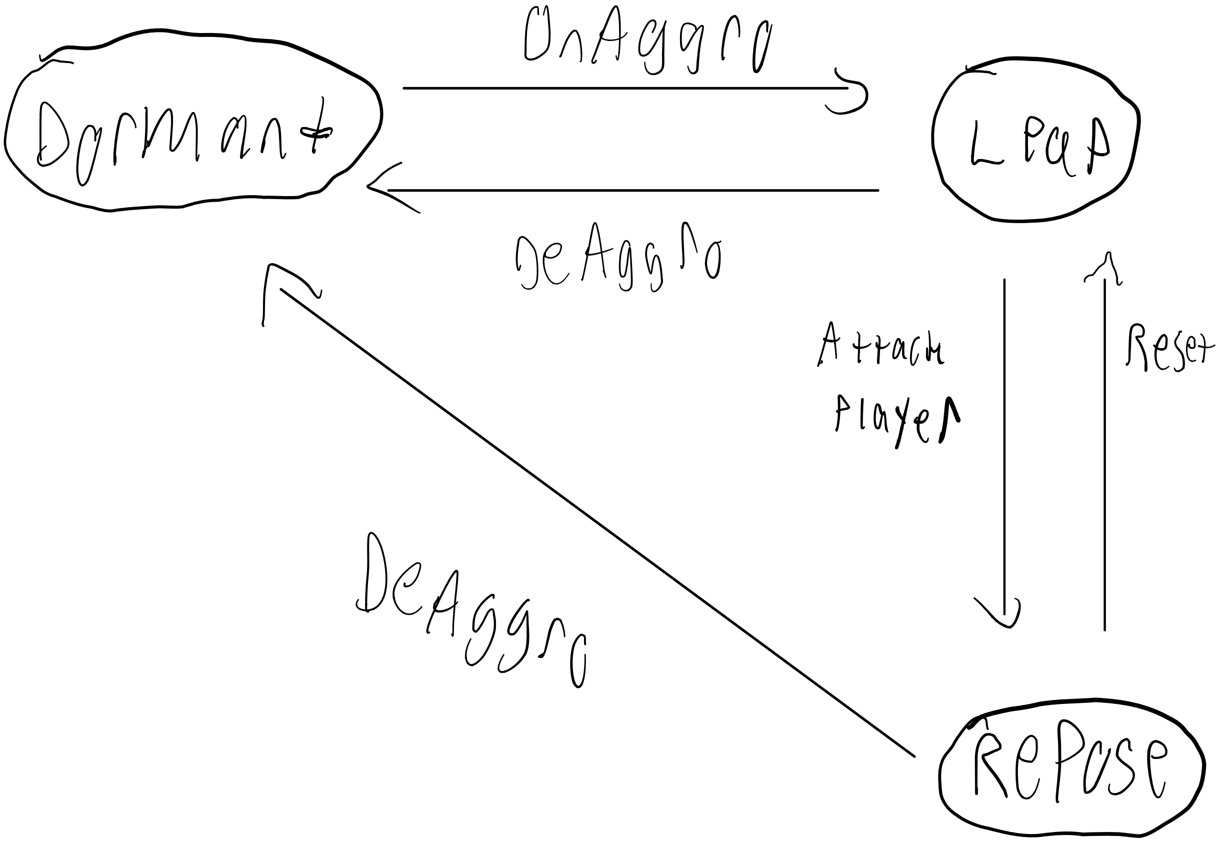
leap attack:

attack player in a straight line
linear/constant velocity

Reposition:

- leap to location on the ceiling

detect player?


OnAggro
Dormant
Leap
deAggro
Attack
Player
Reset
DeAggro
Repose

Conditions:

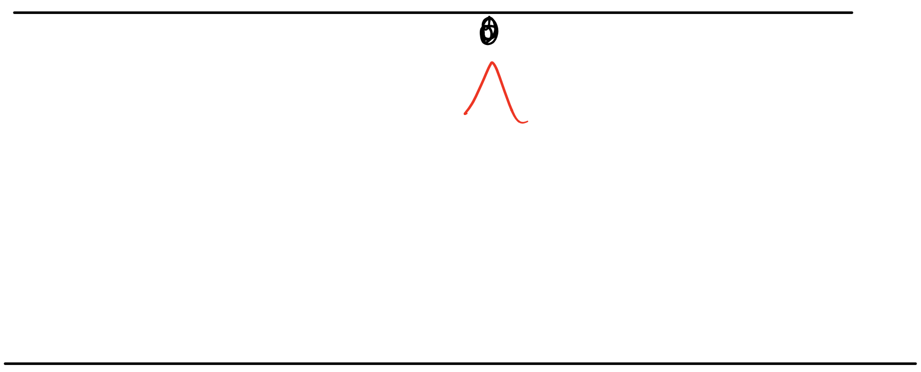
lock on position:

get the player's last known position: indicated by position when player caught agro.

- warp to this position. quickly.

Repose:

- Find the next available

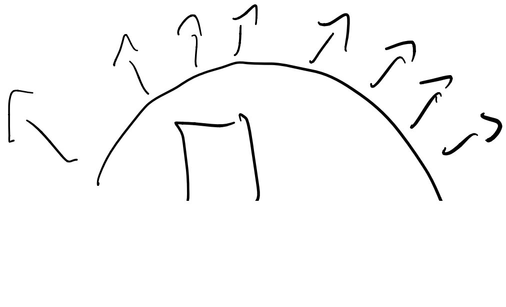


```
On Aggro:
    - Attack Player (
        dormat -> leap -> repose
          - use reference in "flying character"
                                to move it
On DeAggro:
    - checkState
      - go back to dormant

// call in Update
    AttackPlayer():
      State = leap;

        if( leap ) {
          get current_player position;
          get nearest adjacent surface.

          lerp to that position w/ respect to
            player and surface position
              - linear/const velocity
          State = repose;
        } else if(repose) {
          - find nearest ceiling/wall
```



```
                —            ,

          ~leap to that wall.

            switch::leap();
            add cooldown;
          }
    }

    //deAggro
    Dormant(){
        state::dormant

        active = false;

    }
```