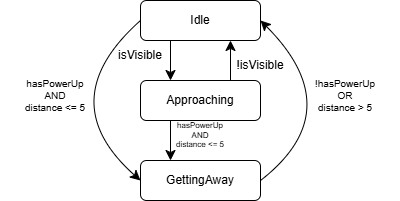
| (From)  Current State | (Under) Condition | To State |
| --- | --- | --- |
| Idle  Approaching  Idle  GettingAway  Approaching | isVisible  !isVisible  hasPowerUp AND distance <= 5  !hasPowerUp OR distance > 5  hasPowerUp AND distance <= 5 | Approaching  Idle  GettingAway  Idle  GettingAway |



Given a problem of size n and two algorithms with costs f(n)>0 and g(n)>0 each, we say that f(n) is in O(g(n)) when

there exist constants C>0 and n0≥0, so f(n) ≤ C𑛀g(n) for all n≥n0

f(n) = 1000000\*n, g(n) = n, as this growth depends on the grow of n

f(n)=2000+n⋅sin(n), g(n) = n, as sin(n) must be between -1 and 1, and the constant doesn’t matter as n grows

f(n)=5+n⋅(n−1)/2+log(n)⋅n

f(n)=5+n2/2-n/2+log(n)⋅n

g(n) = n2, as n grows, n2 will be the dominant factor