* Attack State
  + BuildingBase = Mathf.Clamp(1 - myBases.Count, 0, 1) \* Mathf.Clamp(Gold - (Constants.COST[UnitType.BASE] - 1), 0, 1);
    - **I wanted to keep the base count at one**
  + BuildingBarracks = Mathf.Clamp((myBarracks.Count - enemyBarracks.Count) - 3, 0, 1) \* Mathf.Clamp(Gold - (Constants.COST[UnitType.BARRACKS] - 1), 0, 1);
    - **I wanted to keep the barracks count at four**
  + BuildingRefineries = Mathf.Clamp(1 - myRefineries.Count, 0, 1) \* Mathf.Clamp(Gold - (Constants.COST[UnitType.REFINERY] - 1), 0, 1);
    - **I wanted to keep the refineries count at one**
  + BuildingArchers = Mathf.Clamp((enemyArchers.Count - myArchers.Count) - 4, 0, 1) \* Mathf.Clamp(Gold - (Constants.COST[UnitType.ARCHER] - 1), 0, 1);
    - **I wanted my archer count to be 4 more than that of the enemies archer count**
  + BuildingSoldiers = Mathf.Clamp((enemySoldiers.Count - mySoldiers.Count) - 4, 0, 1) \* Mathf.Clamp(Gold - (Constants.COST[UnitType.SOLDIER] - 1), 0, 1);
    - **I wanted my soldier count to be 4 more than that of the enemies soldier count**
  + BuildingWorkers = Mathf.Clamp(myWorkers.Count - 4, 0, 1) \* Mathf.Clamp(Gold - (Constants.COST[UnitType.WORKER] - 1), 0, 1);
    - **I wanted to keep my worker count at 4**
  + GatherGold = Mathf.Clamp(150 - Gold, 0, 1);
    - **I wanted to collect gold if my gold is less than 150**
  + AttackWithArchers = Mathf.Clamp((enemyWorkers.Count + enemyArchers.Count + enemySoldiers.Count) - 4, 0, 1);
    - **I wanted to attack if the enemies exceed four plus their counts for workers, archers, and soldiers**
  + AttackWithSoldiers = Mathf.Clamp((enemySoldiers.Count + enemyArchers.Count + enemyWorkers.Count) - 4, 0, 1);
    - **I wanted to attack if the enemies exceed four plus their counts for workers, archers, and soldiers**
  + AttackEverything = Mathf.Clamp(enemyAgentNbr - (enemyArchers.Count + enemyWorkers.Count + enemySoldiers.Count), 0, 1) \* Mathf.Clamp(enemyArchers.Count + enemySoldiers.Count + enemyWorkers.Count, 0, 1);
    - **I wanted to attack everything based off the total number of agents minus workers, archers, and soldiers count times workers plus archers plus soldiers**
* Build State
  + BuildingBase = Mathf.Clamp(1 - myBases.Count, 0, 1) \* Mathf.Clamp(Gold - (Constants.COST[UnitType.BASE] - 1), 0, 1);
    - **I wanted to keep the base count at one**
  + BuildingBarracks = Mathf.Clamp((enemyBarracks.Count - myBarracks.Count) + 2, 0, 1) \* Mathf.Clamp(Gold - (Constants.COST[UnitType.BARRACKS] - 1), 0, 1);
    - **I wanted to keep my barracks count 2 more than that of the enemies barracks count**
  + BuildingRefineries = Mathf.Clamp(3 - myRefineries.Count, 0, 1) \* Mathf.Clamp(Gold - (Constants.COST[UnitType.REFINERY] - 1), 0, 1);
    - **I wanted my refineries to be at a count of 3**
  + BuildingArchers = Mathf.Clamp(enemyArchers.Count \* 2, 0, 1) \* Mathf.Clamp(Gold - (Constants.COST[UnitType.ARCHER] - 1), 0, 1);
    - **I wanted my archers count to be two times the enemies**
  + BuildingSoldiers = Mathf.Clamp(mySoldiers.Count \* 1, 0, 1) \* Mathf.Clamp(Gold - (Constants.COST[UnitType.SOLDIER] - 1), 0, 1);
    - **I wanted my soldiers count to be the same as the enemies**
  + BuildingWorkers = Mathf.Clamp((enemyBases.Count - myWorkers.Count) + 5, 0, 1) \* Mathf.Clamp(Gold - (Constants.COST[UnitType.WORKER] - 1), 0, 1);
    - **I wanted my workers count to be six**
  + GatherGold = Mathf.Clamp(1500 - Gold, 0, 1);
    - **I wanted my gold to be no lower than 1500**
  + AttackWithArchers = Mathf.Clamp(myArchers.Count - 10, 0, 1);
    - **I wanted my archers to be at a count of ten**
  + AttackWithSoldiers = Mathf.Clamp(mySoldiers.Count - 10, 0, 1);
    - **I wanted my soldiers to be at a count of ten**
  + AttackEverything = Mathf.Clamp(enemyAgentNbr - (enemyArchers.Count + enemyWorkers.Count + enemySoldiers.Count), 0, 1) \* Mathf.Clamp(enemyArchers.Count + enemySoldiers.Count + enemyWorkers.Count, 0, 1);
    - **I wanted to attack everything based off the total number of agents minus workers, archers, and soldiers count times workers plus archers plus soldiers**
* Win
  + BuildingBase = Mathf.Clamp(0, 0, 1);
  + BuildingBarracks = Mathf.Clamp(0, 0, 1);
  + BuildingRefineries = Mathf.Clamp(0, 0, 1);
  + BuildingArchers = Mathf.Clamp(0, 0, 1);
  + BuildingSoldiers = Mathf.Clamp(0, 0, 1);
  + BuildingWorkers = Mathf.Clamp(0, 0, 1);
  + GatherGold = Mathf.Clamp(0, 0, 1);
  + AttackWithArchers = Mathf.Clamp(0, 0, 1);
  + AttackWithSoldiers = Mathf.Clamp(0, 0, 1);
  + AttackEverything = Mathf.Clamp(1, 0, 1);
  + **This state just cleans up anything and everything**
  + **The only action that can be taken is AttackEverything**