# Game Update Summary

## 1. General Changes

• The entire system flow and algorithms were reworked from the ground up, providing a much smoother experience.  
• In the previous version, unlimited day/night reinforcement caused the game to run indefinitely.  
• Now, each game always ends with a winner; even when predator/prey roles are assigned randomly, the game concludes.

## 2. Reinforcement Mechanism

• Separate energy reinforcement boosts have been defined for Orcs (predators) and Dwarves (prey).  
• Reinforcement decreases dynamically: after every 500 turns, the boost amount gradually reduces.  
• The effect of reinforcement energy increases and agent reproduction is capped; after a certain number of turns or duration, its impact fades.

## 3. Weather and Day/Night Cycle Effects

• Random weather states (“clear”, “rain”, “storm”) have been added to the map, changing at set intervals.  
• Rain and storms increase energy drain and slow down movement speed.  
• The day/night cycle adjusts agents’ energy loss rates and manages predator/prey role swaps.

## 4. Agent Features

• Each agent now has speed and vision\_radius attributes, initialized with random values.  
• Movement behavior is guided by these attributes, affecting pursuit or evasion tactics.  
• Agents age over time and have a maximum lifespan; older agents die naturally.

## 5. Resources and Obstacles

• Random obstacles and resource nodes have been placed on the map.  
• Agents can gain energy from resource nodes; depleted nodes regenerate after a cooldown period.

## 6. Shortcut Keys

• P – Pause/Resume the game  
• H – Toggle heatmap view  
• M – Toggle music on/off  
• R – Trigger instant reinforcement

## 7. Other Improvements

• An event log now displays recent actions on screen.  
• High score tracking and a minimap feature have been implemented.  
• Visual enhancements: health bars, animation steps, and particle effects.

## 8. Q-Learning loop:

* **State representation**: At each time step, an agent observes a discrete state tuple
* 
* **Action selection**: Using an **ε-greedy** policy, the agent either explores a random move or exploits its current Q-table to pick the action with the highest estimated value.
* **Reward signal**: After moving (or waiting), the agent’s reward is computed as the change in its energy (gained by eating food, lost by movement).
* **Q-update**: The agent updates its Q-table entry for (s,a)(s,a)(s,a) according to



where **α** is the learning rate and **γ** the discount factor.

Over the course of a single run, agents gradually bias toward actions that historically yielded higher energy gains,

i.e., “learning” to seek food and avoid stronger enemies.