Feature: Carrier Conveyor Belt

Author: Chris Dillinger

Step 1:

**Investigate and Observe Current State to ensure it is fully understood.**

To be sure this feature increases carrier resource gathering effectiveness, I need to benchmark the effectiveness of current state behavior. I will look at the case of an Ad well which should be also apply to Mn and Elixir wells. Afterwards I will analyze anchor delivery effectiveness. For sake of simplicity, I will disregard pathfinding optimizations.

Diagram

Description automatically generated

Fig 1.

Diagram

Description automatically generated

Fig 2.

Fig 1 and Fig 2 demonstrate our current state Carrier behavior.

Fig 1 shows the HQ producing 3 Carriers. Each Carrier runs this strategy: sense initial well, travel to well, fill up to max capacity, travel to HQ, drop off resources, and repeat. If a carrier senses an obstacle in between their current location and the intended path towards the direction of their target location, they move one location to the right. This is repeated until there is no longer an obstacle in-between the carrier’s current location and the intended path towards the direction of the carrier’s target location.

Fig 2 shows a Carrier move 5 times before dropping resources off at HQ.

**What to measure?**

* Number of turns used to deliver 40 kg Ad from Well to HQ.

**What noise may decrease the accuracy of my benchmarking?**

I noticed the HQ builds carriers in a random direction. To simplify this a little, I merged a PR to update HQ to prefer building in the direction of the first sensed well. Link to PR: <https://github.com/team-remember-to-hydrate/battlecode23-team-remember-to-hydrate/pull/50>. After some consideration, this won’t matter much since I can focus on the carrier traveling from Well to HQ. Our largest gain will probably be from eliminating the need for a full carrier to travel from Well back to HQ. For example, Fig 3 below demonstrates how a carrier could have transferred resources through another carrier instead of moving.

Diagram

Description automatically generated

Fig 3.

**Current State Measurements**

Carrier C (Fig 2) took 10 turns to move 40 kg of resources to HQ.

Explanation:

1. C moved 5 locations as shown in Fig 2.
2. C’s movement cooldown is 20 turns since movement cooldown is FLOOR(5 + 3m/8) where m is kg of resources and 5 + 120/8 = 5 + 15 = 20.
3. C took 10 turns to move 5 times since carriers can move one location when their movement cooldown is less than 10 and after every turn, action cooldowns are decremented by 10 and 10 must be decremented from 20 twice before we arrive at a value less than 10.

**Current Implementation**

This is the part of the existing CarrierStrategy.java logic that runs when a Carrier is moving from a well back to an HQ to drop off all its resources, so this is where the returning carrier may decide to transfer resources instead of traveling around a group.

else if(*amountResourcesHeld* == GameConstants.*CARRIER\_CAPACITY*){  
 Pathing.*moveWithBugNav*(rc, *hqLoc*); // <--   
 *tryDropAllResources*(rc, *hqLoc*);  
}

Step 2:

**Identify Future State**

Carrier D (Fig 3) should take X turns to move 40 kg of resources to HQ.

Explanation (happy path):

1. D retrieves max capacity of Ad (40 kg) from the Well
2. D checks if it can move towards HQ and is blocked by another (empty) carrier E
3. D checks if it can transfer resources to E, and does, leaving D empty and E with 40 kg Ad
4. Empty carriers travel to well, and full carriers travel to HQ

Step 3:

**Implement**