Jakub Skalski

Code Keeper

16th December 2021

Introduction

In this report we will focus on creating a dedicated agent for the *Code Keeper* game on the CodingGames platform. The solution will be based on a if-then-else structure with no simulations. Agent is implemented in Python and uses BFS as its pathfinder modified to skip wasteful health potions. The general pseudocode for the agent is as follows:

```
If there are enemies nearby
      Perform combat
If HeroHealth <= 15 and potion nearby
      Obtain the potion
If HeroHealth >= 10 and EnoughTimeLeft
      Perform exploitation
If ExitReachable
      Run to the exit
If Else
      Perform exploration
```

Combat

Attempt to shoot the closest orc from a reasonable distance if none are present compare the value of the best use of scythe against the hammer as follows:

```
If Hammer hits 3 targets
      Use Hammer
If Scythe hits target 2 tiles away
      Use Scythe
If Hammer hits 2 targets
      Use Hammer
If Else
      Use Sword
```

Exploitation

Choose one of the discovered bounties using the following heuristic, which calculates the value of each bounty based on the current disposition of resources it restores and the its distance from the hero:

```
def loot(player, target, type, dist):
   dist = dist[target[0]][target[1]]

   if type == GOLD:
       return 100 / dist
   elif type == HPOT:
       return (20 - player.health)**2 / dist
   elif type == CHAM:
       return 10 * (10 - player.cham) / dist
   elif type == CSCY:
       return 10 * (10 - player.cscy) / dist
   elif type == CBOW:
       return 15 * (10 - player.cbow) / dist
   raise Exception(f"bad target {type}")
```

If none are obtainable, perform exploration instead.

Exploration

Simply move to the closest undiscovered tile, if none exist earn points through breaking boxes then exit.

So.. how good is it?

Agent passes all validator tests but struggles with seed 4 and 6. As far as the multiplayer test goes it scores 7th place as of the time of taking this screenshot: