Challenge: Battling Knights

The Rules

Knights and Arena

There are four knights who are about to do battle.

```
RED (R)
BLUE (B)
GREEN (G)
YELLOW (Y)
```
Their world consists of an 8x8 square "Arena" which looks suspiciously like a chess-board. The Arena is surrounded by water on all sides.

The 64 tiles on the board are identified with (row, col) co-ordinates with (0,0) being the top left tile and (7,0) being the bottom left tile (row 7 col 0).

Each knight starts in one corner of the board.

```
R (0,0) (top left)
B (7,0) (bottom left)
G (7,7) (bottom right)
Y (0,7) (top right)
```

Items

Around the board are the following four items.

```
Axe (A): +2 Attack
Dagger (D): +1 Attack
Helmet (H): +1 Defence
MagicStaff (M): +1 Attack, +1 Defence
```
They start at the following locations:

```
Axe (A) (2,2)
Dagger (D) (2,5)
MagicStaff (M) (5,2)
Helmet (H) (5,5)
```
If a Knight moves onto a tile with an item they are immediately equipped with that item, gaining the bonus. A Knight may only hold one item. If a knight with an item moves over another item then they ignore it. If a knight moves onto a tile which has two items on it then they pick up the best item in this order: (A, M, D, H). Knights will pick up an item on a tile before fighting any enemies on that tile. Knights that die in battle drop their item (if they have one). Knights that drown throw their item to the bank before sinking down to Davy Jones' Locker - the item is left on the last valid tile that the knight was on.

Movement

Each Knight moves one tile at a time in one of four directions.

```
North (N) (UP)
East (E) (RIGHT)
```

```
South (S) (DOWN)
West (W) (LEFT)
```
If a knight moves off the board then they are swept away and drown immediately. Further moves do not apply to DROWNED knights. The final position of a DROWNED knight is null.

Fighting

Each Knight has a base attack and defence score of 1:

```
Attack (1)
Defence (1)
```
If one knight moves onto the tile of another knight then they will attack. The knight already on the tile will defend.

The outcome of a fight is determined as follows:

- The attacker takes their base attack score and adds any item modifiers.
- The attacker adds 0.5 to their attack score (for the element of surprise).
- The defender takes their base defence score and adds any item modifiers.
- The attackers final attack score is compared to the defenders final defence score.
- The higher score wins, the losing knight dies.

DEAD knights drop any equipped items immediately. Further moves do not apply to DEAD knights. The final position of a DEAD knight is the tile that they die on.

A DEAD or DROWNED knight has attack 0 and defence 0.

Game

The initial state of the board looks like this:

```
(0,0) _ _ _ _ _ _ _ _   (0,7)
      |R|_|_|_|_|_|_|Y|
      |_|_|_|_|_|_|_|_|
      |_|_|A|_|_|D|_|_|
      |_|_|_|_|_|_|_|_|
      |_|_|_|_|_|_|_|_|
      |_|_|M|_|_|H|_|_|
      |_|_|_|_|_|_|_|_|
      |B|_|_|_|_|_|_|G|
(7,0)                   (7,7)
```
You are supplied a list of movements in the following format:

```
GAME-START
<Knight>:<Direction>
<Knight>:<Direction>
<Knight>:<Direction>
.
.
.
GAME-END
```
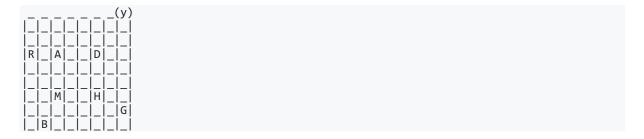For example:

```
GAME-START
R:S
R:S
```

```
B:E
G:N
Y:N
GAME-END
```
After these four movements the board would look like this (lower-case denotes a dead or drowned knight):

```
_ _ _ _ _ _ _(y)
|_|_|_|_|_|_|_|_|
|_|_|_|_|_|_|_|_|
|R|_|A|_|_|D|_|_|
|_|_|_|_|_|_|_|_|
|_|_|_|_|_|_|_|_|
|_|_|M|_|_|H|_|_|
|_|_|_|_|_|_|_|G|
|_|B|_|_|_|_|_|_|
```

Instructions

Your code should:

Open a file called `moves.txt` and, if the contents are a valid set of moves, determine the final state of the board. The output should be a json file called `final_state.json` with the following information:

- Position of the knights
- Status of the knights (LIVE, DEAD, DROWNED)
- Attack Power of each knight (including weapons but not surprise bonus)
- Defence Power of each knight (inluding weapons)
- Position of the items (and whether they are held by a knight or not)

In the following format:

```
{
    "red": [<R position>,<R status>,<R item (null if no item)>,R Attack,<R Defence>],
    "blue": [<B position>,<B status>,<B item (null if no item)>,B Attack,<B Defence>],
    "green": [<G position>,<G status>,<G item (null if no item)>,G Attack,<G Defence>],
    "yellow": [<Y position>,<Y status>,<Y item (null if no item)>,Y Attack,<Y Defence>],
    "magic_staff": [<M position>,<M equipped>],
    "helmet": [<H position>,<H equipped>],
    "dagger": [<D position>,<D equipped>],
    "axe": [<A position>,<A equipped>],
}
```
The output for the final board in the example above should therefore be:

```
{
    "red": [[2,0],"LIVE",null,1,1],
    "blue": [[7,1],"LIVE",null,1,1],
    "green": [[6,7],"LIVE",null,1,1],
    "yellow": [null,"DROWNED",null,0,0],
    "magic_staff": [[5,2],false],
    "helmet": [[5,5],false],
    "dagger": [[2,5],false],
    "axe": [[2,2],false]
}
```
Please include a README with your solution detailing how to use the code you write to determine the output of an instruction file.