Little Battle

This is an individual assignment due on **30th May 23:59 (at 23.59 Sydney local time)**, which worth 20% of your final grade. You should complete file_loading_tests.py (with the invalid files you create) and little_battle.py, and then submit them.

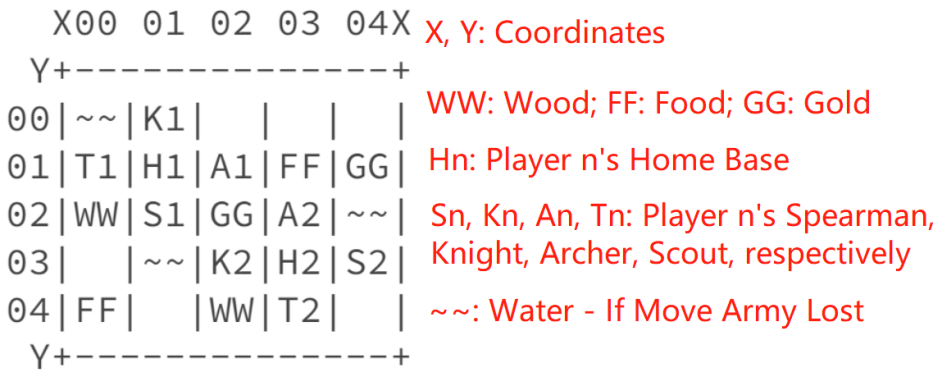
Implement a turn-based strategy (TBS) game that has the following features:

- The game is initialized with a map with resources and two players (Player 1 and Player 2).
- Each player has a home base. Player 1 and Player 2's home bases should always be in positions (1, 1) and (map_width-2, map_height-2), respectively. (coordinates start from (0, 0))
- The winning condition is to capture the other player's home base by an army.
- There are two stages including "Recruit Armies" and "Move Armies" in a player's turn.
- In stage "Recruit Armies", each player can recruit armies and only place the newly recruited armies next to their home base (4 positions surrounding the home base).
- Recruiting armies costs resources.
- There are three types of resources on the map: wood (W), food(F), gold(G)
- Each player initially has 2W, 2F and 2G.
- There are four types of soldiers: Spearman (S) costs 1W, 1F; Archer (A) costs 1W, 1G; Knight (K) costs 1F, 1G, Scout (T) costs 1W, 1F, 1G
- Spearman (S) counters Knight (K); Knight (K) counters Archer (A); Archer (A) counters Spearman (S); All other types of armies counter Scout (T); The table below shows the outcome of a fight:

	Spearman (S)	Knight (K)	Archer (A)	Scout (T)
Spearman (S)	Both disappear	K disappear	S disappear	T disappear
Knight (K)	K disappear	Both disappear	A disappear	T disappear
Archer (A)	S disappear	A disappear	Both disappear	T disappear
Scout (T)	T disappear	T disappear	T disappear	Both disappear

- In stage "Move Armies", each Spearman (S), Archer (A), and Knight (K) can move one step in each turn while Scout (T) can move up to two steps (one step or two steps in the same direction) but move only once.
- Each player can command an army to collect resources

The following graph shows an example of a 5x5 game map (from 5x5 to 7x7 with the same pattern):



Note: everything is case-sensitive in this assignment.

Read Configuration File

The program needs to read game configuration from a file that specifies the map size, water, resources via `python3 little_battle.py <filepath>`

The program should terminate once it encounters the first error. The error checking should be in the following order:

Load File

If the user does not provide a single argument in the command line, print `Usage: python3 little_battle.py <filepath>`

If the provided file is not found, raise a `FileNotFoundError`

File Contents

Here is an 5x5 example of the configuration file (You can find it on Ed):

```
Frame: 5x5 # widthxheight
Water: 0 0 4 2 1 3 # positions (0, 0), (4, 2), (1, 3) should be water
Wood: 0 2 2 4 # positions (0, 2), (2, 4) should be wood
Food: 0 4 3 1 # positions (0, 4), (3, 1) should be food
Gold: 4 1 2 2 # positions (4, 1), (2, 2) should be gold
```

Raise a `SyntaxError` with error message `Invalid Configuration File: format error!` if the file content is not as the following format (five lines with the required labels):

```
Frame: ...
Water: ...
Wood: ...
Food: ...
Gold: ...
```

Check the content of Frame:

- If it is not in the format of `widthxheight`, raise a `SyntaxError` with error message `Invalid Configuration File: frame should be in format widthxheight!`
- Width and height should be between [5, 7]. if not, raise an `ArithmeticError` with error message `Invalid Configuration File: width and height should range from 5 to 7!`

Check from the second line to the last line:

- If a line contains non-integer characters, raise a `ValueError` with error message `Invalid Configuration File: <line_name> (e.g., Water) contains non integer characters!`

- If a line has an odd number of elements, raise a SyntaxError with error message Invalid Configuration File: <line_name (e.g., Water)> has an odd number of elements!
- If a line contains a position that is out of the map, raise an ArithmeticError with error message Invalid Configuration File: <line_name> contains a position that is out of map.
- The following positions should not be occupied with resources or water: home bases (1, 1), (width-2, height-2), and the surrounding positions: (0, 1), (1, 0), (2, 1), (1, 2), (width-3, height-2), (width-2, height-3), (width-1, height-2), (width-2, height-1). Therefore, if those positions appear in the file, raise a ValueError with error message Invalid Configuration File: The positions of home bases or the positions next to the home bases are occupied!
- If a position appeared multiple times, raise a SyntaxError with error message Invalid Configuration File: Duplicate position (x, y)!

Print Configuration file <filepath> was loaded. if no errors.

Some Definitions

Print the prices:

```
Recruit Prices:
  Spearman (S) - 1W, 1F
  Archer (A) - 1W, 1G
  Knight (K) - 1F, 1G
  Scout (T) - 1W, 1F, 1G
```

(Please notice the double space before each label)

Display the map:

Print Please check the battlefield, commander.

Three example maps are shown below, and the print should follow the patterns.

```
  X00 01 02 03 04X
 Y+--------------+
00|~~|  |  |  |  |
01|  |H1|  |FF|GG|
02|WW|  |GG|  |~~|
03|  |~~|  |H2|  |
04|FF|  |WW|  |  |
 Y+--------------+
```

```
  X00 01 02 03 04 05X
 Y+------------------+
00|WW|  |  |  |  |  |
01|  |H1|  |  |GG|~~|
02|  |  |  |WW|  |  |
03|FF|~~|~~|~~|  |WW|
04|  |  |GG|  |  |  |
05|  |  |  |  |H2|  |
06|  |  |~~|WW|  |FF|
 Y+------------------+
```

```
  X00 01 02 03 04 05 06X
 Y+--------------------+
00|WW|  |  |  |  |  |  |
01|  |H1|  |  |GG|~~|  |
02|  |  |  |WW|  |  |  |
03|FF|~~|~~|~~|  |WW|  |
04|  |  |  |  |GG|  |  |
05|  |  |  |  |  |H2|  |
06|  |  |GG|  |WW|  |~~|
 Y+--------------------+
```

Print armies to move:

```
Armies to Move:
  Spearman: (x1, y1), (x2, y2)
  Archer: (x3, y3)
  Knight: (x4, y4), (x5, y5)
  Scout: (x6, y6)
```

(There is an empty line in the end)

It shows the positions of your armies that haven't moved in this stage in the order of recruitment time. Please notice the final empty line here. Omit the category if no unit in that category.

Game Walkthrough

Notes:

- User input should always be in a new line.
- All x and y represent sample numbers you need to replace with actual values according to the requirements.
- User can enter QUIT, DIS, or PRIS any time to quit the game, display the map, or print the prices, respectively except when the program is asking the commander's name with winning condition.

Walkthrough:

1. Print Configuration file <filepath> was loaded.
2. Print Game Started: Little Battle! (enter QUIT to quit the game) and an empty line.
3. **Display the map**, and print (enter DIS to display the map) and an empty line.
4. **Print the prices**, and print (enter PRIS to display the price list) and an empty line.
5. For each turn of player X (1 or 2, start from 1):
   a. Print -Year X- and an empty line. (X start from 617, and it should increment after both players finished their turn)
   b. Print +++Player X's Stage: Recruit Armies+++ and an empty line
   c. Print [Your Asset: Wood - x1 Food - x2 Gold - x3]
   d. If insufficient resources to recruit any units, print No resources to recruit any armies. and go to step **5-e**. If all four positions next to the home base are occupied, print No place to recruit new armies. And go to step **5-e**. Otherwise, print an empty line and ask Which type of army to recruit, (enter) 'S', 'A', 'K', or 'T'? Enter 'NO' to end this stage.
      i. Positive cases: S/A/K/T, print an empty line and ask You want to recruit a <Spearman/Archer/Knight/Scout>. Enter two integers as format 'x y' to place your army.
         1. Positive cases: unoccupied x y next to the home base
            a. apply the cost
            b. print an empty line, You has recruited a <Spearman/Archer/Knight/Scout>. and another empty line
            c. go to step **5-c** for more recruitments.
         2. Negative cases: Invalid inputs, print Sorry, invalid input. Try again. Go back to step **5-d-i**
         3. Edge cases:
            a. x y not next to the home base or occupied, print You must place your newly recruited unit in an unoccupied position next to your home base. Try again. Go back to step **5-d-i**.
            b. DIS, **display the map**, and go to step **5-d-i**
            c. PRIS, **print the prices**, and go to step **5-d-i**
            d. QUIT, terminate the game
      ii. Negative cases: invalid input, print Sorry, invalid input. Try again. Go back to step **5-d**.
      iii. Edge cases:
         1. insufficient resources, print Insufficient resources. Try again. Go back to step **5-d**
         2. NO, go to step **5-e**

3. DIS, **display the map**, and go to step **5-d**
4. PRIS, **print the prices**, and go to step **5-d**
5. QUIT, terminate the game

e. Print an empty line and ===Player X's Stage: Move Armies===.

f. Print an empty line. If Player X has no units to move, print No Army to Move: next turn. with an empty line and go to step **5-a**, the other player's turn. Otherwise **print armies to move**.

g. Ask Enter four integers as a format 'x0 y0 x1 y1' to represent move unit from (x0, y0) to (x1, y1) or 'NO' to end this turn.

i. Positive cases: four valid integers
1. Print an empty line and You have moved <Spearman/Archer/Knight/Scout> from (x0, y0) to (x1, y1). (It still should be (x1, y1) even if the Scout (T) died in between)
2. behave as **Move Results**
3. go back to step **5-f**.

ii. Negative cases: invalid inputs, Print Invalid move. Try again. Go back to step **5-f**.

iii. Edge cases:
1. NO, print an empty line. go to step **5-a**, the other player's turn.
2. DIS, **display the map**, and go to step **5-f**
3. PRIS, **print the prices**, and go to step **5-f**
4. QUIT, terminate the game

Move Results

Valid Moves

- Each army can move along four directions, including north, east, south, and west.
- Spearman (S), Archer (A), and Knight (K) can move one step in each turn while Scout (T) can move up to two steps (one step or two steps in the same direction). However, each army including Scout can move only once each turn.
- If an army moves to water or a counter enemy, it will disappear. Print We lost the army <Spearman/Archer/Knight/Scout> due to your command!
- If an army moves to an enemy with the same type, both it and the enemy will disappear. Print We destroyed the enemy <Spearman/Archer/Knight/Scout> with massive loss!
- If an army moves to a counter enemy, the enemy will disappear, and your army will move to that position. Print Great! We defeated the enemy <Spearman/Archer/Knight/Scout>!
- If an army moves to a resource, the resource will disappear, and your army will move to that position. Print Good. We collected 2 <Wood/Food/Gold>. Scout (T) can collect two fields of resources (with two messages of collection) if they are on their path.
- Although Scout (T) can move up to two steps, something in-between may destroy it.
- If an army moves to the enemy's home base (or a Scout (T) passes the enemy's home)
  - Print The army <Spearman/Archer/Knight/Scout> captured the enemy's capital. And an empty line
  - Ask What's your name, commander? QUIT, DIS, or PRIS will be treated as the commander's name.
  - Print an empty line and ***Congratulation! Emperor <name> unified the country in <year>.***
  - Terminate the game.

Invalid Moves

If the move is invalid, go to step **5-g-ii**.

- The army cannot move to its commander's other armies or home base but Scout (T) can pass its own commander's another army or home base to reach the second step.
- The army cannot move outside of the game map.
- The army cannot move from (x, y) to (x, y).

Restrictions

- You are free to import any module pre-installed in the default Ed server environment (a white list)
- Don't use exit() or sys.exit() in load_config_file(filepath) of little_battle.py.
- Not allowed to use any source code outside Ed server environment
- Please make sure it is your individual work (We have reported some cases of academic dishonesty to Educational Integrity Office for assignment 1).
- Hardcoding will result in zero in auto-marking part. Your tutor will report if any hardcoding is found.

Code Submission

- To make a submission, you will need to press the "Mark" button.
- You may submit as many times as you wish without penalty.
- You are able to view previous submissions from the code submission section.
- Every submission you make includes the little_battle.py and file_loading_tests.py (with invalid files you create) files

The following rules apply to your code submission:

- little_battle.py will be graded by the automarker
- Only the last submission will be graded. Both the final little_battle.py and file_loading_tests.py (with the invalid files you create) files of the last submission will be graded by your tutor.
- Submission after the due date will incur a late penalty as described in the unit of study outline.
- Your submission must be able to compile and run within the Ed environment provided.

After each submission, the marking system will automatically check your code against the public test cases.

The Python version that is presently being used on Ed system is Python 3.9.2

Please ensure you carefully follow the assignment specification. Your program output must exactly match the output shown in the specification.

Late is late!

The computer does not discriminate about what is late. There are two files for submission

- little_battle.py
- file_loading_tests.py (with the invalid files you create)

They are both in your Ed workspace. Make sure these are the ones for your submission.

Changes to these files after the deadline will incur a penalty for both (not individually).

Marking Criteria

This assignment is 20% of your final course grade.

The grade is based on the number of test cases passed as well as manual marking by your tutor.

We will provide you with some sample test cases, but these do not test all the functionality described in the assignment. It is crucial that you thoroughly test your own code.

Automatic Tests 14 / 20

These are marked by a computer,

- public test cases
- hidden test cases
- private test cases

The test cases will be released progressively, so you need to write your own test cases to test your codes first. Test cases will be varied after the deadline.

Manual Marking 6 / 20

Your tutor will consider:

- the correctness of exception handling (raise the correct exception for the given scenarios). 1 / 6
- the quality style (0.3), layout (0.3), and comments (0.4) of your code. 1 / 6
  - Please ensure your code is comprehensible for the purposes of marking. We recommend that your code adheres to the PEP 8 style guide. Note that we adopt the two-space indentation instead of four-space to increase readability in this assignment.
  - code and logical structures, readability, appropriateness of code comments.
- evaluation of your tests. 4 / 6
  - 0.1/1 for each appropriate invalid file in folder “invalid_files” (10 files in total)
  - 0.3/3 for completion of each unit test in file_loading_tests.py (10 tests in total)
  - Please refer to the Ed assignment workspace

Academic declaration

By submitting this assignment, you declare the following:

I declare that I have read and understood the University of Sydney Academic Dishonesty and Plagiarism in Coursework Policy, and except where specifically acknowledged, the work contained in this assignment or project is my own work and has not been copied from other sources or been previously submitted for award or assessment.

I understand that failure to comply with the Academic Dishonesty and Plagiarism in Coursework Policy, can lead to severe penalties as outlined under Chapter 8 of the University of Sydney By-Law 1999 (as amended). These penalties may be imposed in cases where any significant portion of my submitted work has been copied without proper acknowledgment from other sources, including published work, the internet, existing programs, the work of other students, or work previously submitted for other awards or assessments.

I rely that I may be asked to identify those portions of the work contributed by me and required to demonstrate my knowledge of the relevant material by answering oral questions or by undertaking supplementary work, either written or in the laboratory, in order to arrive at the final assessment mark.

I acknowledge that the School of Computer Science, in assessing this assignment, may reproduce it entirely, may provide a copy to another member of faculty, and or communicate a copy of this assignment to a plagiarism checking service or in house computer program, and that a copy of the assignment may be maintained by the service or the School of Computer Science for the purpose of future plagiarism checking.

Warning: Any attempts to deceive or disrupt the marking system will result in an immediate zero for the entire assignment. Negative marks can be assigned if you do not properly follow the assignment description or your code is unnecessarily or deliberately obfuscated.