

1. **Map:** The Universe is a 50x50 square.
  - a. Map locations referenced by location. 1.1 - 50.50
    - i. 1.1 is upper left corner, 50.50 is lower left
  - b. Player A's home base takes up 4 squares, starting at square 9.24, ending square 10.25
  - c. Player B's home base takes up 4 squares, starting at square 39.24, ending at square 40.25
  - d. **Resource collection squares located as follows**
    - i. 10.10
    - ii. 10.40
    - iii. 40.10
    - iv. 40.40
    - v. 20.20
    - vi. 20.20
    - vii. 30.30
    - viii. 30.20
  - e. **Units:**
    - i. **Hand To Hand unit:**
      1. Range: 1 Square in any direction
      2. Damage: TBD
      3. Speed: 1 square in any direction
      4. Cost: TBD
      5. Build time: TBD
    - ii. **Ranged Unit:**
      1. Range: 2 squares in any direction
      2. Damage: TBD
      3. Speed: 1 square in any direction
      4. Cost: TBD
      5. Build time: TBD
2. **Cycle**
  - a. The game is played in cycle, at the start of the game, the system randomly determines which play goes first. A cycle is one players process of their actions. They go in the following steps
    - i. Download map data
      1. XML or JSON from the server will tell an AI it is their cycle.
      2. Cycle data is as follows:
        - a. Cycle Number: One AI is even cycles, the other is odd, and this tells how many cycles have been performed
        - b. object with arrays of 2,500 map locations with the following info
          - i. Location: 1.1
          - ii. Unit ID: 12

- iii. Unit Team: 1
  - iv. Unit Type: 2
  - v. Unit Hit Points: 10
- 3. Process this map information and have AI determine its cycle.  
Cycles processed in the following order
  - ii. **Upload Actions:** After processing the map cycle state, your AI makes determinations on what needs to happen and then uploads a JSO/XML with its Collect, Strike Phase, Move Phase, and Build orders. They will be executed in those orders.
  - iii. **Collect:** For every resource square your unit sits on, you get TBD resources
  - iv. **Strike:** The next phase of a cycle is the “strike” phase.
    - 1. Strike phases must be labeled “Strike” in the JSON/XML as an object as follows
      - a. Strike Phase: sent with the left number the attacking unit, and the right number the defending unit.
        - i. `strike{ [1.1,1.2] [2.1,1.2] [2.2,1.2] }`
      - b. The server will process the strikes and move to the next phase
  - v. **Move Phase:** Second Phase of the cycle.
    - 1. Strike phases must be labeled “Strike” in the JSON/XML as an object as follows
      - a. Strike Phase: sent with the left number the attacking unit, and the right number the defending unit.
        - i. `move{ [1.1,1.2] [2.1,1.2] [2.2,1.2] }`
      - b. The server will process the moves and move to the next phase.
      - c. If you try to move into a square already occupied, the server will ignore that move. example, if you have 3 units attack 1 enemy unit and you assume you should have killed the enemy and try to move into that square, however failed to kill the enemy in the Strike phase, your move will be ignored by the server as invalid. This could have a chain reaction for other moves that get invalidated.
  - vi. **Build Phase:** from your HQ, you que up the server to build new units. No more than 12 new units may spawn during any cycle, but unlimited may be queued up.
    - 1. Qued up units are not displayed to the opposing AI until they spawn.