```
-Day & night cycles
    -Certain actions can only be performed during day or night

-Board
    -NxM rectangle from (0,0) to (N-1, M-1)
    -Can have either street or building or character
    -Player cannot move on building
    -Streets can have money, food, weapons
    -Characters in grid:
        -Street = '.'
        -Building = 'B'
        -Gun = 'G'
        -Bazooka = 'Z'
        -Money = 'M'
        -Food = 'F'
        -Warrior = 'w'
        -Builder = 'b'
        -No weapon = 'n'
        -Hammer = 'h'
        -Gun = 'g'
        -Bazooka = 'b'

-Player classes
    -Builder    (1 in clan)
    -Warriors   (3 in clan)

-Barricades
    -Can only be built by builders
    -Only buildable during the day
    -Can build a barricade at 4 adjacent cells (not diagonally)
    -Cannot build a barricade if inside a barricade
    -Can be occupied by any clan member, only of the clan that built it
    -Initial resistance strenghtened by repeated construction ops.
    -Max number of barricades =
    -Barricades disappear when next day begins
    -citizen in barricade cannot be attacked unless he is attacking
    -If barricade is demolished, citizen inside is not hurt

-Warriors
    -Always carry a weapon
    -They start with a hammer, but can take gun / bazooka
    -During the day:
        -Can only move horizontally / vertically
        -Can enter a barricade
        -If cell moved to has money, he collects it, money disappears
        -If cell moved to has food, he gains health, food disappears. Cannot go
         past max health
        -If cell moved to has weapon, he takes weapon and changes carried weapon to
         best weapon in arsenal.
        -Hammer < gun < bazooka

    -During the night:
        -Cannot move to the same space as same clan member
        -If cell moved to is from another clan, fight starts:
            -Defeated citizen loses fixed number of HP
            -Fight determined by strenght of opponents:
                -Builders have the same strenght always
                -Warriors' strenght depends on carried weapon
                -Between citizens with strenghts N and M, former has possibility of
```

```
                        N/(N+M), and latter M/(N+M).
                    -If after fight a citizen loses all HP, survivor receives certain
                     nº of points
          -If cell moved to is a rival barricade, it loses resistance points = to
           demolition strenght of weapon carried
          -If barricade loses all res. points, it disappears but warrior will not
           move


    -Builders
        -Builders never carry weapons
        -Same day movement as warriors
        -If cell moved to has weapon, weapon disappears
        -During the night:
            -If cell moved to is rival citizen, fight starts
            -If cell moved to is rival barricade, demo strenght is limited


    -Object regeneration
        -After food / money disappear, they regenerate after certain num. of rounds
        -Weapons and citizens also regenerate
        -Weapons are replaced by weapons of same type
        -Citizens respawn with initial HP
        -Warriors respawn with hammer
        -Object C doesn't respawn if citizen is near 2 cells in any direction
        -If object doesn't have empty cell to respawn, it doesn't
        -Respwn priority: Citizens > food / money > weapons


    -Order execution
        -One order x citizen for 1 round
        -If >1 order, only first is executed
        -Each round selected orders to 4 players are executed in random order
        -Randomness still kinda respects order given, consider giving order:
          most urgent -> least urgent
        -Order given can change depending on outside factors, ie. if order is go left,
          take food but food is eaten by rival clanmate, then order becomes attack
          rival



    Parameter                   Default value   Range
    NUM PLAYERS                 4               [4, 4]
    NUM DAYS                    5               [1, ∞)
    NUM ROUNDS PER DAY          50              [2, ∞) (even)
    BOARD ROWS                  15              [12, 25]
    BOARD COLS                  30              [12, 50]
    NUM INI BUILDERS            4               [1, 6]
    NUM INI WARRIORS            2               [1, 4]
    NUM INI MONEY               10              [0, 10]
    NUM INI FOOD                5               [0, 10]
    NUM INI GUNS                4               [0, 5]
    NUM INI BAZOOKAS            2               [0, 4]
    BUILDER INI LIFE            60              [1, ∞)
    WARRIOR INI LIFE            100             [1, ∞)
    MONEY POINTS                5               [1, ∞)
    KILL BUILDER POINTS         100             [1, ∞)
    KILL WARRIOR POINTS         250             [1, ∞)
    FOOD INCR LIFE              20              [1, ∞)
    LIFE LOST IN ATTACK         20              [1, ∞)
    BUILDER STRENGTH ATTACK     1               [1, ∞)
    HAMMER STRENGTH ATTACK      10              [1, ∞)
    GUN STRENGTH ATTACK         100             [1, ∞)
```

```
BAZOOKA STRENGTH ATTACK     1000                [1, ∞)
BUILDER STRENGTH DEMOLISH   3                   [1, ∞)
HAMMER STRENGTH DEMOLISH    10                  [1, ∞)
GUN STRENGTH DEMOLISH       10                  [1, ∞)
BAZOOKA STRENGTH DEMOLISH   30                  [1, ∞)
NUM ROUNDS REGEN BUILDER    50                  [1, ∞)
NUM ROUNDS REGEN WARRIOR    50                  [1, ∞)
NUM ROUNDS REGEN FOOD       10                  [1, ∞)
NUM ROUNDS REGEN MONEY      5                   [1, ∞)
NUM ROUNDS REGEN WEAPON     40                  [1, ∞)
BARRICADE RESISTANCE STEP   40                  [1, ∞)
BARRICADE MAX RESISTANCE    320                 [1, ∞)
MAX NUM BARRICADES          3                   [1, ∞)
```

-Programming the player
    -In own copy of AI*playername*.cc:
        -play() is called every round and decides the actions / orders
        -can define auxiliar type definitions, variables and methods inside your
         player class, but play() is always start point of code.
        -can call functions from files:
            -State.hh   -> Access state of game
            -Action.hh -> Gives orders to citizens
            -Structs.hh -> Useful data structs
            -Settings.hh-> Access game parameters
            -Player.hh -> method me()
            -Random.hh -> RNG
        -AIDemo.cc -> example of how to use these functions
        -NOTE: CANNOT EDIT factory() OR LAST LINE THAT ADDS PLAYER TO POOL

-Playing against the dummy
    -Dummy player only provided through .o file, used for test battles
    -To add Dummy (or other player.o) into game, you must edit Makefile and set
     variable DUMMY_OBJ to correct value

-Restrictions to submitting a player
    -All source code must be in single file (ie. AI*playername*.cc)
    -Cannot use global variables (can use attributes in class function)
    -Can only use std libraries.
    -Cannot use syscalls
    -Game time limit is 1 second.
    -Cannot use cout, cin, cerr in final program
    -If player defeats Dummy, player name is locked, further submissions must
     change player name

-Pro Tips 420 blaze it
    -Read only headers, don't worry about implementation
    -Define basic auxiliary methods, make sure they work
    -Obvious compile often and clean code blabla
    -Use assert to make sure everything is good
    -Switch on DEBUG in Makefile to debug better if things go to shit
    -Keep copies of old versions of player. Make it fight with new version to see
     improvement
    -Make sure the program is fast like Sonic