# Missile Command

1.  The player begins the game with 6 cities to protect from enemy missiles.  The player's cities are the 2x3 rectangles on the board.

2.  Each round the enemy will launch 12 missiles at the cities.  The player has 30 of his own missiles to use to shoot down the enemy missiles.

    2.1.  Each round consists of turns in which the player may choose to launch a missile; the enemy may launch its own; and any existing missiles move.

    2.1.1. First the missiles on the board move.  The player's missiles move 3 spaces vertically each turn.  The enemy missiles move 1 space during the first round, 2 during the second and so on.

    2.1.1.1.    If an enemy missile hits one of the cities, the city and missile are destroyed.

    2.1.1.2.    If an enemy missile hits the player's missile launcher, the player loses 10 missiles for that round and the missile is destroyed

    2.1.1.3.    If an enemy missile hits the ground  between cities the missile is destroyed.

    2.1.1.4.    If an enemy missile passes through an explosion, it is destroyed and the player gets 50 points.

    2.1.1.5.    If a player's missile reaches its destination it explodes.  Explosions grow over a series of turns.

    2.1.2. Each turn, after movement the current explosions change sized based on the number of turns since they exploded.  Exploding missiles follow the following pattern, where the central point is the location targeted by the player:

    ```
    Turn 0/6:                   *

                                *
    Turn 1/5:               *   *   *
                                *


                                *
                            *   *   *
    Turn 2-4:           *   *   *   *   *
                            *   *   *
                                *
    ```

    2.1.3. If the player has any remaining missiles, he may launch one by marking a destination on the board.  Then a line is drawn from the missile launcher (the box on top of the central rectangle) to that destination to show the missile's flight path.  A missile is placed on top of the missile launcher and one is subtracted from the player's remaining missile total.

2.1.4. An enemy missile may be spawned each turn. The following algorithm is used to spawn enemy missiles (pesudocode is given at the end of the instructions for more clarity):

Roll a d6; if it equals 1:

Roll a d6. Divide the result by 2 and round up and do the following that many times (essentially, if a 1 or 2, do it once; 3 or 4, twice; and 5 or 6, three times):

Roll a 1d6. If the result is less than 4:

Roll a d20 and spawn the missile on the corresponding space.

If the result is greater than or equal to 4:

Roll a d20 and spawn the missile on the space equal to roll+20.

Roll a d6. If the result is 1 or 2:

Missile direction is down and to the left

If the result is 3 or 4:

Missile direction is straight down

If the result is 5 or 6:

Missile direction is down and to the right

If the chosen direction will make the missile go off the side of the board, reroll the direction

2.2. The round ends when either the enemy has used all 12 of its missiles and they have all been destroyed or when all the cities have been destroyed, at which point the game is over.

2.3. At the end of the round, if the player has any cities remaining, he gets bonus points equal to 100 times the number of remaining cities and 5 times the number of remaining missiles.

# AI Missile Spawning Pseudocode

```
Each turn {
    roll = 1d6
    if roll == 1 {
        roll = 1d6
        for each ceiling(roll/2) {
            roll = 1d6
            if roll < 4 {
                Spawn missile on space 1d20
            } else {
                Spawn missile on space 1d20 + 20
            }

            do {
                roll = 1d6
                if roll <= 2 {
                    Missile Direction = left
                } else if roll <= 4 {
                    Missile Direction = down
                } else {
                    Missile Direction = right
            } while missile will go off the side of the board
        }
    }
}
```