# KEY:

**Bolded words** are field names

*Italicized words* are parameters

Underlined words are methods that need to be interfaceable with other classes. Other methods are pretty much all internal methods

(?) marks a design question that needs to be considered

# Main

The **Main** class (name unofficial as of yet) runs the program. In its main loop, it calls:

1. Player.getInput(input)
2. Fire.move()
   1. Next to call this for each instance of Fire
3. Level.spawnDragons()
4. Dragon.run()
   1. Need to call this for each instance of Dragon currently in the level
5. Display functions
   1. Level
   2. Wall
   3. Dragon
   4. Fire
   5. Player
6. Screen updating functions

# Player

* **position**
  + x
  + y
* getInput(*Input*)
  + *input* is whatever parameters need to be called for player to know what to do
    - if it ends up not needing any parameters (i.e. it has direct access to input vars), then it will have empty parameters
  + it tells the player what it needs to do based on *input*,
    - and then calls move() and fire() as necessary
* fire()
  + fires a projectile from the player’s current position, matching the player’s angle
* move()
  + moves the player up or down based on input

# Dragon

* **healthPoints** – the amount of HP a dragon has
* **position**
  + **x**
  + **y**
* run()
  + moves the dragon closer to its goal, and determines what it should do
  + calls checkGoal()
* checkGoal()
  + determines if the dragon has reached its goal (the wall)
* damage(*Damage*)
  + receives *damage* information and then processes it
  + may decrease HP or kill dragon, depending on HP left

# Fire

* **fireRate** – the rate at which the weapon reloads
  + (?) should this go in the Player class?
* **firePower** – the damage it does on hitting its target
* move()
  + moves the fire in its current trajectory
  + calls checkCollision()
* checkCollision()
  + loops through each of the dragons in the level and determines if it has collided with it
  + for each dragon it collides with, it:
    - calls doDamage(Dragon)
    - turns itself off (delete itself) (?)
      * although this could be negated, it IS a railgun afterall. Perhaps it go through the dragons without stopping?
* doDamage(*Dragon*)
  + calls *Dragon*.damage(Damage)
    - passes its own attributes, such as firepower

# Wall

* **healthPoints**
  + the amount of health points the wall has before being destroyed
* damage(*Damage*)
  + receives *Damage*
  + decreases health, and then
    - initiates game over if HP is zero
      * (?) do we want to put that code here?
      * (?) or have the Level or Main class check?

# Level

* spawnDragons()
  + determines how many dragons should spawn
  + and spawns them in level appropriate positions