**Space Invader Notes**

**World**

* Fields:
  + Player p: the player
  + ArrayList<ArrayList<Alien>> alienList: a list of columns of Aliens
  + ///////Hashmap<ArrayList<Alien>> alienList: a hashmap of columns of Aliens/////
  + Saucer s: saucer
  + int score: Player’s score
  + ArrayList<Missile>misList: a list of Missiles

* Methods:
  + World onTick():
    - maps act() over alienList and misList
    - checks if any Missile has hit an Alien, if so, call onHit() on Alien, if onHit() is true, remove Alien from alienList. Also remove Missile from misList. Add value of Alien to score.
    - checks if any Missile has hit the player, if so, call onHit() on player. Also remove Missile from misList.
    - randomly add a Saucer to alienList
    - removes Missiles from misList that have left the screen
    - if distMoved is past a certain threshold, map moveDown() and swapDir() over alienList
  + WorldImage makeImage(): produces an image representation of the World
  + World onKeyEvent(): moves Player or has Player shoot
  + WorldImage lastImage(String str): displays gameover/win screen with player’s final score.
  + WorldEnd worldEnds(): ends game when player has 0 lives or alienList is empty or Aliens have invaded Earth

**Actor**

* Fields:
  + int x: The x coordinate of the Actor
  + int y: The y coordinate of the Actor
  + WorldImage img: image representation of this Actor
  + int dx: the horizontal velocity of the Actor
  + int dy: the vertical velocity of the Actor
* Methods:
  + void act(): changes the state of the Actor
  + WorldImage draw-on(WorldImage img): draws the Actor onto the given WorldImage
  + void move()
  + boolean onHit(): updates Actor when they are hit, returns true if Actor is destroyed

**Player extends Actor** is a new Player(int x, int y, WordImage img, int lives)

* Fields:
  + int lives: The amount of lives the Player has
* Methods:
  + void fire(): Player shoots a Missile
  + void onHit(): decrements lives

**Alien extends Actor** is a new Alien(int x, int y, WordImage img)

* Fields:
  + int value: the amount added to the score when this Alien is destroyed
* Methods:
  + void act():
    - Update dx based on y value
    - Randomly calls fire()
    - Call move()
  + void fire(): shoots a Missile
  + void onHit(): returns true if collides with a Missile
  + void swapDir(): sets velocity to (\* -1 velocity)

**StrongAlien extends Alien** is a new StrongAlien(int x, int y, WordImage img)

* Fields:
* Methods:
  + void onHit(): create an Alien in place of this Strong Alien

**Saucer extends Actor** is a new Saucer(int x, int y, WordImage img)

* Fields:
  + int value: the amount added to the score when this Saucer is destroyed
  + boolean canMove: true if the saucer can move on screen
* Methods:
  + void act(): calls move(int v)
  + void onHit(): returns true

**Missile extends Actor**

* Fields:
* Methods:
  + void act(): calls moveUp() or moveDown() depending on movingUp
  + void move(): moves up or down based on dy

**Constants:**

* Initial Alien setup
* Player speed
* Alien start speeds
* Saucer speed
* Alien value
* StringAlien value
* Saucer value