**Space Invaders Design**

**World**

* Fields:
  + Player p: the player
  + ArrayList<ArrayList<Alien>> alienList: a list of rows of Aliens
  + int score: Player’s score
  + ArrayList<Missile> misList: a list of Missiles
  + int distMoved: the distance the aliens have moved since they last shifted downwards
* 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 spd: the speed of the Alien
* Methods:
  + void act(): changes the state of the Actor
  + WorldImage draw-on(WorldImage img): draws the Actor onto the given WorldImage
  + void moveLeft(int dist): moves Actor to the left
  + void moveRight(int dist): moves Actor to the left
  + 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
  + boolean goingRight: true if the Alien is moving rightwards
* Methods:
  + void act():
    - Update spd based on y value
    - Randomly calls fire()
    - Call moveLeft() or moveRight() depending on goingRight value.
  + void moveDown(): Alien drops down
  + void fire(): shoots a Missile
  + void onHit(): returns true
  + void swapDir(): sets goingRight to !goingRight

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

* Fields:
* Methods:
  + void act(): calls moveRight()
  + void moveDown(): Saucer does nothing
  + void onHit(): returns true

**Missile extends Actor**

* Fields:
  + boolean movingUp: true if missile moves up
* Methods:
  + void act(): calls moveUp() or moveDown() depending on movingUp
  + void moveUp(): moves Missile up
  + void moveDown(): moves Missile down

**Constants:**

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