Pseudocode for Space Invaders’ Key Features

Player Spaceship: Movement and Shooting

Player Movement

Input: self
Output: Move the player spaceship horizontally while keeping it within screen bounds.

```
Def move_left(self):
  If self.turtle.xcor() > -380:
    Move the spaceship (-15, 0)

Def move_right(self):
  If self.turtle.xcor() < 380:
    Move the spaceship (15, 0)
```

Shooting Mechanics

Input: None
Output: Fires a bullet from the player's spaceship.

```
Def fire_bullet():
  Access global variable last_bullet_time
  current_time ← get current time
  If (current_time - last_bullet_time) ≥ bullet_cooldown:
    If double_bullets is True:
      Create two bullets at (player.x ± 10, player.y + 20)
      Add both bullets to player_bullets list
    Else:
      Create one bullet at (player.x, player.y + 20)
      Add the bullet to player_bullets list
  Update last_bullet_time to current_time
```

Alien Spaceships: Movement and Attacks

Organized Alien Movement

Input: self, dx, dy
Output: Updates the position of each alien.

```
Def move(self, dx, dy):
   Update alien's x-coordinate ← alien.x + dx
   Update alien's y-coordinate ← alien.y + dy
```

Alien Projectile Firing

Input: None
Output: Fires projectiles from aliens at random intervals.

```
Def fire_alien_bullets():
   For each alien:
      If random.random() < fire_probability:
         Create a projectile at alien's position
         Add the projectile to alien_bullets list
```

Homing Bullets and Advanced Behavior

Homing Bullet Trajectory

Input: None
Output: Moves homing bullets towards the player.

```
Def move_forward():
   If curve_probability > 0:
      Compute target angle ← atan2(player.y - bullet.y, player.x - bullet.x)
      Limit angle difference to max_total_turn
      Update bullet heading incrementally towards target angle
   Move bullet forward at speed
```

Boss Mechanics

Boss Alien Health Bar

Input: Whether the alien boss exists and is alive
Output: Updates the health bar of the boss alien.

```
Def update_boss_health_bar():
   If boss_alien exists and is alive:
      Compute health_ratio ← boss_alien.health / max_health
      Draw red bar proportional to health_ratio below boss alien
```

```
  Else:
    Hide the health bar
```

Collision Detection

Collision Calculation

Input: obj1, obj2
Output: Checks for a collision between two objects.

```
Def is_collision(obj1, obj2):
  Compute distance ← sqrt((obj1.x - obj2.x)^2 + (obj1.y - obj2.y)^2)
  If distance < 20:
    Return True
  Else:
    Return False
```

Power-Ups: Enhancing Gameplay

Power-Up Application

Input: powerup_type
Output: Applies the effects of a power-up.

```
Def apply_powerup(powerup_type):
  If powerup_type is "faster_fire":
    Reduce bullet_cooldown
  Else if powerup_type is "extra_life":
    Increase player's lives
  Else if powerup_type is "double_bullets":
    Enable double bullet mode
  Effects expire after a set duration, reverting to default settings
```