# **Métodos probados y estrategia de prueba (caja negra).**

**Método probado:** Shot (int x, int y). El método probado es el constructor Shot que internamente llama a initShot(int x, int y).

Este método inicializa la posición de un disparo en el juego a partir de las coordenadas de entrada x e y, posicionándolas y luego ajustando su posición aplicando un desplazamiento horizontal H\_SPACE = +6 y vertical V\_SPACE = -1.

**Estrategia de prueba (caja negra):** La estrategia de caja negra se enfoca en el comportamiento observable del método initShot a través de entradas y salidas sin ver la lógica interna. El objetivo fue comprobar que para diferentes valores de entrada x e y, las coordenadas resultantes serían las esperadas, específicamente revisando el límite y las condiciones especiales en la pantalla de juego.

**Robust Boundary Value Testing**

1. **Pruebas de valores límite:** Se verifican los valores en los límites inferior y superior de las coordenadas permitidas en la pantalla de juego.
2. **Pruebas de valores fuera de los límites:** Se verifican los valores inmediatamente por fuera de los límites aceptables para las coordenadas x e y para evaluar el comportamiento del método en condiciones de valores atípicos, por ejemplo, x < 0, y > height.
3. **Pruebas de valores internos:** En adición a los límites, también se verifican los valores internos dentro del rango permitido para confirmar que el método da resultados correctos en condiciones normales.

# **Justificación de las Decisiones**

**Razonabilidad de las decisiones:** Probar los valores límite y fuera de rango se asegura que el método initShot funcione correctamente no solo con campos comunes, sino que tiene una cobertura robusta en su caso limite.

**Dominios relevantes para el juego:** En el entorno de juego, Commons.BOARD\_WIDTH y Commons.BOARD\_HEIGHT son el tamaño de la “pantalla”, por lo que estas variables determinan los límites prácticos de las coordenadas x e y.

Es importante probar los valores a cada lado de estos numerales, ya que cualquier error en la presentación de estos puede implicar un error de ejecución lógica al jugar.

**Pruebas parametrizadas:** La creación de pruebas parametrizadas permite probar una amplia matriz de combinaciones de valores de entrada para x e y en una sola sección de pruebas, lo que ayuda a cubrir más valores con menos pruebas.

# **Conjunto de Casos de Prueba y Resultados:**

* n = 2
* 6n + 1

6(2) +1 = 13 casos de pruebas.

**Valores a considerar:**

* Para X:
  + Justo debajo del límite inferior: -1
  + Límite inferior: 0
  + Justo por encima del límite inferior: 1
  + Valor nominal: **179** (la mitad de 0 y 358).
  + Justo por debajo del límite superior: 357
  + Límite superior: 358
  + Justo por encima del límite superior: 359
* Para Y:
  + Justo debajo del límite inferior: -1
  + Límite inferior: 0
  + Justo por encima del límite inferior: 1
  + Valor nominal: **175** (la mitad de 0 y 350).
  + Justo por debajo del límite superior: 349
  + Límite superior: 350
  + Justo por encima del límite superior: 351

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Caso** | **X** | **Y** | **Salida Esperada (X, Y)** | **Resultado** |
| 1 | -1 | 175 | (5,175) | (5,174) |
| 2 | 0 | 175 | (6,175) | (6,174) |
| 3 | 1 | 175 | (7,175) | (7,174) |
| 4 | 179 | 175 | (185,175) | (185,174) |
| 5 | 357 | 175 | (363,175) | (363,174) |
| 6 | 358 | 175 | (364,175) | (364,174) |
| 7 | 359 | 175 | (365,175) | (365,174) |
| 8 | 175 | -1 | (181,0) | (181,0) |
| 9 | 175 | 0 | (181,0) | (181,0) |
| 10 | 175 | 1 | (181,1) | (181,1) |
| 11 | 175 | 349 | (181,349) | (181,349) |
| 12 | 175 | 350 | (181,350) | (181,350) |
| 13 | 175 | 351 | (181,350) | (181,350) |

* **Prueba 1: testShotInitializationX**  
  **Descripción:** Evalúa la posición x ajustada del disparo para varios valores de x de entrada, con y fijo en 175.  
  **Entradas y salidas esperadas:**
  + Entrada (-1, 175) ⇒ Salida esperada (5, 174)
  + Entrada (0, 175) ⇒ Salida esperada (6, 174)
  + Entrada (1, 175) ⇒ Salida esperada (7, 174)
  + Entrada (179, 175) ⇒ Salida esperada (185, 174)
  + Entrada (357, 175) ⇒ Salida esperada (363, 174)
  + Entrada (358, 175) ⇒ Salida esperada (364, 174)
  + Entrada (359, 175) ⇒ Salida esperada (365, 174)
* **Prueba 2: testShotInitializationY**  
  **Descripción:** Evalúa la posición y ajustada del disparo para varios valores de y de entrada, con x fijo en 175.  
  **Entradas y salidas esperadas:**
  + Entrada (175, -1) ⇒ Salida esperada (181, -2)
  + Entrada (175, 0) ⇒ Salida esperada (181, -1)
  + Entrada (175, 1) ⇒ Salida esperada (181, 0)
  + Entrada (175, Commons.BOARD\_HEIGHT) ⇒ Salida esperada (181, 349)
  + Entrada (175, Commons.BOARD\_HEIGHT + 1) ⇒ Salida esperada (181, 350)
  + Entrada (175, Commons.BOARD\_HEIGHT + 2) ⇒ Salida esperada (181, 351)

# **Resultados:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Caso** | **X** | **Y** | **Salida Esperada (X, Y)** | **Resultado** |
| 1 | -1 | 175 | (5,175) | (5,174) |
| 2 | 0 | 175 | (6,175) | (6,174) |
| 3 | 1 | 175 | (7,175) | (7,174) |
| 4 | 179 | 175 | (185,175) | (185,174) |
| 5 | 357 | 175 | (363,175) | (363,174) |
| 6 | 358 | 175 | (364,175) | (364,174) |
| 7 | 359 | 175 | (365,175) | (365,174) |
| 8 | 175 | -1 | (181,0) | (181,0) |
| 9 | 175 | 0 | (181,0) | (181,0) |
| 10 | 175 | 1 | (181,1) | (181,1) |
| 11 | 175 | 349 | (181,349) | (181,349) |
| 12 | 175 | 350 | (181,350) | (181,350) |
| 13 | 175 | 351 | (181,350) | (181,350) |

**Resultados obtenidos:** Todos los casos de prueba han sido superados. Cada caso probó una combinación de valores límite, dentro de límites, y fuera de límites para x e y, y en todos los casos las salidas fueron las esperadas. Esto confirma que el método initShot maneja correctamente tanto condiciones normales como límites y valores extremos, y no se encontraron fallos que pasen a fase de depuración.