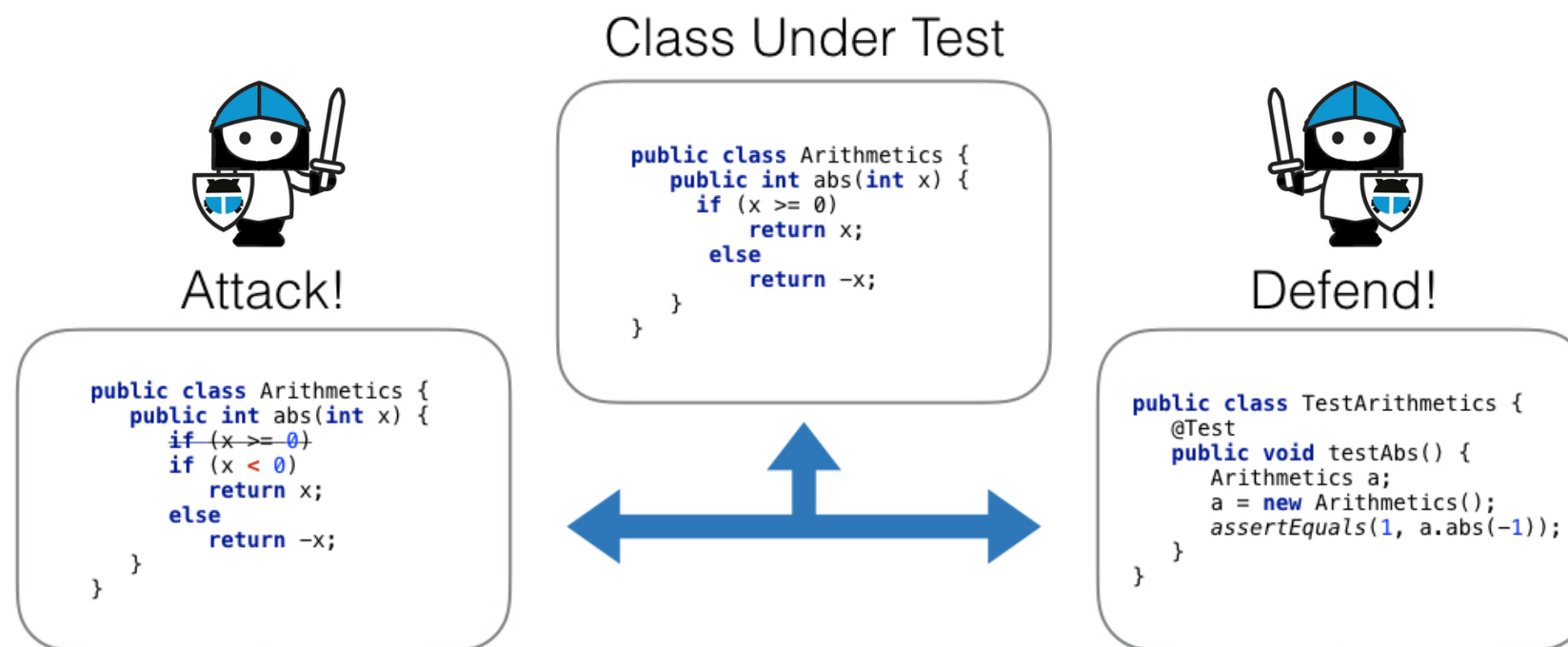




Code Defenders

Teaching Software Testing Concepts using a Mutation Testing Game

Benjamin Clegg, **José Miguel Rojas**, Gordon Fraser





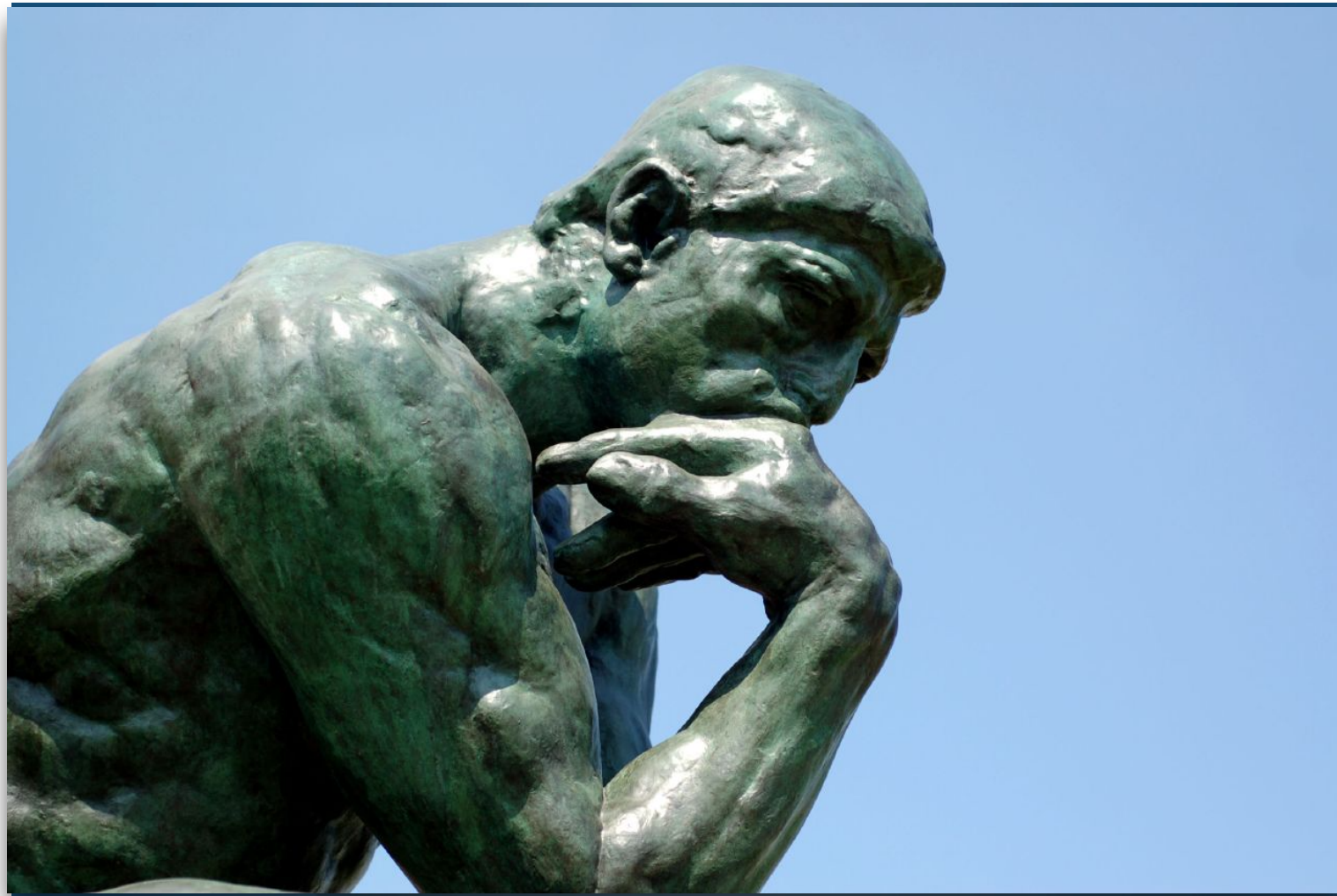
Software Testing



Fundamental for Software Quality



Software Testing



by Brian Hillegas (Flickr)

Fundamental for Software Quality
...but *very* demanding



Software Testing Education





Code Defenders



by François Philipp (Flickr)

Software Testing Gamification



Mutation Testing



Mutation Testing

Helps shifting testing effort from coverage to fault detection



Mutation Testing

Helps shifting testing effort from coverage to fault detection



Mutation Testing

Helps shifting testing effort from coverage to fault detection

```
int abs(int x) {  
    if (x >= 0)  
        return x;  
    else  
        return -x;  
}
```



Mutation Testing

Helps shifting testing effort from coverage to fault detection

```
@Test  
void testAbs() {  
    int res = abs(42);  
    assertEquals(42, res);  
}
```

```
int abs(int x) {  
    if (x >= 0)  
        return x;  
    else  
        return -x;  
}
```



Mutation Testing

Helps shifting testing effort from coverage to fault detection

```
@Test  
void testAbs() {  
    int res = abs(42);  
    assertEquals(42, res);  
}
```



```
int abs(int x) {  
    if (x >= 0)  
        return x;  
    else  
        return -x;  
}
```



Mutation Testing

Helps shifting testing effort from coverage to fault detection

```
@Test
void testAbs() {
    int res = abs(42);
    assertEquals(42, res);
}
```



```
int abs(int x) {
    if (x >= 0)
        return x;
    else
```

```
int abs(int x) {
    if (x <= 0)
        return x;
    else
        return -x;
}
```




Mutation Testing

Helps shifting testing effort from coverage to fault detection

```
@Test
void testAbs() {
    int res = abs(42);
    assertEquals(42, res);
}
```



```
int abs(int x) {
    if (x >= 0)
        return x;
    else
```

return x;



```
int abs(int x) {
    if (x <= 0)
        return x;
    else
        return -x;
}
```



Mutation Testing

Helps shifting testing effort from coverage to fault detection

```
@Test
void testAbs() {
    int res = abs(42);
    assertEquals(42, res);
}
```



```
int abs(int x) {
    if (x >= 0)
        return x;
    else
```

return -x;



```
int abs(int x) {
    if (x <= 0)
        return x;
    else
        return -x;
}
```

Too many mutants, often trivial, redundant or equivalent



Code Defenders

Class Under Test

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x >= 0)  
            return x;  
        else  
            return -x;  
    }  
}
```



Code Defenders

Class Under Test

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x >= 0)  
            return x;  
        else  
            return -x;  
    }  
}
```

Attackers





Code Defenders

Class Under Test

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x >= 0)  
            return x;  
        else  
            return -x;  
    }  
}
```

Attackers



Defenders





Code Defenders

Class Under Test

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x >= 0)  
            return x;  
        else  
            return -x;  
    }  
}
```

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x < 0)  
            return x;  
        else  
            return -x;  
    }  
}
```

Attackers



Defenders



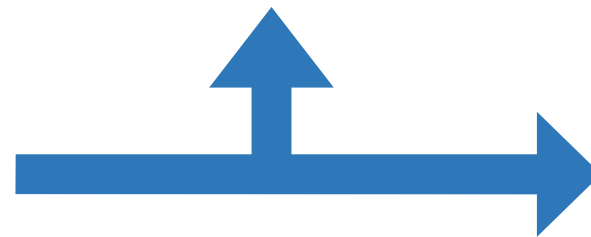


Code Defenders

Class Under Test

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x >= 0)  
            return x;  
        else  
            return -x;  
    }  
}
```

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x < 0)  
            return x;  
        else  
            return -x;  
    }  
}
```



Attackers



Defenders





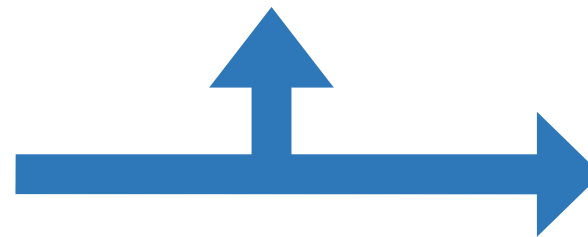
Code Defenders

Class Under Test

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x >= 0)  
            return x;  
        else  
            return -x;  
    }  
}
```

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x < 0)  
            return x;  
        else  
            return -x;  
    }  
}
```

```
public class TestArithmetics {  
    @Test  
    public void testAbs() {  
        Arithmetics a;  
        a = new Arithmetics();  
        assertEquals(1, a.abs(-1));  
    }  
}
```



Attackers



Defenders





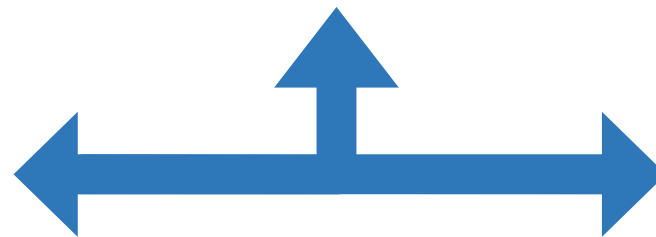
Code Defenders

Class Under Test

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x >= 0)  
            return x;  
        else  
            return -x;  
    }  
}
```

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x < 0)  
            return x;  
        else  
            return -x;  
    }  
}
```

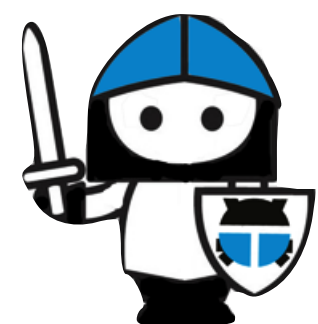
```
public class TestArithmetics {  
    @Test  
    public void testAbs() {  
        Arithmetics a;  
        a = new Arithmetics();  
        assertEquals(1, a.abs(-1));  
    }  
}
```



Attackers



Defenders





Code Defenders

Score points for
surviving mutants

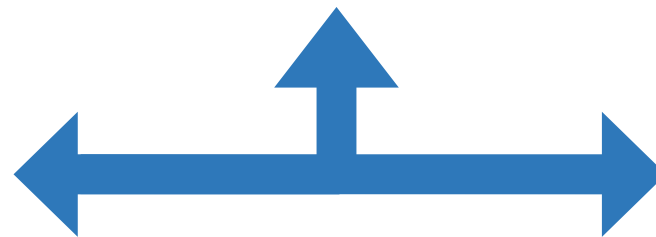
```
public class Arithmetics {  
    public int abs(int x) {  
        if (x < 0)  
            return x;  
        else  
            return -x;  
    }  
}
```

Attackers



Class Under Test

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x >= 0)  
            return x;  
        else  
            return -x;  
    }  
}
```



```
public class TestArithmetics {  
    @Test  
    public void testAbs() {  
        Arithmetics a;  
        a = new Arithmetics();  
        assertEquals(1, a.abs(-1));  
    }  
}
```

Defenders





Code Defenders

Score points for
surviving mutants

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x < 0)  
            return x;  
        else  
            return -x;  
    }  
}
```

Attackers



Class Under Test

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x >= 0)  
            return x;  
        else  
            return -x;  
    }  
}
```

Score points for
effective tests

```
public class TestArithmetics {  
    @Test  
    public void testAbs() {  
        Arithmetics a;  
        a = new Arithmetics();  
        assertEquals(1, a.abs(-1));  
    }  
}
```

Defenders





Code Defenders

Score points for
surviving mutants

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x < 0)  
            return x;  
        else  
            return -x;  
    }  
}
```

Attackers



Class Under Test

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x >= 0)  
            return x;  
        else  
            return -x;  
    }  
}
```

Score points for
effective tests

```
public class TestArithmetics {  
    @Test  
    public void testAbs() {  
        Arithmetics a;  
        a = new Arithmetics();  
        assertEquals(1, a.abs(-1));  
    }  
}
```

Defenders



Equivalent Mutant Duels



Code Defenders

Score points for
surviving mutants

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x < 0)  
            return x;  
        else  
            return -x;  
    }  
}
```

Attackers



Class Under Test

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x >= 0)  
            return x;  
        else  
            return -x;  
    }  
}
```

Score points for
effective tests

```
public class TestArithmetics {  
    @Test  
    public void testAbs() {  
        Arithmetics a;  
        a = new Arithmetics();  
        assertEquals(1, a.abs(-1));  
    }  
}
```

Defenders



equivalent!

Equivalent Mutant Duels



Code Defenders

Class Under Test

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x >= 0)  
            return x;  
        else  
            return -x;  
    }  
}
```

Score points for
surviving mutants

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x < 0)  
            return x;  
        else  
            return -x;  
    }  
}
```

Attackers



no way! here is
a killing test!

Score points for
effective tests

```
public class TestArithmetics {  
    @Test  
    public void testAbs() {  
        Arithmetics a;  
        a = new Arithmetics();  
        assertEquals(1, a.abs(-1));  
    }  
}
```

Defenders



equivalent!

Equivalent Mutant Duels



Code Defenders

Class Under Test

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x >= 0)  
            return x;  
        else  
            return -x;  
    }  
}
```

Score points for
surviving mutants

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x < 0)  
            return x;  
        else  
            return -x;  
    }  
}
```

Attackers



no way! here is
a killing test!

Score points for
effective tests

```
public class TestArithmetics {  
    @Test  
    public void testAbs() {  
        Arithmetics a;  
        a = new Arithmetics();  
        assertEquals(1, a.abs(-1));  
    }  
}
```

Defenders



equivalent!

Equivalent Mutant Duels



Code Defenders

Class Under Test

Score points for
surviving mutants

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x < 0)  
            return x;  
        else  
            return -x;  
    }  
}
```

Attackers



no way! here is
a killing test!

oh no! :(

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x >= 0)  
            return x;  
        else  
            return -x;  
    }  
}
```

Score points for
effective tests

```
public class TestArithmetics {  
    @Test  
    public void testAbs() {  
        Arithmetics a;  
        a = new Arithmetics();  
        assertEquals(1, a.abs(-1));  
    }  
}
```

Defenders



equivalent!

Equivalent Mutant Duels



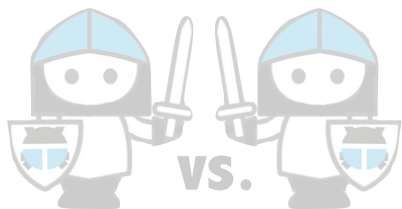
Code Defenders



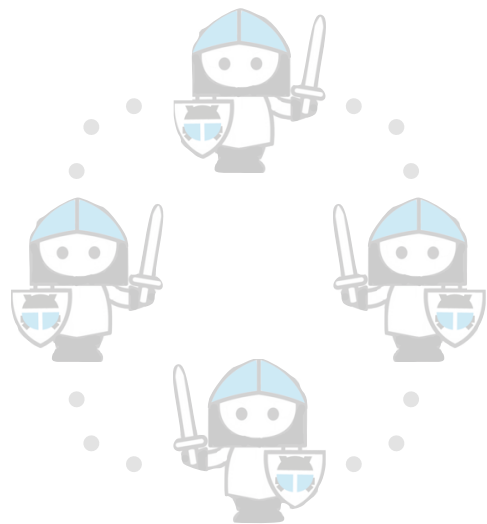
vs.

Single-player

EVASUITE / MAJOR



Duels



Multi-player

Code Defenders - Game 2260

code-defenders.org/multiplayer/play?id=2260

Code Defenders

games upload class leaderboard help netwalkshaw

DEFENDER::ACTIVE SparseIntArray Show Scoreboard

Class Under Test Write a new JUnit test here Defend!

```
35 /**
36  * Creates a new SparseIntArray containing no mappings that will not
37  * require any additional memory allocation to store the specified
38  * number of mappings. If you supply an initial capacity of 0, the
39  * sparse array will be initialized with a light-weight representation
40  * not requiring any additional array allocations.
41  */
42 public SparseIntArray(int initialCapacity) {
43     if (initialCapacity == 0) {
44         mKeys = SparseIntArray.EMPTY_INT_ARRAY;
45         mValues = SparseIntArray.EMPTY_INT_ARRAY;
46     } else {
47         mKeys = new int[initialCapacity];
48         mValues = new int[mKeys.length];
49     }
50     mSize = 0;
51 }
52
53 /**
54  * Given the current size of an array, returns an ideal size to which the array should grow.
55  * This is typically double the given size, but should not be relied upon to do so in the
56  * future.
57  */
58 public static int growSize(int currentSize) {
59     return currentSize <= 4 ? 8 : currentSize + (currentSize >> 1);
60 }
61 }
```

```
1 /* no package name */
2
3 import org.junit.*;
4 import static org.junit.Assert.*;
5
6 public class TestSparseIntArray {
7     @Test(timeout = 4000)
8     public void test() throws Throwable {
9         // test here!
10     }
11 }
12 }
```

Existing Mutants

alive (35)	killed (68)	equivalent (5)
Search...		
Mutant 5838 Creator: amin [UID: 428]		Claim Equivalent
Modified line 244		
Mutant 5855 Creator: abrahamc2 [UID: 432]		Claim Equivalent
Modified line 211		
Mutant 5823 Creator: gregoryg [UID: 426]		Claim Equivalent
Modified line 178		
Mutant 5886 Creator: gregoryg [UID: 426]		Claim Equivalent
Modified line 190		

JUnit tests

Test 4034 | Creator: alessiogambi [UID: 433]

```
1 /* no package name */
2
3 import org.junit.*;
4 import static org.junit.Assert.*;
5
6 public class TestSparseIntArray {
7     @Test(timeout = 4000)
8     public void test() throws Throwable {
9         SparseIntArray sia = new SparseIntArray(0);
10         sia.put(0, 7);
11         sia.put(1, 8);
12         sia.delete(0);
13         assertEquals(1, sia.size());
14         assertEquals(8, sia.get(1));
15     }
16 }
```



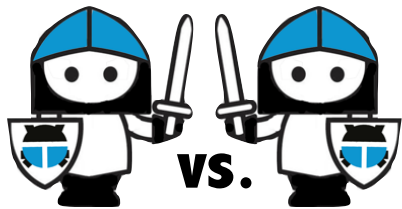
Code Defenders



vs.

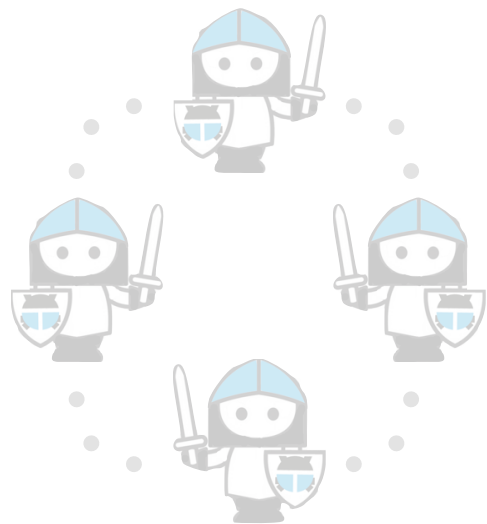
Single-player

EVASUITE / MAJOR



vs.

Duels



Multi-player

Code Defenders - Game 2260

code-defenders.org/multiplayer/play?id=2260

Code Defenders

games upload class leaderboard help netwalkinshaw

DEFENDER::ACTIVE SparseIntArray Show Scoreboard

Class Under Test Write a new JUnit test here Defend!

```
35 /**
36  * Creates a new SparseIntArray containing no mappings that will not
37  * require any additional memory allocation to store the specified
38  * number of mappings. If you supply an initial capacity of 0, the
39  * sparse array will be initialized with a light-weight representation
40  * not requiring any additional array allocations.
41  */
42 public SparseIntArray(int initialCapacity) {
43     if (initialCapacity == 0) {
44         mKeys = SparseIntArray.EMPTY_INT_ARRAY;
45         mValues = SparseIntArray.EMPTY_INT_ARRAY;
46     } else {
47         mKeys = new int[initialCapacity];
48         mValues = new int[mKeys.length];
49     }
50     mSize = 0;
51 }
52
53 /**
54  * Given the current size of an array, returns an ideal size to which the array should grow
55  * This is typically double the given size, but should not be relied upon to do so in the
56  * future.
57  */
58 public static int growSize(int currentSize) {
59     return currentSize <= 4 ? 8 : currentSize + (currentSize >> 1);
60 }
61 }
```

```
1 /* no package name */
2
3 import org.junit.*;
4 import static org.junit.Assert.*;
5
6 public class TestSparseIntArray {
7     @Test(timeout = 4000)
8     public void test() throws Throwable {
9         // test here!
10     }
11 }
12 }
```

Existing Mutants

alive (35)	killed (68)	equivalent (5)
Search...		
Mutant 5838 Creator: amin [UID: 428]		Claim Equivalent
Modified line 244		
Mutant 5855 Creator: abrahamc2 [UID: 432]		Claim Equivalent
Modified line 211		
Mutant 5823 Creator: gregoryg [UID: 426]		Claim Equivalent
Modified line 178		
Mutant 5886 Creator: gregoryg [UID: 426]		Claim Equivalent
Modified line 190		

JUnit tests

Test 4034 | Creator: alessiogambi [UID: 433]

```
1 /* no package name */
2
3 import org.junit.*;
4 import static org.junit.Assert.*;
5
6 public class TestSparseIntArray {
7     @Test(timeout = 4000)
8     public void test() throws Throwable {
9         SparseIntArray sia = new SparseIntArray(0);
10         sia.put(0, 7);
11         sia.put(1, 8);
12         sia.delete(0);
13         assertEquals(1, sia.size());
14         assertEquals(8, sia.get(1));
15     }
16 }
```



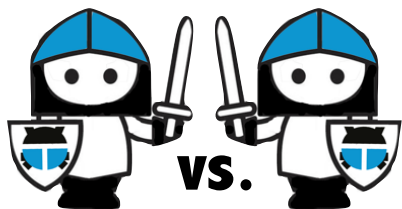
Code Defenders



Single-player

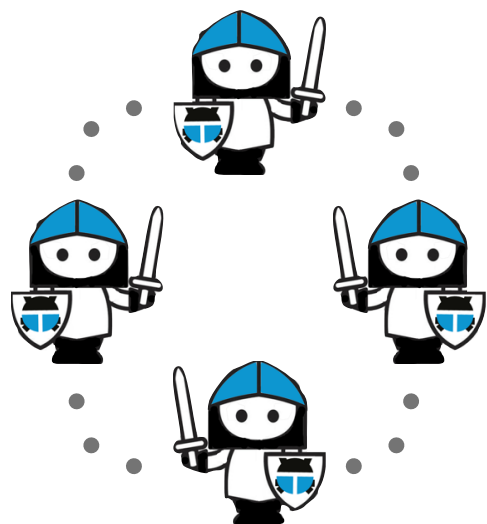
vs.

EVASUITE / MAJOR

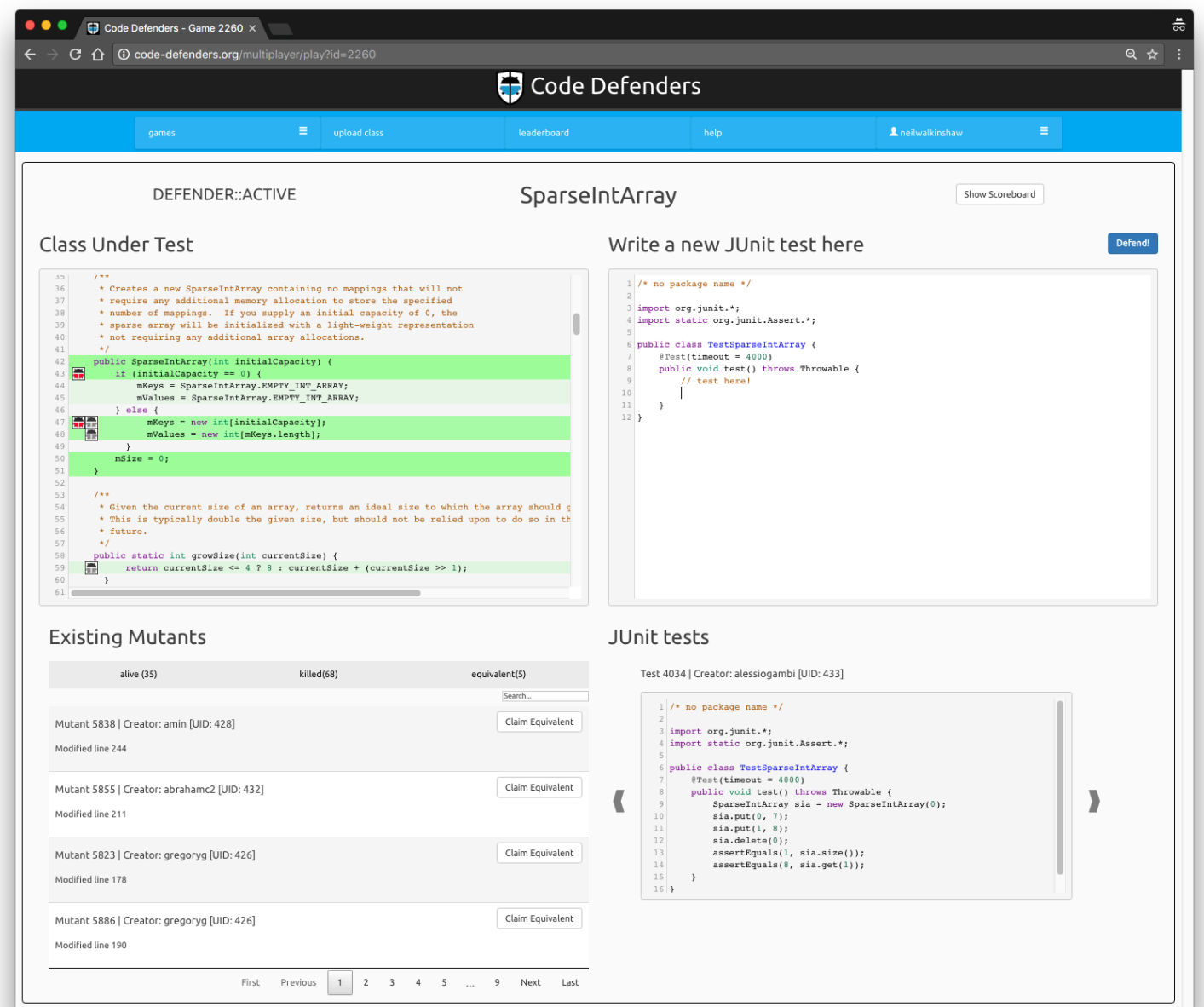


Duels

vs.



Multi-player





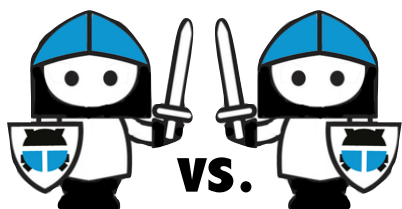
Code Defenders



Single-player

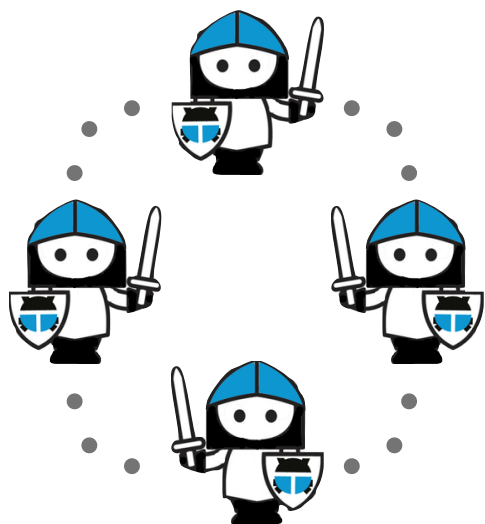
vs.

EVASUITE / MAJOR



Duels

vs.



Multi-player

DEFENDER::ACTIVE

Spa

Class Under Test

```
35  /**
36   * Creates a new SparseIntArray containing no mappings that will not
37   * require any additional memory allocation to store the specified
38   * number of mappings. If you supply an initial capacity of 0, the
39   * sparse array will be initialized with a light-weight representation
40   * not requiring any additional array allocations.
41   */
42  public SparseIntArray(int initialCapacity) {
43      if (initialCapacity == 0) {
44          mKeys = SparseIntArray.EMPTY_INT_ARRAY;
45          mValues = SparseIntArray.EMPTY_INT_ARRAY;
46      } else {
47          mKeys = new int[initialCapacity];
48          mValues = new int[mKeys.length];
49      }
50      mSize = 0;
51  }
52
53  /**
54   * Given the current size of an array, returns an ideal size to which the array should
55   * This is typically double the given size, but should not be relied upon to do so in
56   * future.
57   */
58  public static int growSize(int currentSize) {
59      return currentSize <= 4 ? 8 : currentSize + (currentSize >> 1);
60  }
61  }
```

Existing Mutants

alive (35)

killed(68)

equivalent(5)

Search...

Mutant 5838 | Creator: amin [UID: 428]

Claim Equivalent



Early Adopters



Early Adopters



Paul Amman



Early Adopters

Paul Amman



Jeff Offutt



Early Adopters

Paul Amman



Jeff Offutt



Early Adopters



Paul Amman



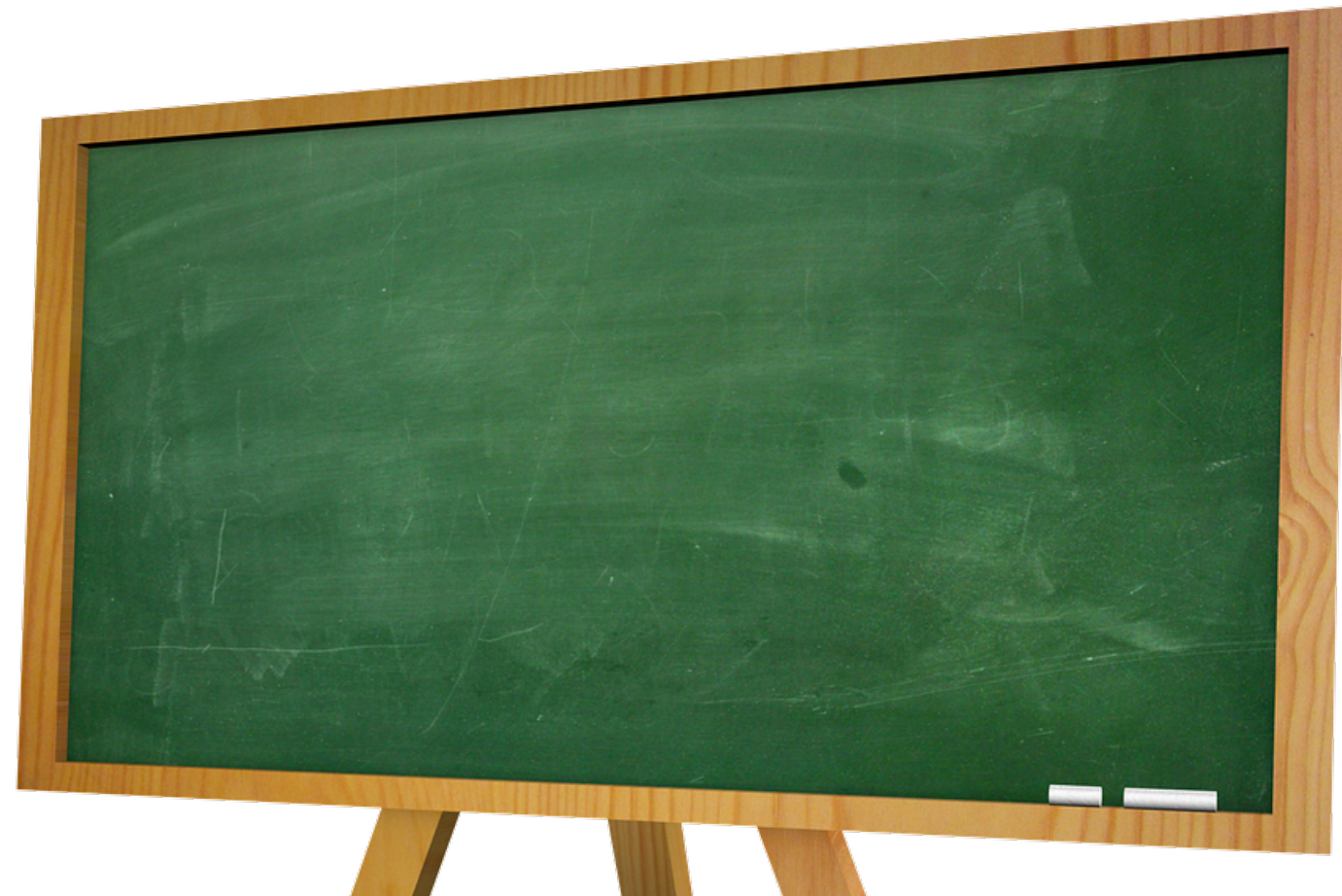
Jeff Offutt



René Just (UMass)



Teaching Software Testing Concepts with Code Defenders





Encode Software Testing Concepts as Puzzles



Encode Software Testing Concepts as Puzzles





Encode Software Testing Concepts as Puzzles





Statement Coverage

Defending puzzle



Statement Coverage

Defending puzzle





Statement Coverage

Defending puzzle

Given this **code under test**

```
int m(int x) {  
    if (int x == 5)  
        return 3;  
    return (x*8);  
}
```



Statement Coverage

Defending puzzle



Given this **code under test**

```
int m(int x) {  
    if (int x == 5)  
        return 3;  
    return (x*8);  
}
```



Statement Coverage

Defending puzzle

...and this **surviving mutant**

Given this **code under**

```
int m(int x) {  
    if (int x == 5)  
        return 3;  
    return (x*8);  
}
```

```
int m(int x) {  
    if (int x == 5)  
        return 3;  
    return (x*3+7);  
}
```



Statement Coverage

Defending puzzle



Given this **code under test**

```
int m(int x) {  
    if (int x == 5)  
        return 3;  
    return (x*8);  
}
```

...and this **surviving mutant**

```
int m(int x) {  
    if (int x == 5)  
        return 3;  
    return (x*3+7);  
}
```



Statement Coverage

Defending puzzle

and this **test suite**

Given this **code** un

```
int m(int x)
  if (int x =
    return 3;
  return (x*8
}
```

```
@Test
void test() {
  int x = 5;
  assertEquals(3, m(x));
}

@Test
void test() {
}
```




Statement Coverage

Defending puzzle



Given this **code under test**

```
int m(int x) {  
    if (int x == 5)  
        return 3;  
    return (x*8);  
}
```

...and this **surviving mutant**

```
int m(int x) {  
    if (int x == 5)  
        return 3;  
    return (x*3+7);  
}
```

and this **test suite**

```
@Test  
void test() {  
    int x = 5;  
    assertEquals(3,m(x));  
}  
@Test  
void test() {  
  
}
```



Statement Coverage

Defending puzzle

create a **new test** that **detects** the mutant



and this **test suite**

Given this **code under test**

```
int m(int x) {  
    if (int x == 5)  
        return 3;  
    return (x*8);  
}
```

...and this **surviving mutant**

```
int m(int x) {  
    if (int x == 5)  
        return 3;  
    return (x*3+7);  
}
```

```
@Test  
void test() {  
    int x = 5;  
    assertEquals(3,m(x));  
}  
@Test  
void test() {  
  
}
```



Statement Coverage

Defending puzzle

create a **new test** that **detects** the mutant

and this **test suite**

Given this **code under test**

```
int m(int x) {  
    if (int x == 5)  
        return 3;  
    return (x*8);  
}
```

...and this **surviving mutant**

```
int m(int x) {  
    if (int x == 5)  
        return 3;  
    return (x*3+7);  
}
```

```
@Test  
void test() {  
    int x = 5;  
    assertEquals(3,m(x));  
}  
@Test  
void test() {
```





Statement Coverage

Defending puzzle

create a **new test** that **detects** the mutant

and this **test suite**

Given this **code under test**

```
int m(int x) {  
    if (int x == 5)  
        return 3;  
    return (x*8);  
}
```

...and this **surviving mutant**

```
int m(int x) {  
    if (int x == 5)  
        return 3;  
    return (x*3+7);  
}
```

```
@Test  
void test() {  
    int x = 5;  
    assertEquals(3,m(x));  
}  
@Test  
void test() {  
    int x = 3;  
    assertEquals(24,m(x));  
}
```





Branch Coverage

Attacking puzzle



Branch Coverage

Attacking puzzle





Branch Coverage

Attacking puzzle

Given this **code under test**

```
int m(int x) {  
    int y = 1;  
    if (x == 5) {  
        y = 3;  
    }  
    return (y * 8);  
}
```



Branch Coverage

Attacking puzzle



Given this **code under test**

```
int m(int x) {  
    int y = 1;  
    if (x == 5) {  
        y = 3;  
    }  
    return (y * 8);  
}
```



Branch Coverage

Attacking puzzle

...and this **passing test**

Given this code

```
int m(int  
    int y =  
    if (x =  
        y = 3  
    }  
    return  
}
```

```
@Test  
public void test() {  
    int x = 5;  
    assertEquals(24, m(x));  
}
```



Branch Coverage

Attacking puzzle



Given this **code under test**

```
int m(int x) {  
    int y = 1;  
    if (x == 5) {  
        y = 3;  
    }  
    return (y * 8);  
}
```

...and this **passing test**

```
@Test  
public void test() {  
    int x = 5;  
    assertEquals(24, m(x));  
}
```



Branch Coverage

Attacking puzzle

...create a surviving mutant.

Given this code under test

```
int m(int x) {  
    int y = 1;  
    if (x == 5) {  
        y = 3;  
    }  
    return (y * 8)  
}
```

```
int m(int x) {  
    int y = 1;  
    if (x == 5) {  
        y = 3;  
    }  
    return (y * 8);  
}
```



Branch Coverage

Attacking puzzle



Given this **code under test**

```
int m(int x) {  
    int y = 1;  
    if (x == 5) {  
        y = 3;  
    }  
    return (y * 8);  
}
```

...and this **passing test**

```
@Test  
public void test() {  
    int x = 5;  
    assertEquals(24, m(x));  
}
```

...create a surviving mutant.

```
int m(int x) {  
    int y = 1;  
    if (x == 5) {  
        y = 3;  
    }  
    return (y * 8);  
}
```



Branch Coverage

Attacking puzzle

Given this **code under test**

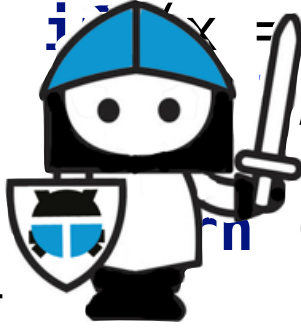
```
int m(int x) {  
    int y = 1;  
    if (x == 5) {  
        y = 3;  
    }  
    return (y * 8);  
}
```

...and this **passing test**

```
@Test  
public void test() {  
    int x = 5;  
    assertEquals(24, m(x));  
}
```

...create a surviving mutant.

```
int m(int x) {  
    int y = 1;  
    if (x == 5) {  
        y = 3;  
    }  
    return (y * 8);  
}
```





Branch Coverage

Attacking puzzle

Given this **code under test**

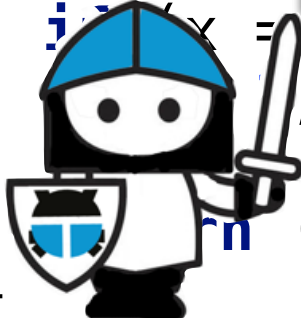
```
int m(int x) {  
    int y = 1;  
    if (x == 5) {  
        y = 3;  
    }  
    return (y * 8);  
}
```

...and this **passing test**

```
@Test  
public void test() {  
    int x = 5;  
    assertEquals(24, m(x));  
}
```

...create a surviving mutant.

```
int m(int x) {  
    int y = 0;  
    if (x == 5) {  
        y = 3;  
    }  
    return (y * 8);  
}
```





Branch Coverage

Attacking puzzle



Given this **code under test**

```
int m(int x) {  
    int y = 1;  
    if (x == 5) {  
        y = 3;  
    }  
    return (y * 8);  
}
```

...and this **passing test**

```
@Test  
public void test() {  
    int x = 5;  
    assertEquals(24, m(x));  
}
```

...create a surviving mutant.

```
int m(int x) {  
    int y = 0;  
    if (x == 5) {  
        y = 3;  
    }  
    return (y * 8);  
}
```



Other Testing Concepts



Other Testing Concepts

Testing loops



Other Testing Concepts

Testing loops

Data-flow testing



Other Testing Concepts

Testing loops

Data-flow testing

Boundary value analysis



Other Testing Concepts

Testing loops

Data-flow testing

Boundary value analysis

Model-based testing



Other Testing Concepts

Testing loops

Data-flow testing

Boundary value analysis

Model-based testing

Combinatorial testing



Other Testing Concepts

Testing loops

Data-flow testing

Boundary value analysis

Model-based testing

Combinatorial testing

Mutation testing



Related Work





Related Work



When Thursday 14:00-15:30

Session Testing II

Room Libertador C



Related Work



When Thursday 14:00-15:30

Session Testing II

Room Libertador C

**SPOILER
ALERT**

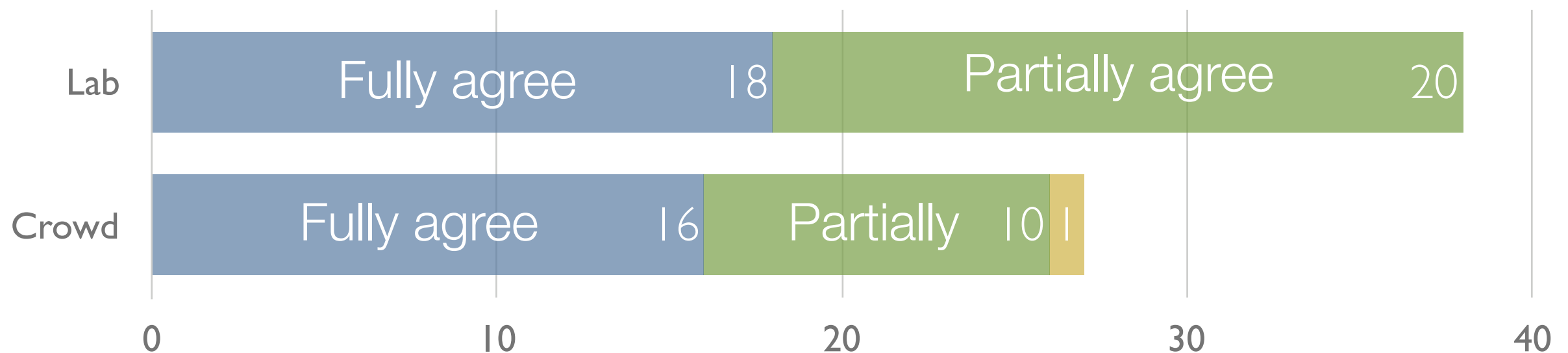


Code Defenders is Fun



Code Defenders is Fun

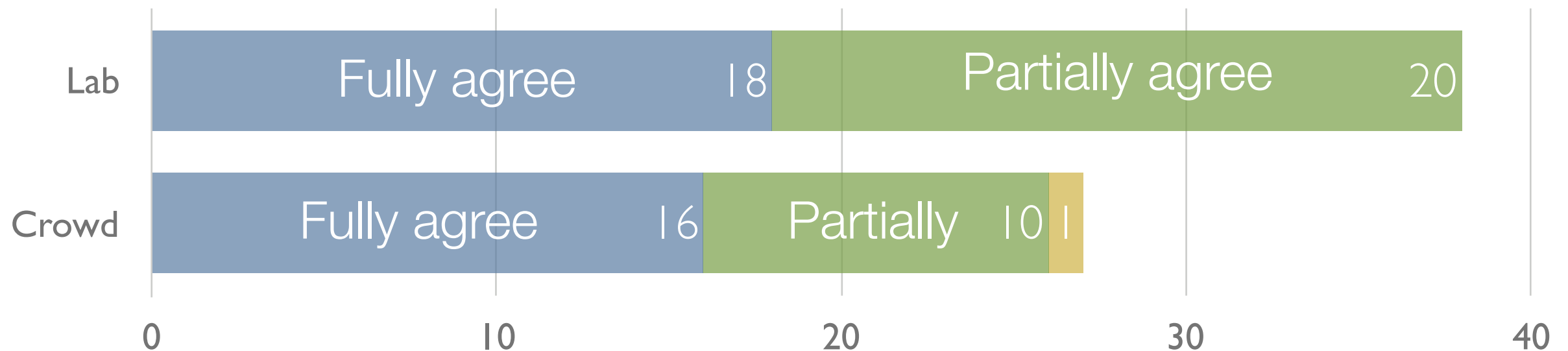
I enjoyed playing Code Defenders



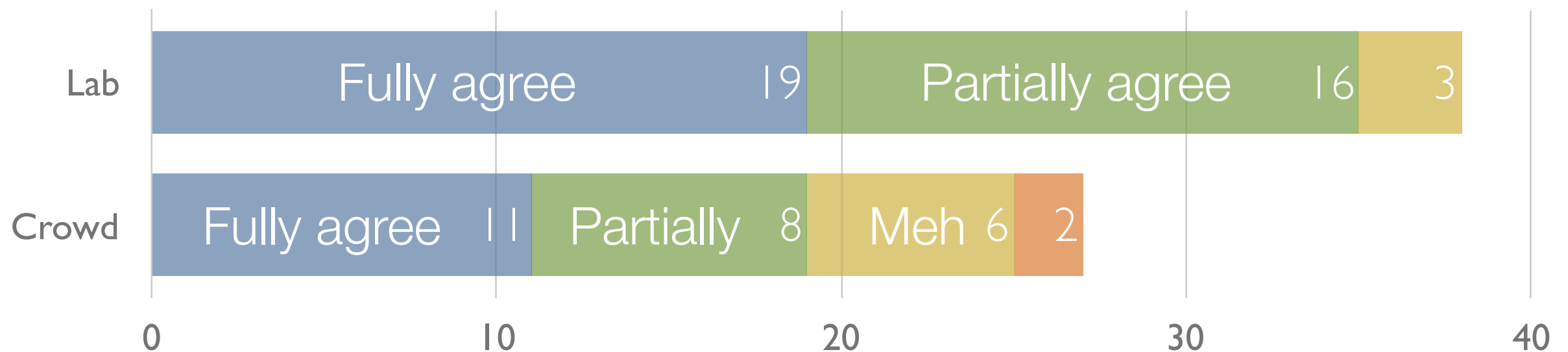


Code Defenders is Fun

I enjoyed playing Code Defenders

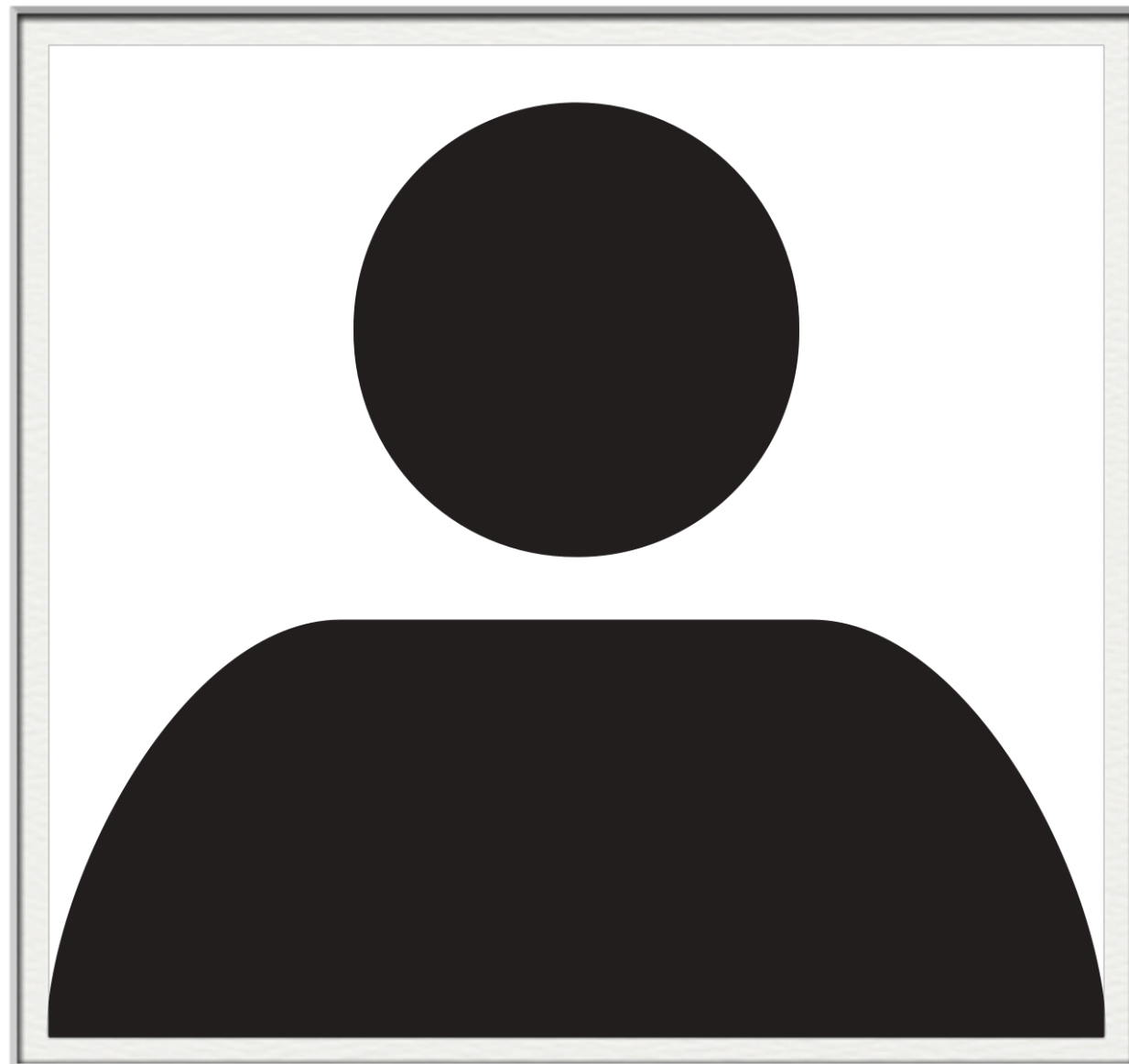


Playing Code Defenders is more fun than manual testing





Early Adopters



You?



Code Defenders - Game 2260 x

code-defenders.org/multiplayer/play?id=2260

Code Defenders

gamesupload classleaderboardhelpneilwalkinshaw

DEFENDER::ACTIVE

SparseIntArray

Show Scoreboard

Class Under Test

```
35  /**
36   * Creates a new SparseIntArray containing no mappings that will not
37   * require any additional memory allocation to store the specified
38   * number of mappings. If you supply an initial capacity of 0, the
39   * sparse array will be initialized with a light-weight representation
40   * not requiring any additional array allocations.
41   */
42   public SparseIntArray(int initialCapacity) {
43       if (initialCapacity == 0) {
44           mKeys = SparseIntArray.EMPTY_INT_ARRAY;
45           mValues = SparseIntArray.EMPTY_INT_ARRAY;
46       } else {
47           mKeys = new int[initialCapacity];
48           mValues = new int[mKeys.length];
49       }
50       mSize = 0;
51   }
52
53   /**
54   * Given the current size of an array, returns an ideal size to which the array should grow
55   * This is typically double the given size, but should not be relied upon to do so in the
56   * future.
57   */
58   public static int growSize(int currentSize) {
59       return currentSize <= 4 ? 8 : currentSize + (currentSize >> 1);
60   }
61
```

Write a new JUnit test here

Defend!

```
1  /* no package name */
2
3  import org.junit.*;
4  import static org.junit.Assert.*;
5
6  public class TestSparseIntArray {
7      @Test(timeout = 4000)
8      public void test() throws Throwable {
9          // test here!
10         |
11         |
12     }
```

Existing Mutants

alive (35)	killed(68)	equivalent(5)
Search...		
Mutant 5838 Creator: amin [UID: 428]		
Modified line 244		Claim Equivalent
Mutant 5855 Creator: abrahamc2 [UID: 432]		
Modified line 211		Claim Equivalent
Mutant 5823 Creator: gregoryg [UID: 426]		
Modified line 178		Claim Equivalent
Mutant 5886 Creator: gregoryg [UID: 426]		
Modified line 190		Claim Equivalent

First Previous 1 2 3 4 5 ... 9 Next Last

JUnit tests

Test 4034 | Creator: alessiogambi [UID: 433]

```
1  /* no package name */
2
3  import org.junit.*;
4  import static org.junit.Assert.*;
5
6  public class TestSparseIntArray {
7      @Test(timeout = 4000)
8      public void test() throws Throwable {
9          SparseIntArray sia = new SparseIntArray(0);
10         sia.put(0, 7);
11         sia.put(1, 8);
12         sia.delete(0);
13         assertEquals(1, sia.size());
14         assertEquals(8, sia.get(1));
15     }
16 }
```

code-defenders.org