



JavaONE TUT3970

CRUSH CANDY
WITH DUKESCRIPT

ANTON EPPLÉ

- ▶ @monacotoni
- ▶ www.dukescript.com
- ▶ Dukehoff GmbH



SAFE HARBOR STATEMENT

SAFE HARBOR STATEMENT

The following is intended for information purposes only. I can not be held responsible for the overuse of effects and animations in this presentation. If any person in this room has a medical condition that is triggered by fast moving objects on the screen and/or explosions, he/she should probably better leave now...

(I got carried away by the topic.)

AGENDA

AGENDA

★ MATCH3 GAMES

AGENDA

★ MATCH3 GAMES

★ DUKESCRIPT

AGENDA

- ★ MATCH3 GAMES
- ★ DUKESCRIPT
- ★ GAME LOGIC & IMPLEMENTATION

AGENDA

- ★ MATCH3 GAMES
- ★ DUKESCRIPT
- ★ GAME LOGIC & IMPLEMENTATION
- ★ RESPONSIVE LAYOUT & DESIGN

AGENDA

- ★ **MATCH3 GAMES**
- ★ **DUKESCRIPT**
- ★ **GAME LOGIC & IMPLEMENTATION**
- ★ **RESPONSIVE LAYOUT & DESIGN**
- ★ **DIRTY HACKS :-)**

AGENDA

- ★ MATCH3 GAMES
- ★ DUKESCRIPT
- ★ GAME LOGIC & IMPLEMENTATION
- ★ RESPONSIVE LAYOUT & DESIGN
- ★ DIRTY HACKS :-)
- ★ APP STORES



"Ask Questions"



MATCH3



MATCH3



MOVES LEFT

15

0



MOVES LEFT

15

0









FARM Heroes Saga



\$\$\$

KING

<http://venturebeat.com/2015/05/14/candy-crush-saga-maker-king-beats-wall-streets-expectations/>

KING Q1 2015

<http://venturebeat.com/2015/05/14/candy-crush-saga-maker-king-beats-wall-streets-expectations/>

KING
Q1 2015
550 MIO USERS

KING
Q1 2015
550 MIO USERS
\$569 MIO

KING
Q1 2015
550 MIO USERS
\$569 MIO
CANDY CRUSH SAGA

KING

Q1 2015

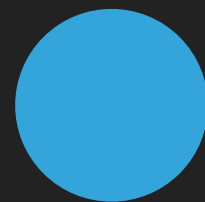
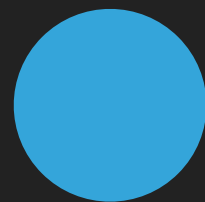
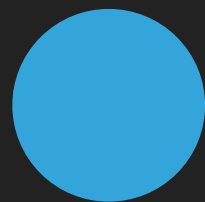
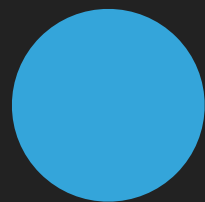
550 MIO USERS

\$569 MIO

CANDY CRUSH SAGA

\$216 MIO

DUKESCRIP

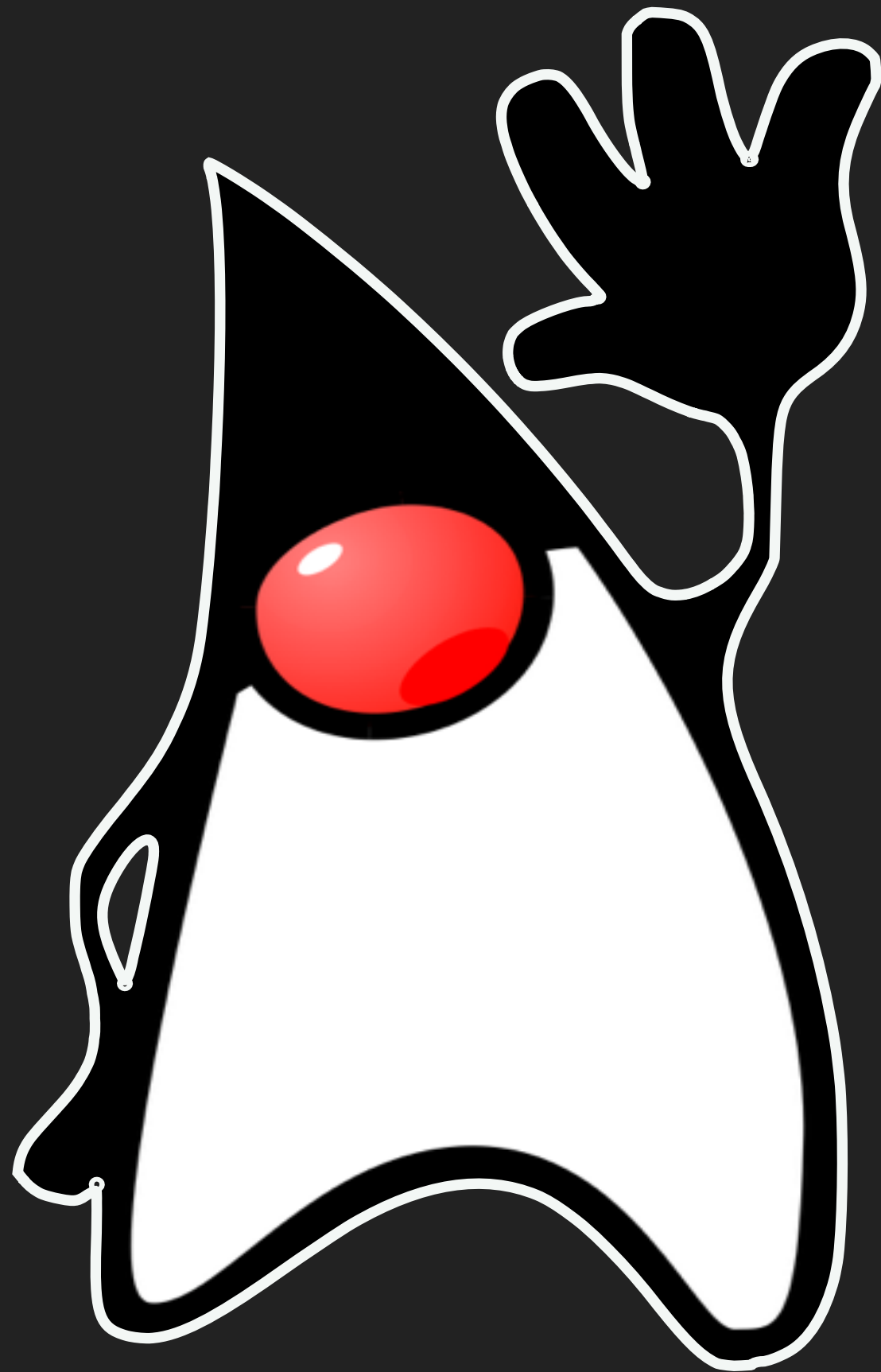




DUKESCRIPT







HTML





HOW DOES IT WORK?

Insert HTML5 Renderer here

DukeScript glues them together

Insert JVM here

HOW DOES IT WORK?

HTML5 Component renders the
result

DukeScript and it's Java APIs control the
Renderer

JVM executes Java Business Logic,
written with DukeScript Java APIs

DESKTOP

`javafx.scene.web.WebView`

DukeScript glues them together

Hotspot

ANDROID

android.webkit.WebView

DukeScript glues them together

Dalvik/ART

IOS

NSObject.UIResponder.UIView
.UIWebView

DukeScript glues them together

RoboVM

BROWSER

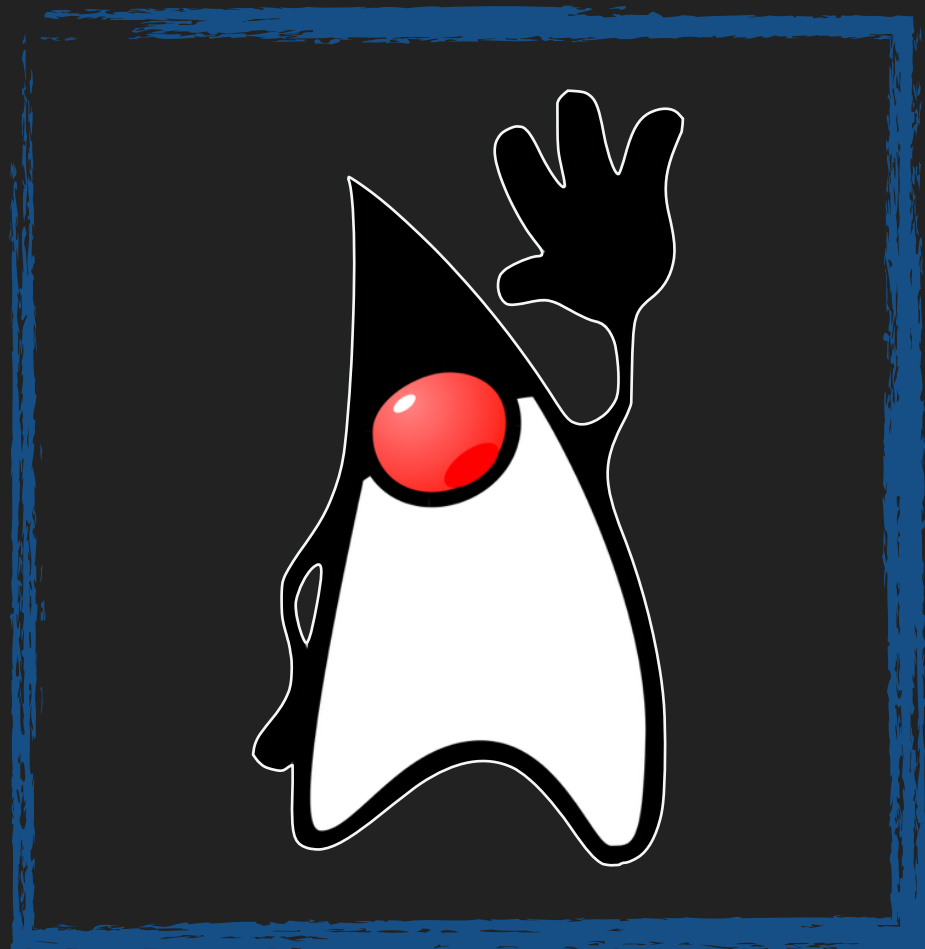
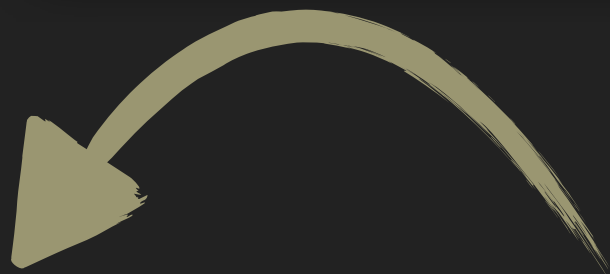
Chrome...

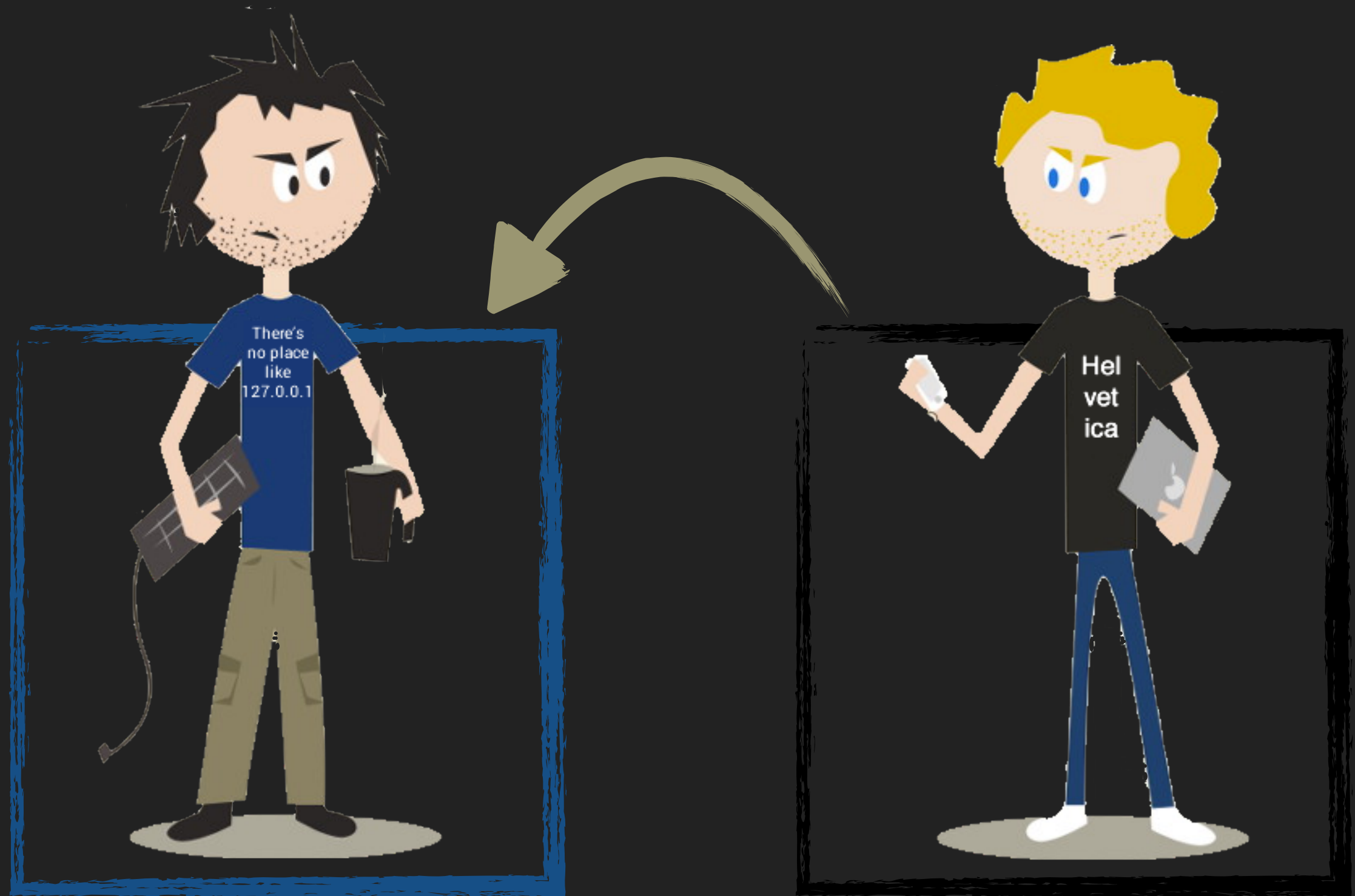
DukeScript glues them together

`bck2brwsr/teavm`



Knockout.





Infographic by: Shane Snow. Shane Snow is an entrepreneur, writer, and recent Columbia MS/Digital Media graduate. Visit his [personal site](#) and follow him on Twitter @[shanesnow](#).

GETTING STARTED

GETTING STARTED

▶ JDK 1.8

GETTING STARTED

- ▶ JDK 1.8
- ▶ NetBeans 8.1

GETTING STARTED

- ▶ JDK 1.8
- ▶ NetBeans 8.1
- ▶ DukeScript Plugin

GETTING STARTED

- ▶ JDK 1.8
- ▶ NetBeans 8.1
- ▶ DukeScript Plugin
- ▶ Or Maven and any IDE

GETTING STARTED

- ▶ JDK 1.8
- ▶ NetBeans 8.1
- ▶ DukeScript Plugin
- ▶ Or Maven and any IDE

GETTING STARTED

- ▶ JDK 1.8
- ▶ NetBeans 8.1
- ▶ DukeScript Plugin
- ▶ Or Maven and any IDE
- ▶ https://dukescript.com/getting_started.html







ACHIEVEMENT UNLOCKED!



DUKESCRIPT ADEPT



GAME LOGIC





GAME LOGIC



MOVES LEFT

15

0



MOVES LEFT

15

0



MOVES LEFT

15

row = index / level.getColumns();

0



MOVES LEFT

15

row = index / level.getColumns();

0



MOVES LEFT

15

row = 25 / level.getColumns();

0



MOVES LEFT

15

row = 25 / 8;

0



MOVES LEFT
15

0

row = 3;



MOVES LEFT

15

row = 3;

0

0

1

2

3

4

5

6

7



MOVES LEFT

15

col = index % level.getRows();

0



MOVES LEFT

15

col = index % level.getRows();

0



MOVES LEFT

15

col = 25 % level.getRows();

0



MOVES LEFT
15

0

col = 25 % 8;



MOVES LEFT
15

0

col = 1;



MOVES LEFT
15

0

col = 1;

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63
0	1	2	3	4	5	6	7

Progress bar with 10 segments, 1 segment filled, 2 stars.

MOVES LEFT

15

0



MOVES LEFT
15

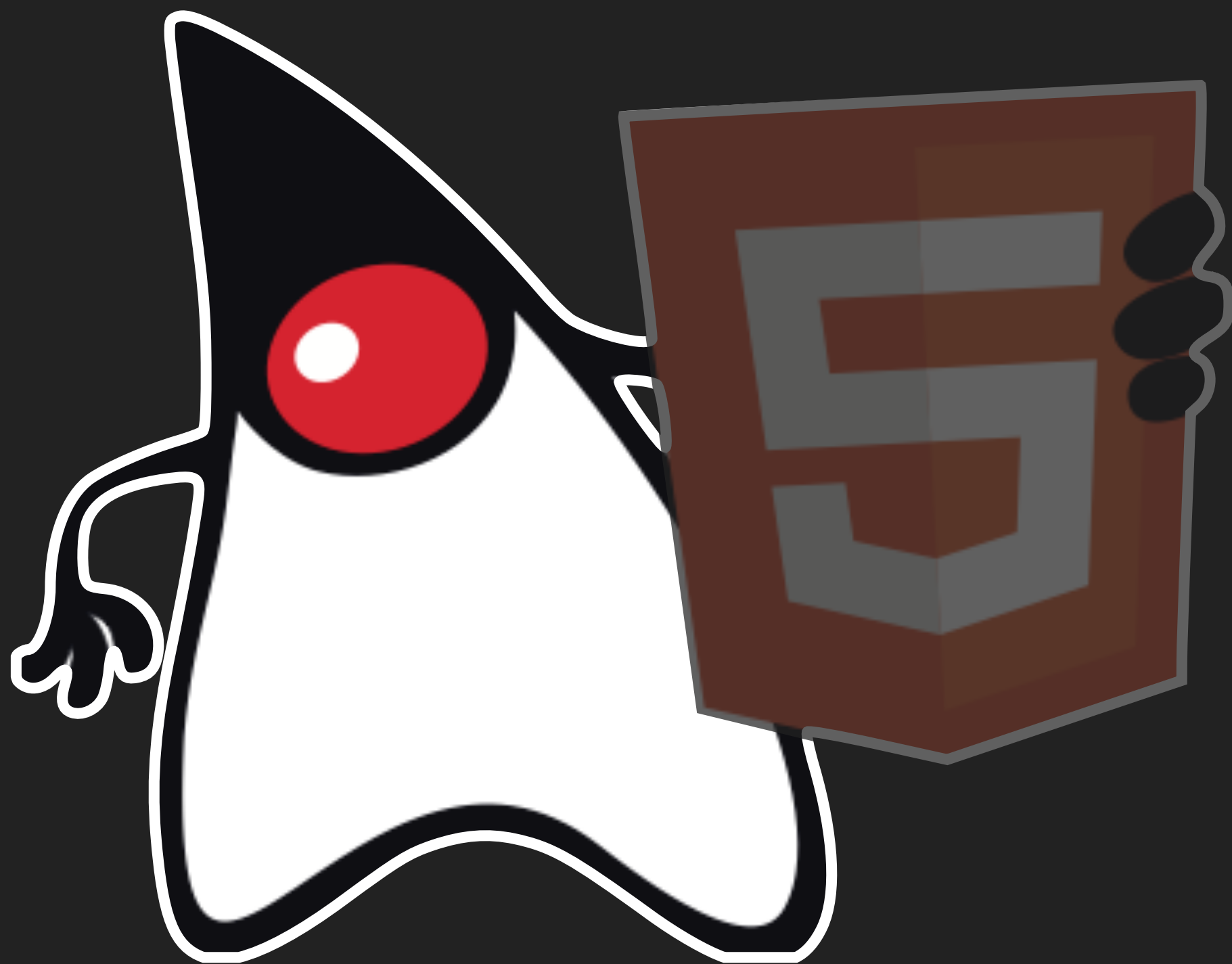


= 4

0







VIEWMODEL



VIEWMODEL

```
@Model(className = "Tile",
```



VIEWMODEL

```
@Model(className = "Tile",  
  properties = {
```



VIEWMODEL

```
@Model(className = "Tile",  
    properties = {  
        @Property(name = "type", type = int.class)
```



VIEWMODEL

```
@Model(className = "Tile",  
    properties = {  
        @Property(name = "type", type = int.class)  
    })
```



VIEWMODEL

```
@Model(className = "Tile",  
    properties = {  
        @Property(name = "type", type = int.class)  
    })  
  
class TileVM {}
```



VIEWMODEL GENERATED

```
public final class Tile implements Cloneable {  
    private static final Html4JavaType TYPE = new Html4JavaType();  
    private final org.netbeans.html.json.spi.Proto proto;  
    private int prop_type;  
    public int getType() {  
        proto.accessProperty("type");  
        return prop_type;  
    }  
    public void setType(int v) {  
        proto.verifyUnlocked();  
        Object o = prop_type;  
        if (TYPE.isSame(o , v)) return;  
        prop_type = v;  
        proto.valueHasMutated("type", o, v);  
    }  
    //...
```



VIEWMODEL NESTED



VIEWMODEL NESTED

```
@Model(className = "Level", properties = {
```



VIEWMODEL NESTED

```
@Model(className = "Level", properties = {  
    @Property(name = "columns", type = int.class),
```



VIEWMODEL NESTED

```
@Model(className = "Level", properties = {  
    @Property(name = "columns", type = int.class),  
    @Property(name = "rows", type = int.class),
```



VIEWMODEL NESTED

```
@Model(className = "Level", properties = {  
    @Property(name = "columns", type = int.class),  
    @Property(name = "rows", type = int.class),  
    @Property(name = "tiles", type = Tile.class,  
                ),
```



VIEWMODEL NESTED

```
@Model(className = "Level", properties = {  
    @Property(name = "columns", type = int.class),  
    @Property(name = "rows", type = int.class),  
    @Property(name = "tiles", type = Tile.class,  
                ),
```



VIEWMODEL NESTED

```
@Model(className = "Level", properties = {  
    @Property(name = "columns", type = int.class),  
    @Property(name = "rows", type = int.class),  
    @Property(name = "tiles", type = Tile.class, array = true ),
```



VIEWMODEL NESTED

```
@Model(className = "Level", properties = {
    @Property(name = "columns", type = int.class),
    @Property(name = "rows", type = int.class),
    @Property(name = "tiles", type = Tile.class, array = true ),
    @Property(name = "play", type = boolean.class),
    @Property(name = "moves", type = int.class),
    @Property(name = "score", type = int.class)
})
class LevelVM {
```





ACHIEVEMENT UNLOCKED!

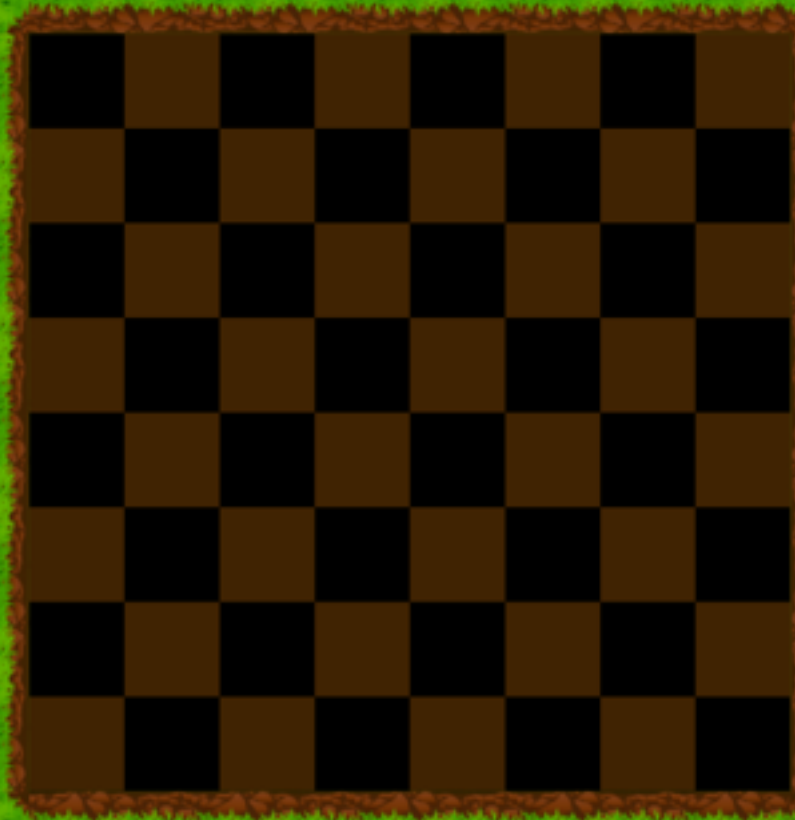


VIEWMODEL PRACTICIONER



MOVES LEFT
15

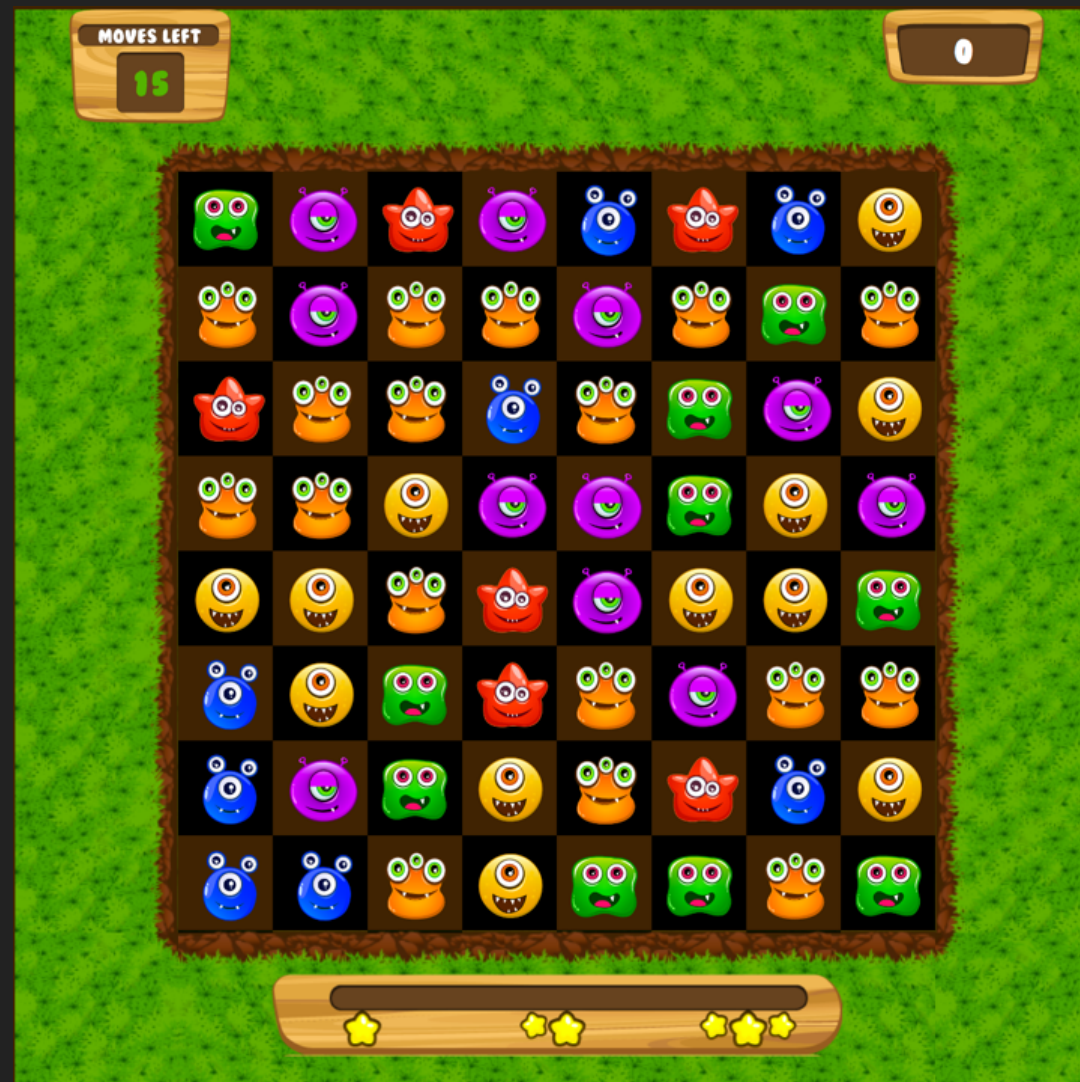
0





```
Level level = new Level(...);  
for (int i = 0; i < 64; i++) {  
    level.getTiles().set(i, new Tile(random.nextInt( 4 )));  
}
```





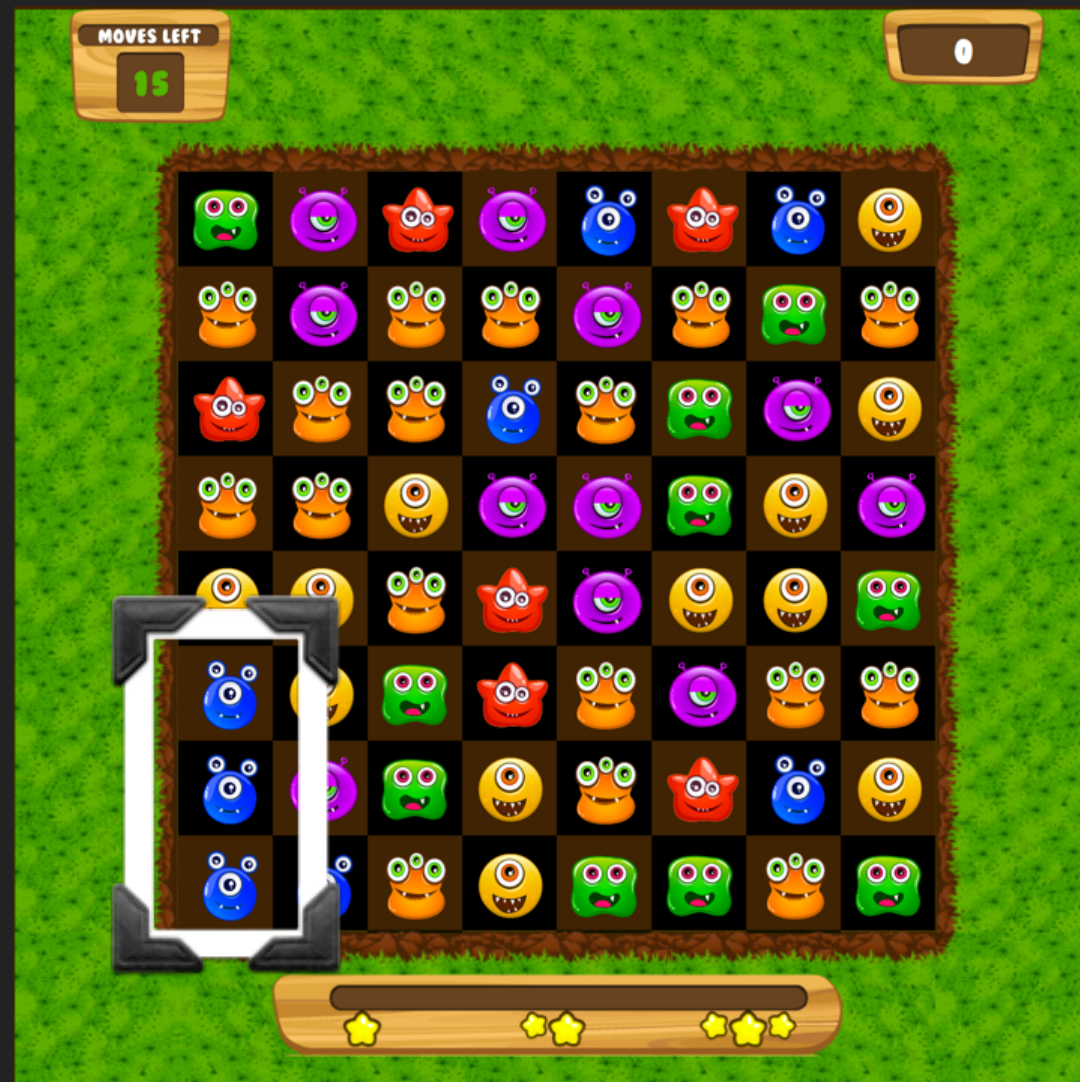
```

Level level = new Level(...);
for (int i = 0; i < 64; i++) {

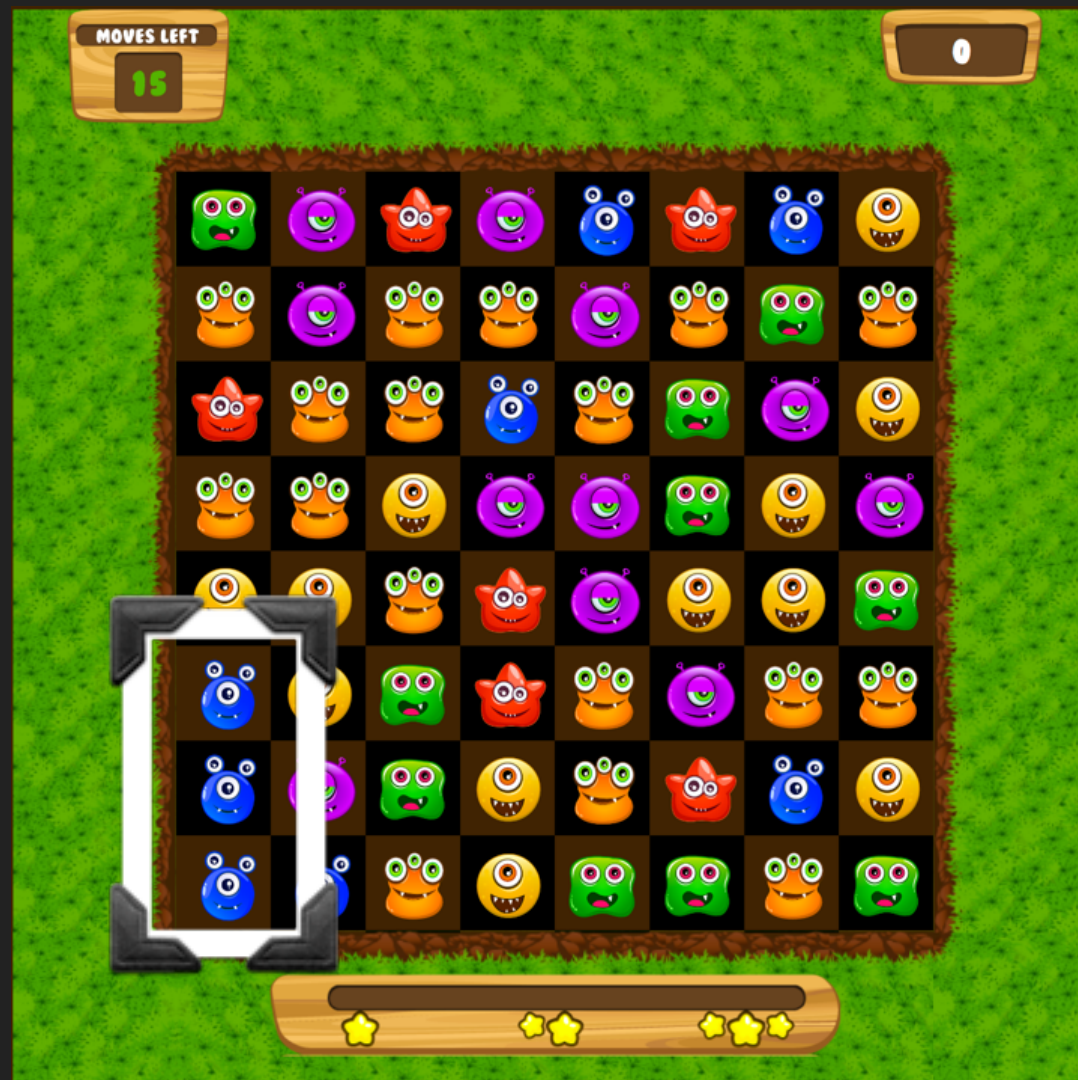
    level.getTiles().set(i, new Tile(random.nextInt( 4 )));

}

```

```
Level level = new Level(...);  
for (int i = 0; i < 64; i++) {  
    level.getTiles().set(i, new Tile(random.nextInt( 4 )));  
}
```



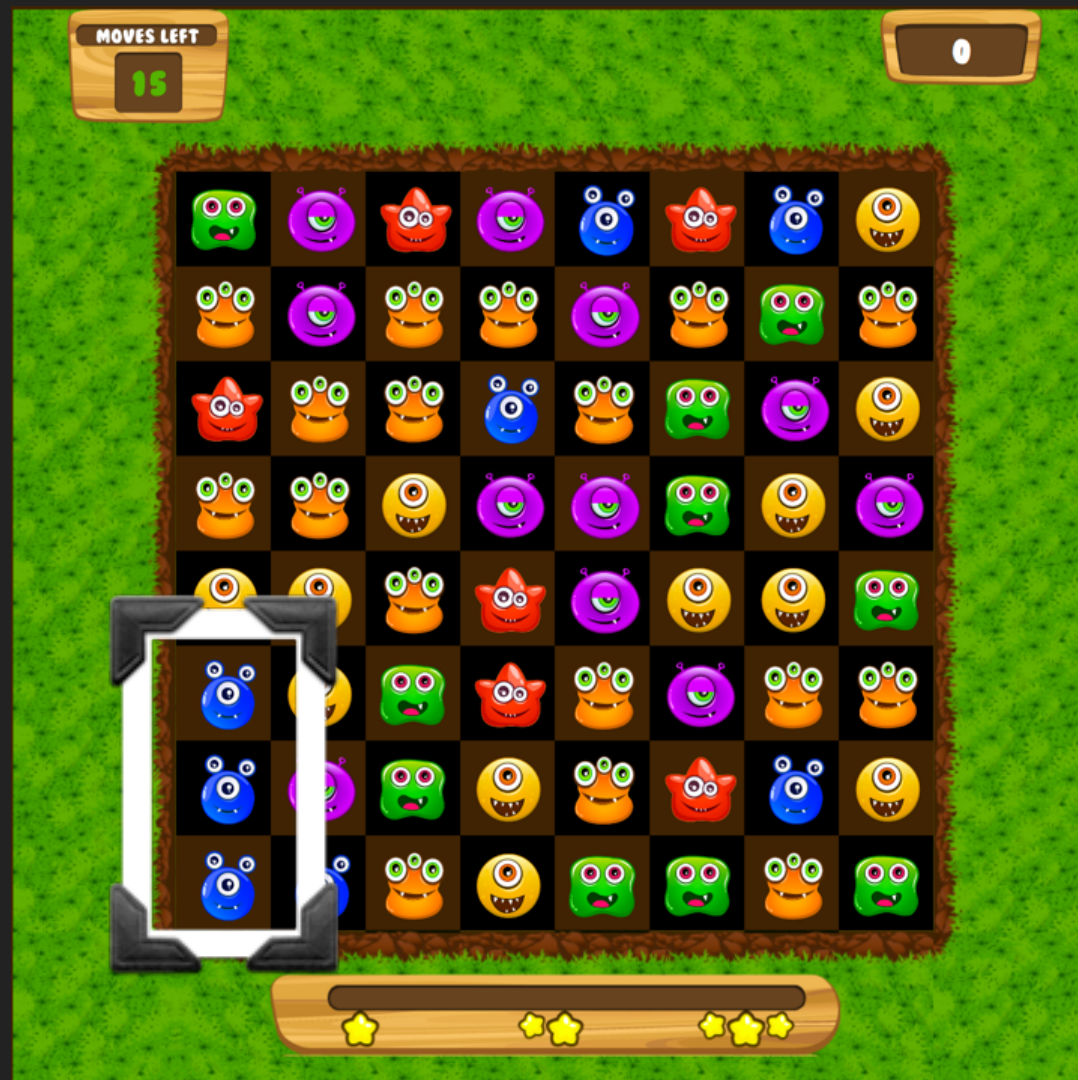
```

Level level = new Level(...);
for (int i = 0; i < 64; i++) {

    level.getTiles().set(i, new Tile(random.nextInt( 4 )));

}

```



```

Level level = new Level(...);
for (int i = 0; i < 64; i++) {
    do{

        level.getTiles().set(i, new Tile(random.nextInt( 4 )));

    }while (isHorizontalMatch(level, i)
            || isVerticalMatch(level, i));

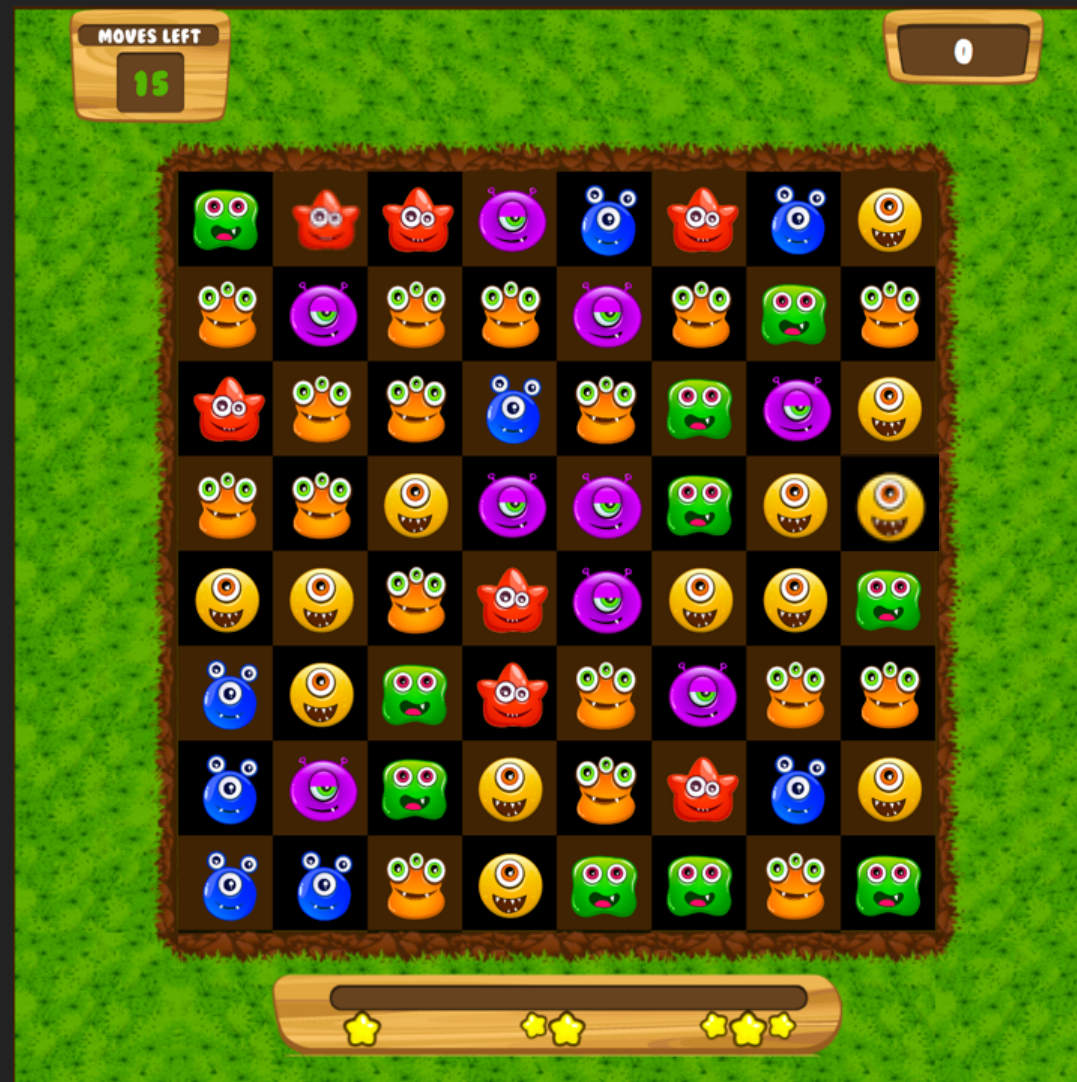
}

```

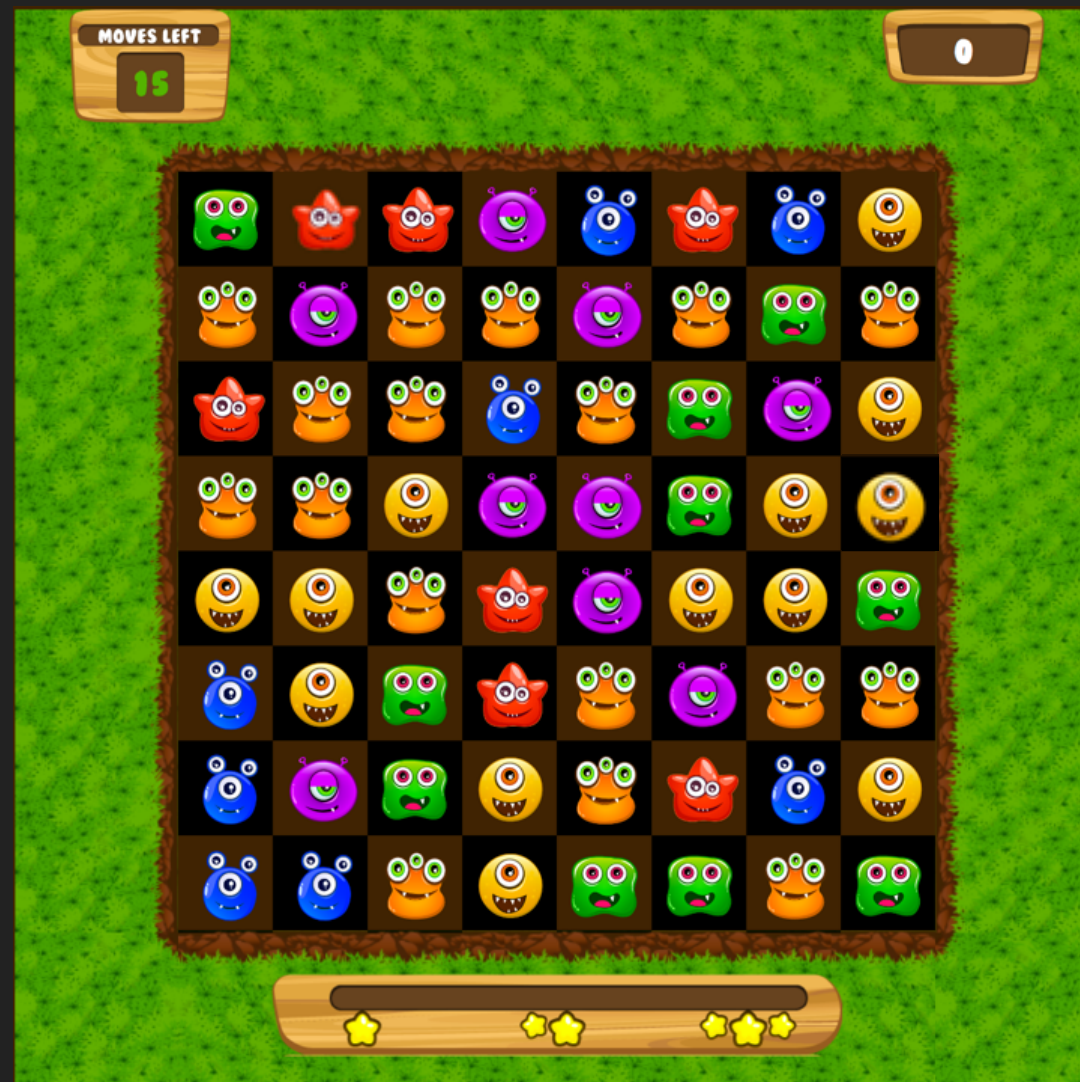

MOVES LEFT
15

0

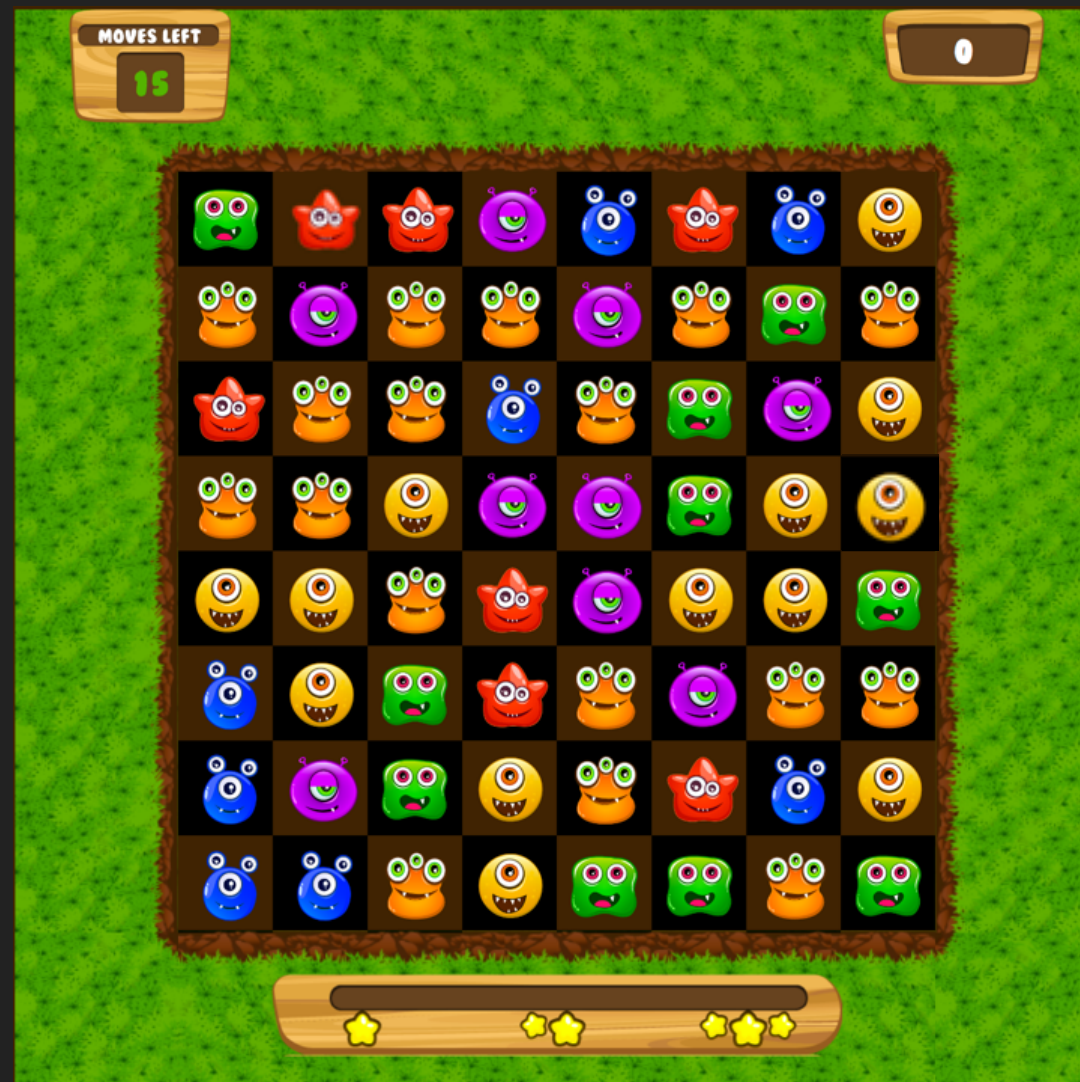




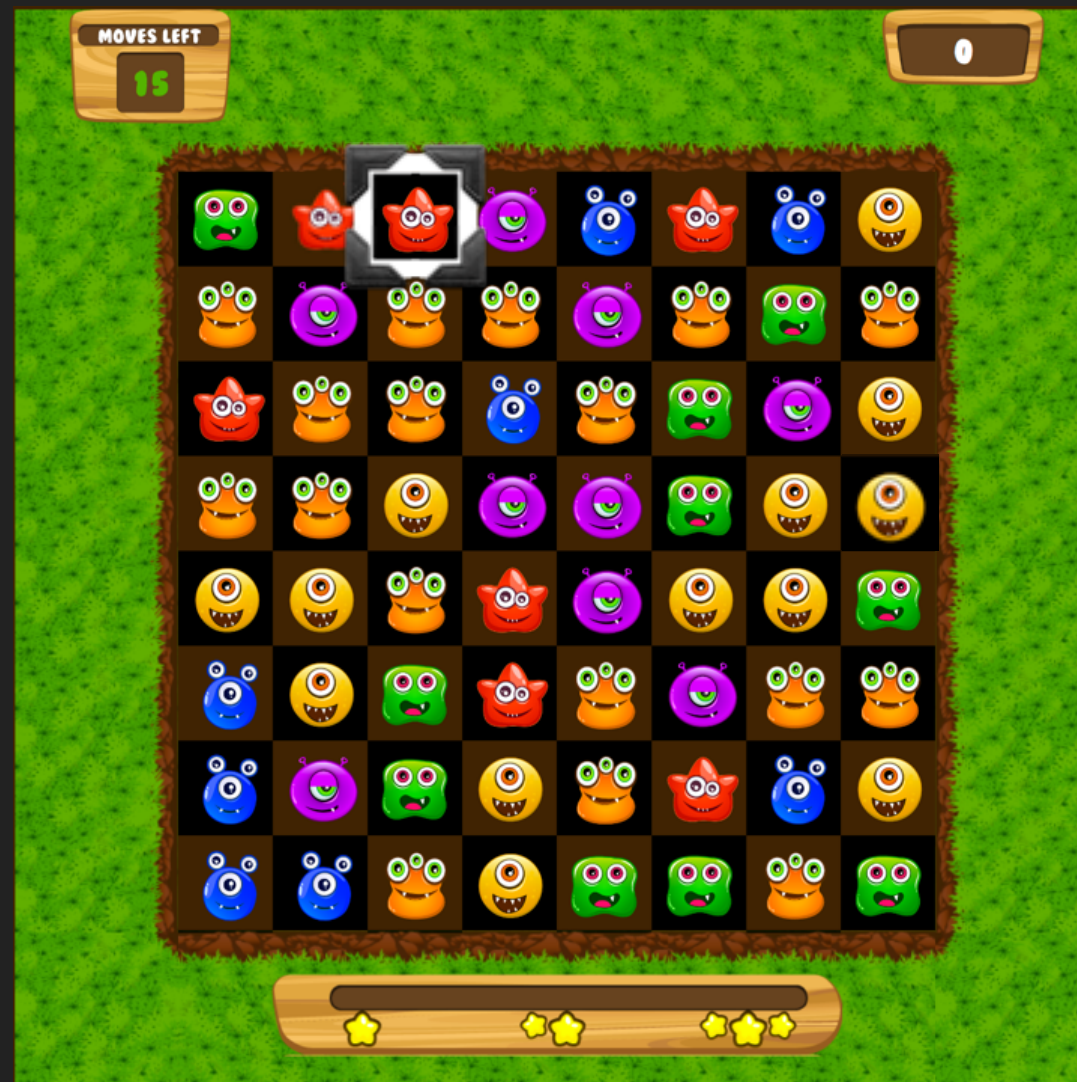
```
public static boolean isHorizontalMatch(Level level, int i) {
```



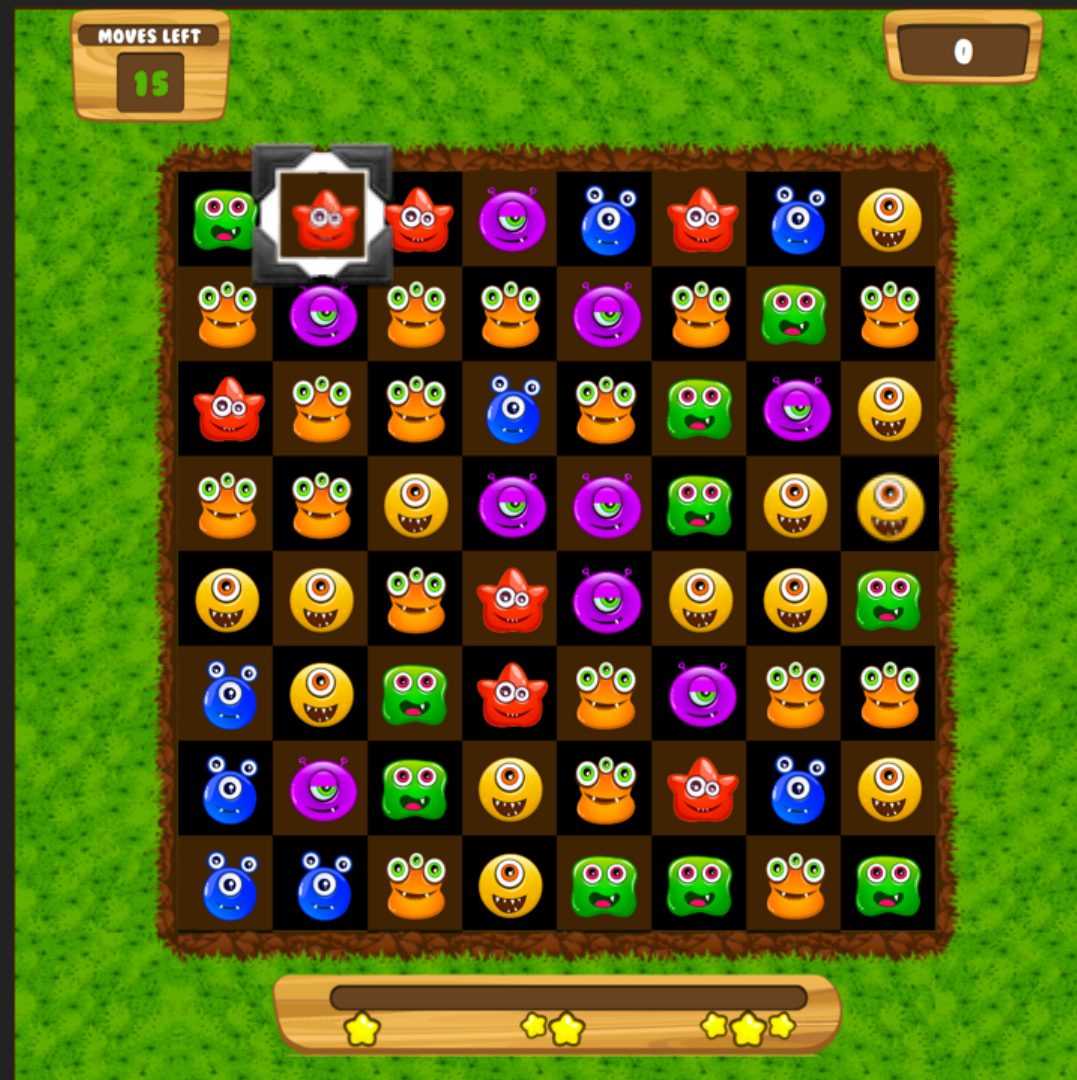
```
public static boolean isHorizontalMatch(Level level, int i) {
    List<Tile> tiles = level.getTiles();
```



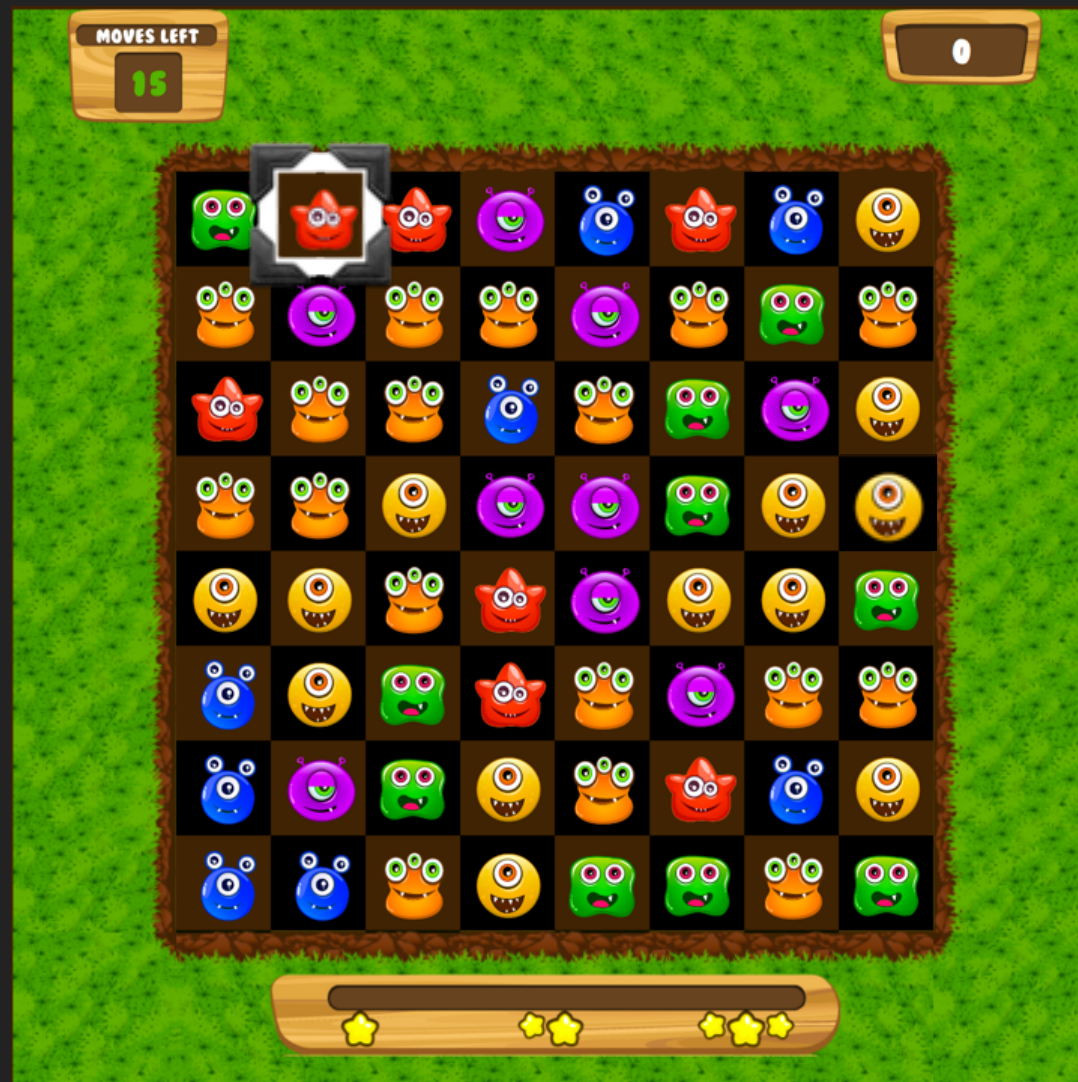
```
public static boolean isHorizontalMatch(Level level, int i) {
    List<Tile> tiles = level.getTiles();
    return tiles.get(i).getType() == tiles.get(i - 1).getType()
```

```
public static boolean isHorizontalMatch(Level level, int i) {
    List<Tile> tiles = level.getTiles();
    return tiles.get(i).getType() == tiles.get(i - 1).getType()
}
```

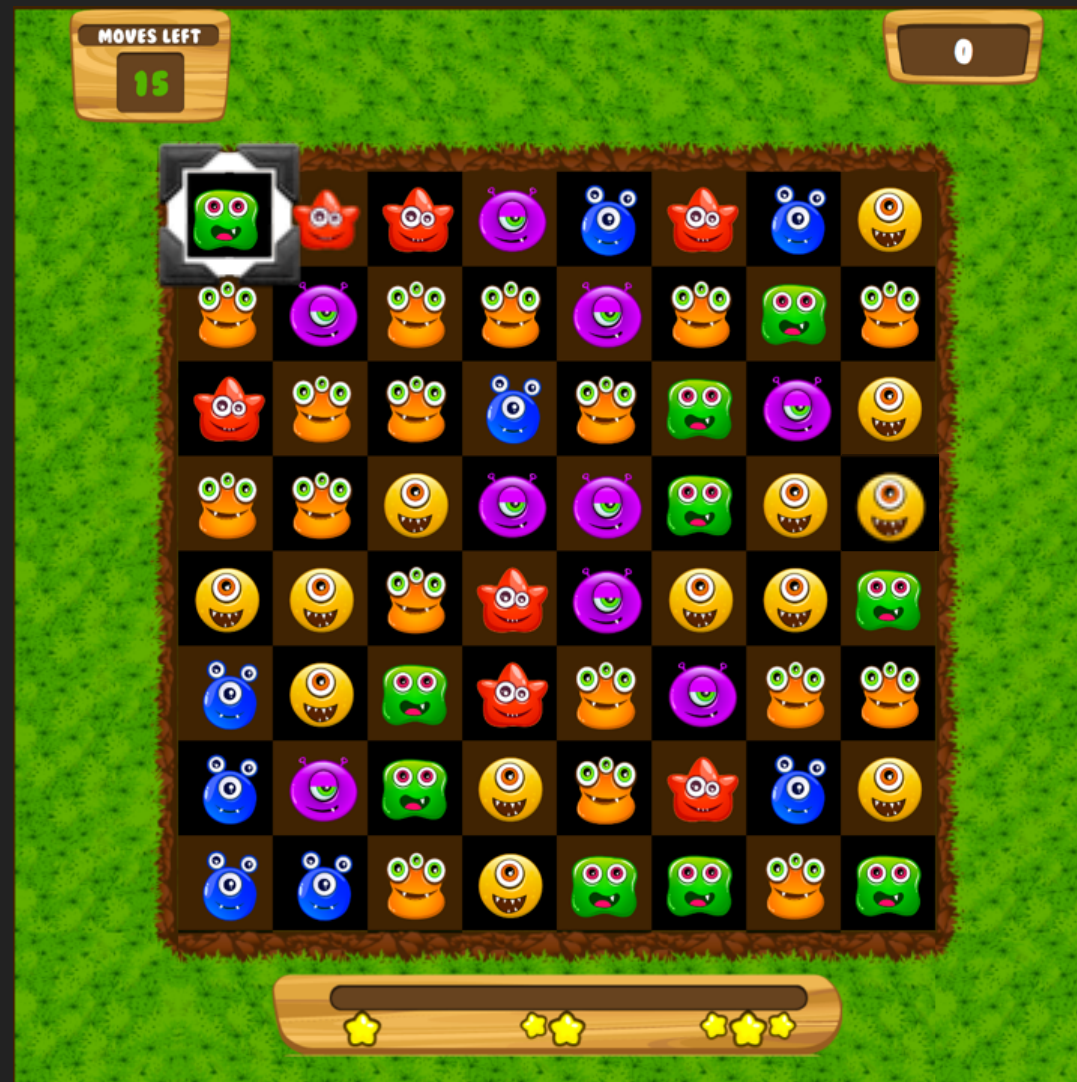



```
public static boolean isHorizontalMatch(Level level, int i) {
    List<Tile> tiles = level.getTiles();
    return tiles.get(i).getType() == tiles.get(i - 1).getType()
}
```

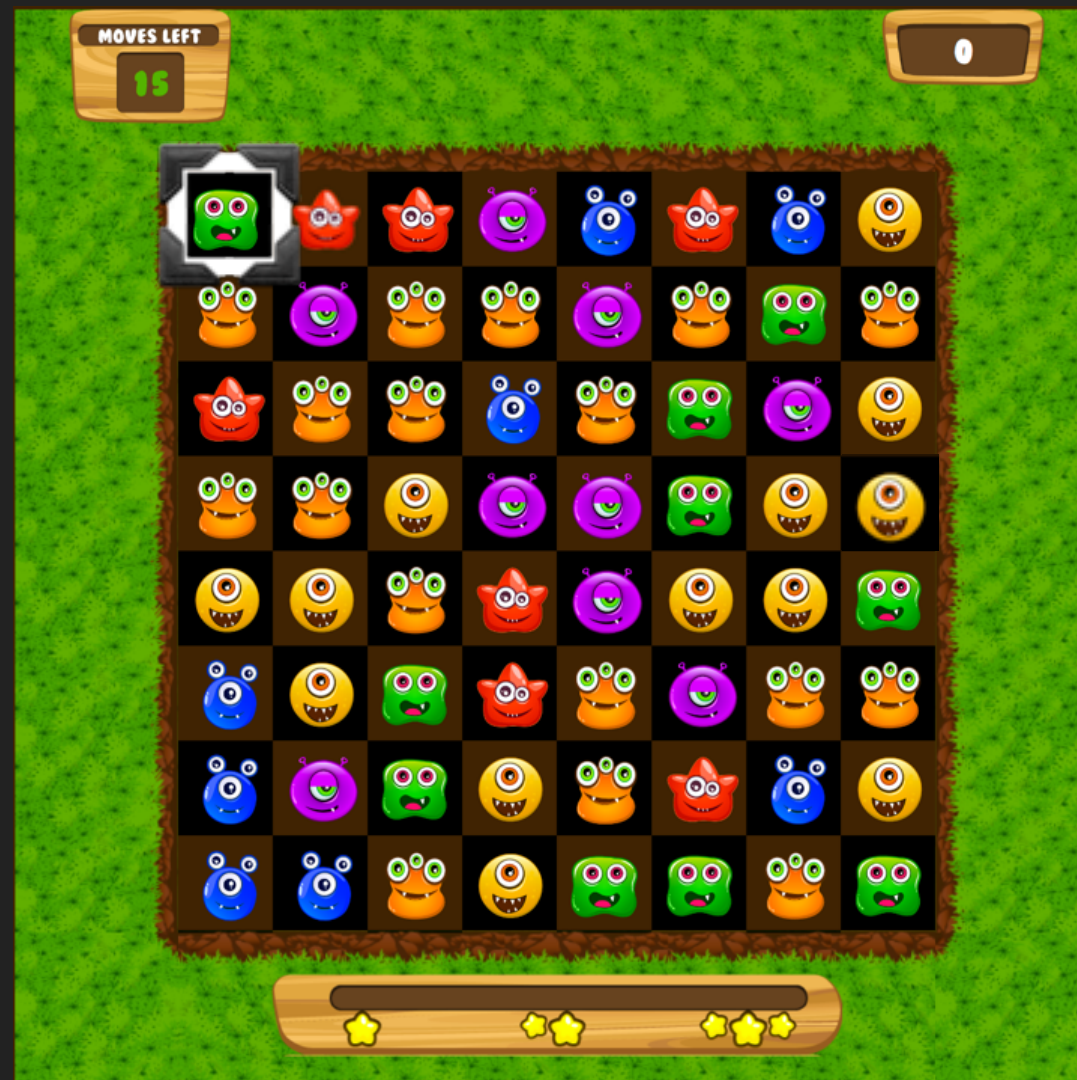


```
public static boolean isHorizontalMatch(Level level, int i) {
    List<Tile> tiles = level.getTiles();
    return tiles.get(i).getType() == tiles.get(i - 1).getType()
        && tiles.get(i).getType() == tiles.get(i - 2).getType()

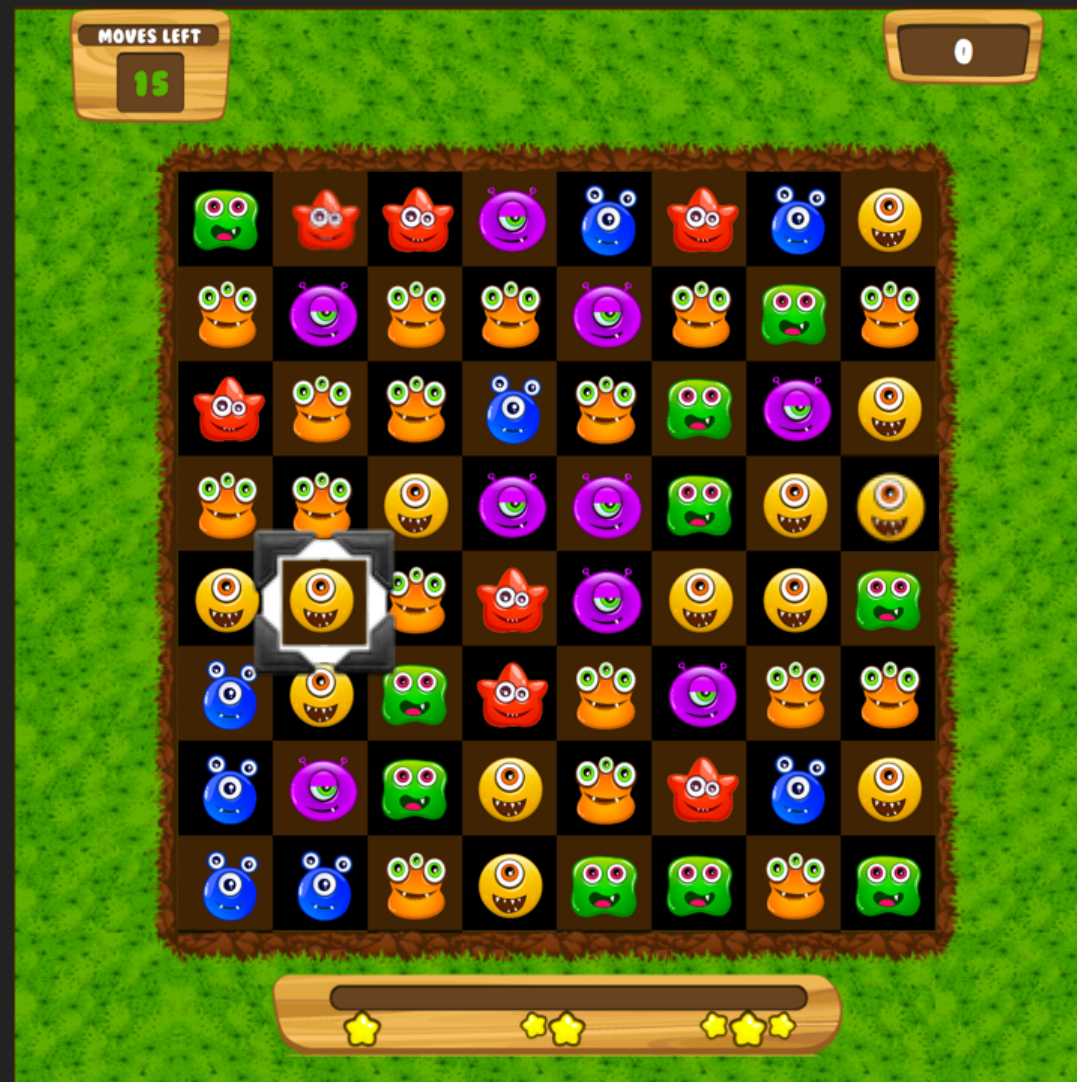
```

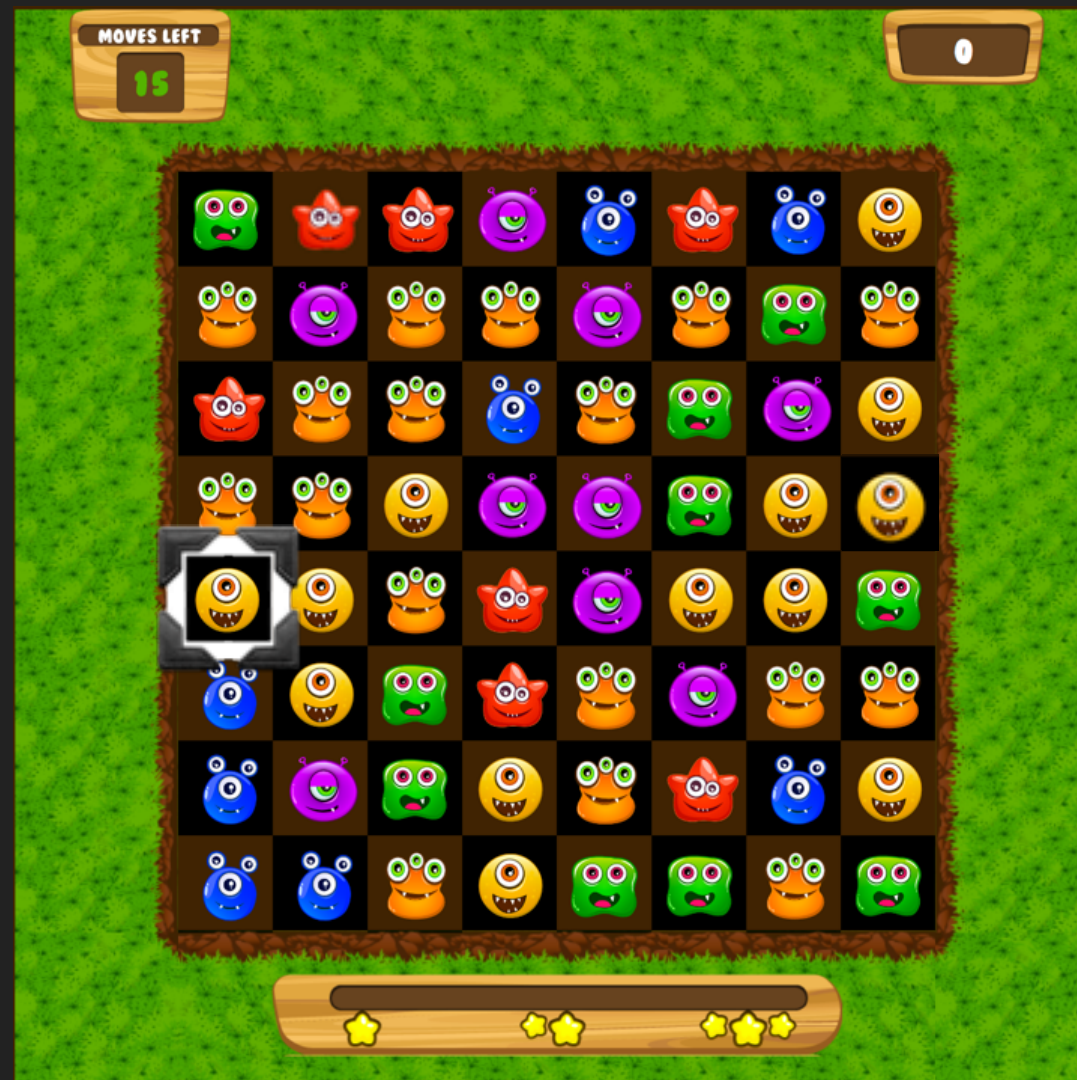
```
public static boolean isHorizontalMatch(Level level, int i) {
    List<Tile> tiles = level.getTiles();
    return tiles.get(i).getType() == tiles.get(i - 1).getType()
        && tiles.get(i).getType() == tiles.get(i - 2).getType()
}
```



```
public static boolean isHorizontalMatch(Level level, int i) {
    List<Tile> tiles = level.getTiles();
    return tiles.get(i).getType() == tiles.get(i - 1).getType()
        && tiles.get(i).getType() == tiles.get(i - 2).getType()
```

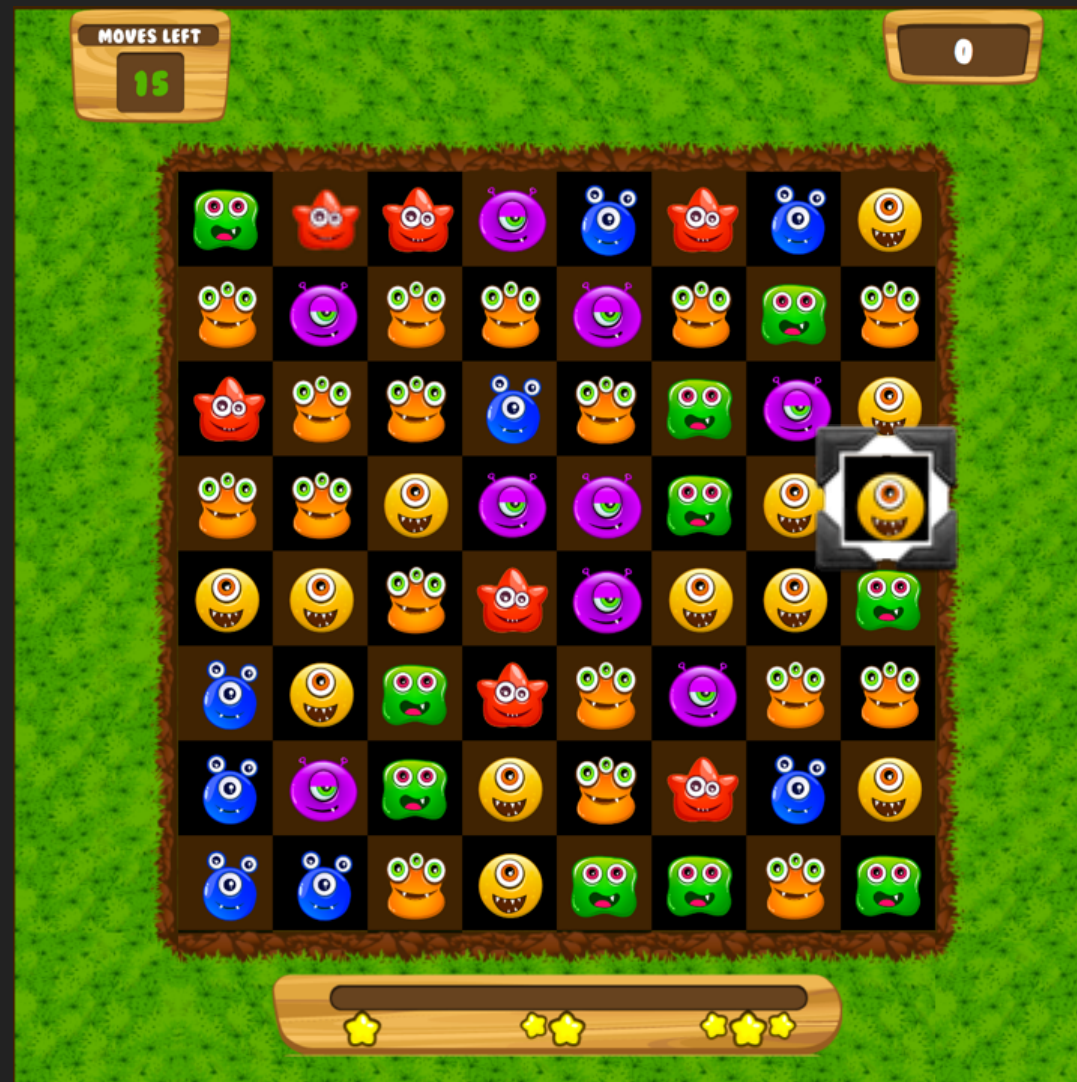



```
public static boolean isHorizontalMatch(Level level, int i) {
    List<Tile> tiles = level.getTiles();
    return tiles.get(i).getType() == tiles.get(i - 1).getType()
        && tiles.get(i).getType() == tiles.get(i - 2).getType()
```

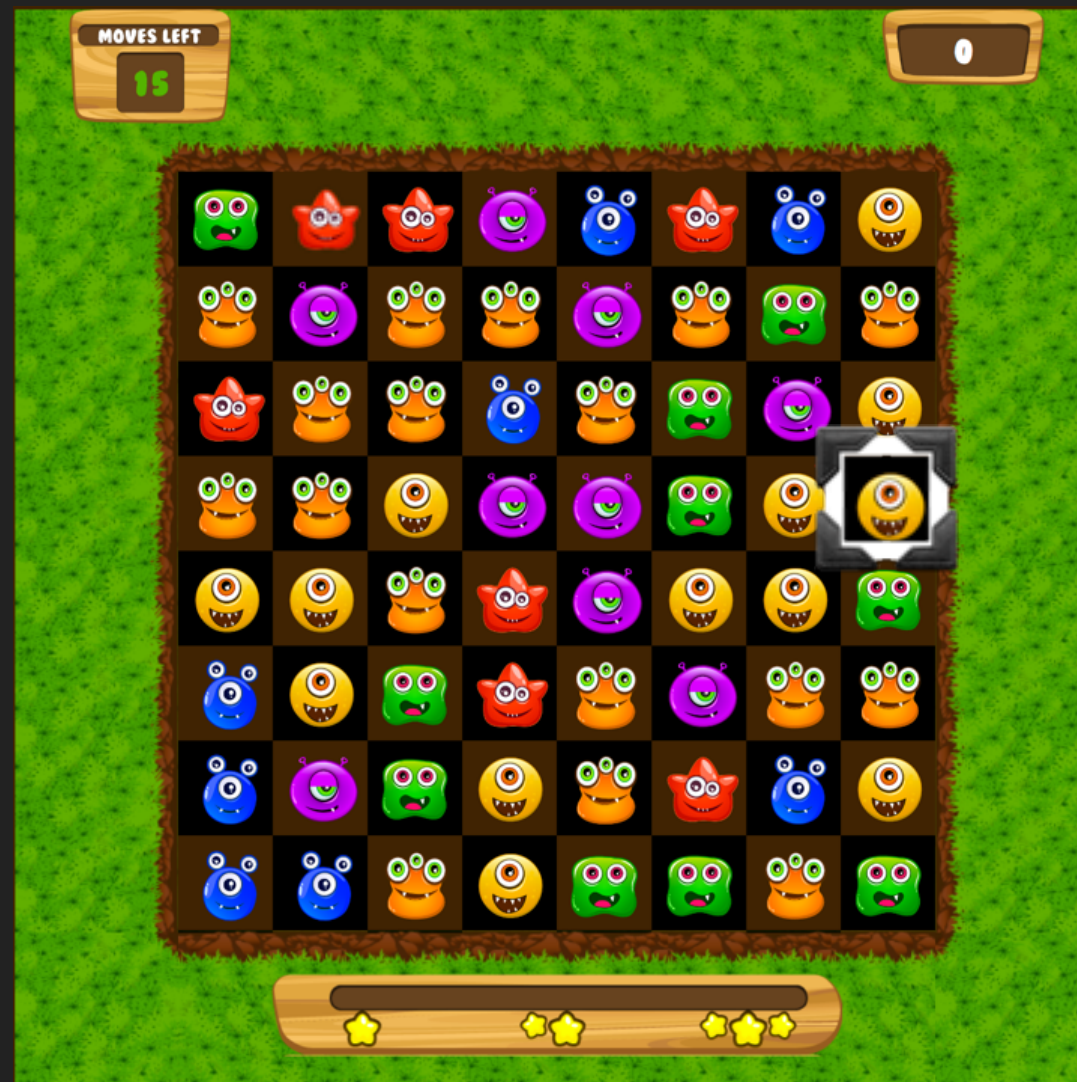


```
public static boolean isHorizontalMatch(Level level, int i) {
    List<Tile> tiles = level.getTiles();
    return tiles.get(i).getType() == tiles.get(i - 1).getType()
        && tiles.get(i).getType() == tiles.get(i - 2).getType()

```

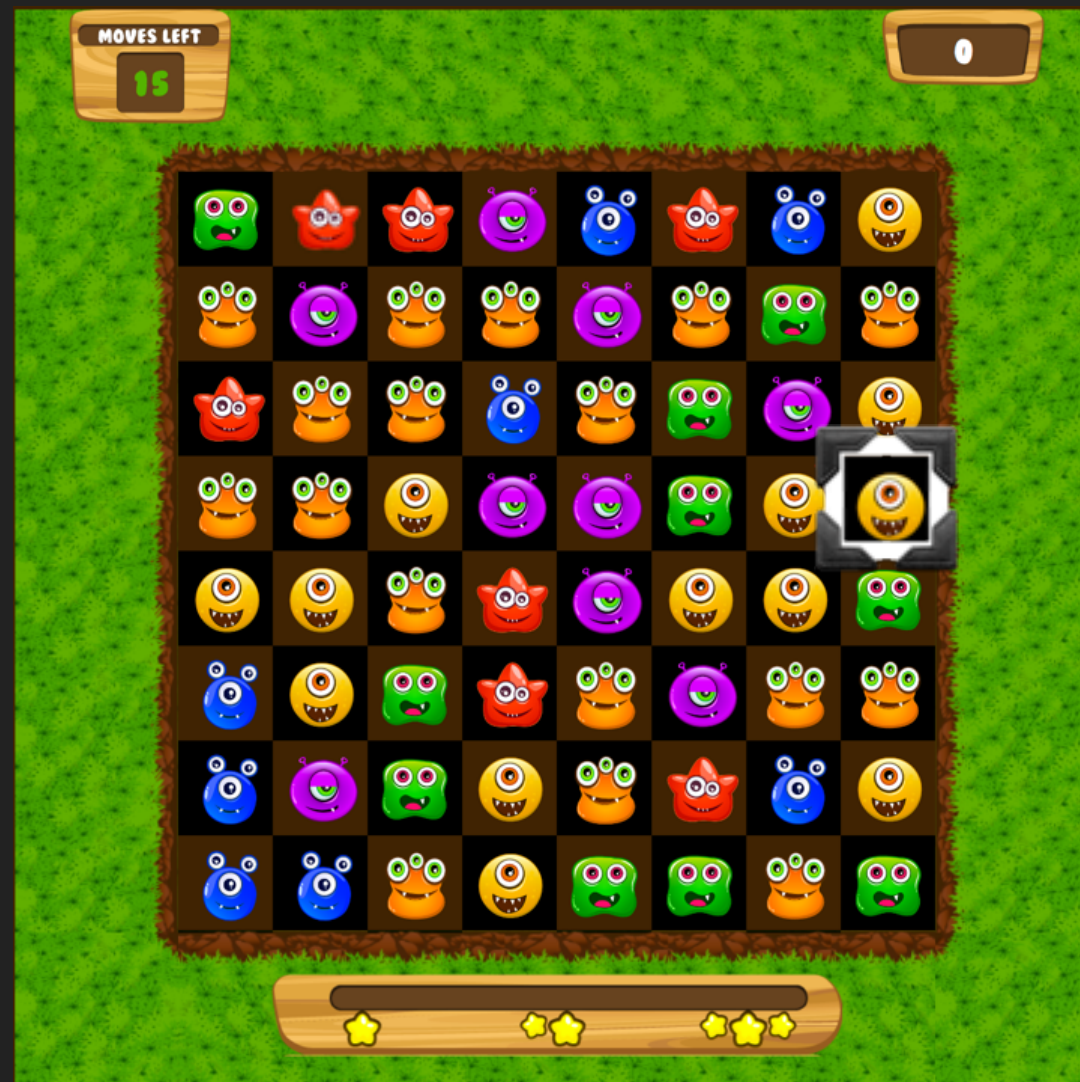
```
public static boolean isHorizontalMatch(Level level, int i) {
    List<Tile> tiles = level.getTiles();
    return tiles.get(i).getType() == tiles.get(i - 1).getType()
        && tiles.get(i).getType() == tiles.get(i - 2).getType()
}
```



```

public static boolean isHorizontalMatch(Level level, int i) {
    List<Tile> tiles = level.getTiles();
    return tiles.get(i).getType() == tiles.get(i - 1).getType()
        && tiles.get(i).getType() == tiles.get(i - 2).getType()
        && columnNumber(level, i) >= 2;
}

```

```

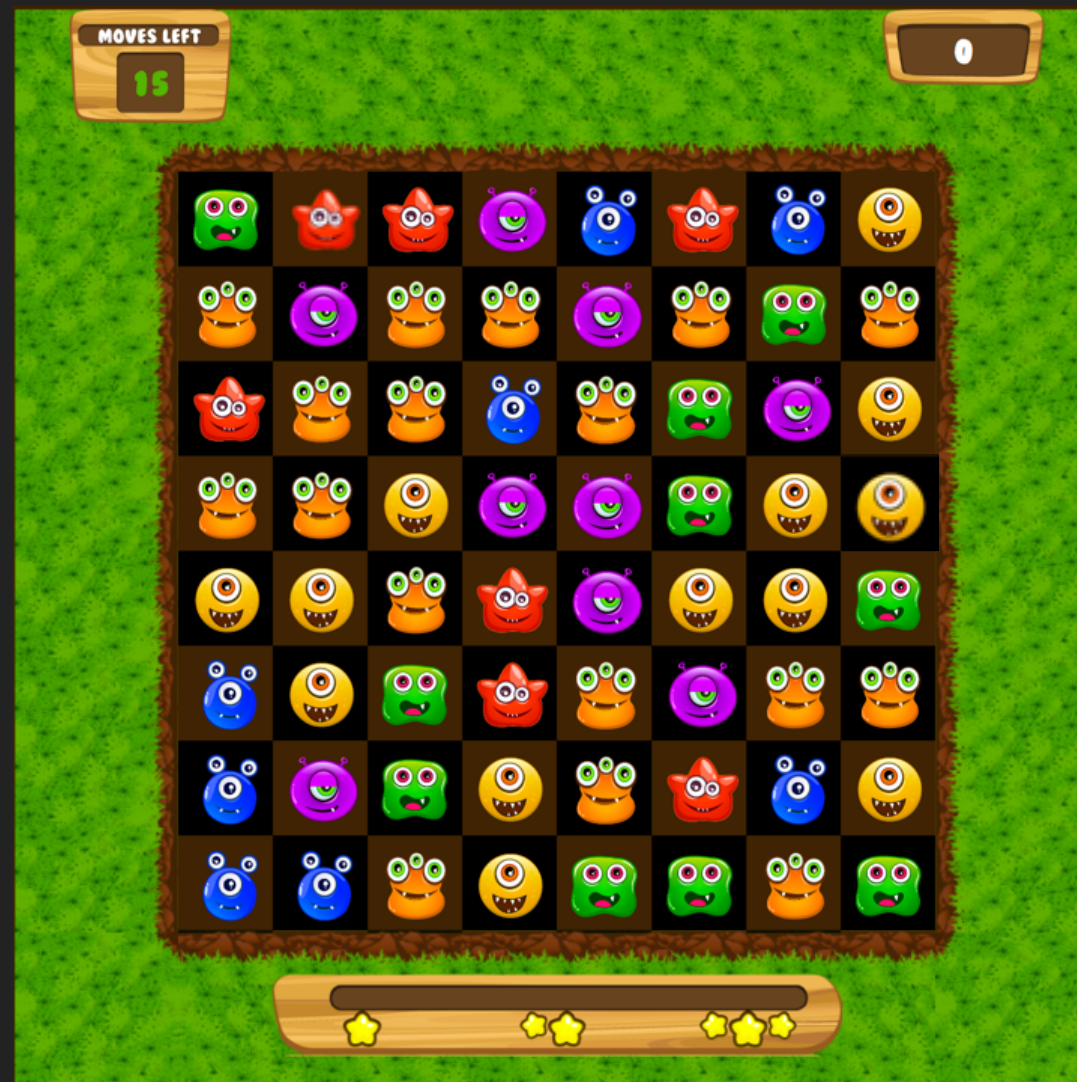
public static boolean isHorizontalMatch(Level level, int i) {
    List<Tile> tiles = level.getTiles();
    return tiles.get(i).getType() == tiles.get(i - 1).getType()
        && tiles.get(i).getType() == tiles.get(i - 2).getType()
        && columnNumber(level, i) >= 2;
}

```

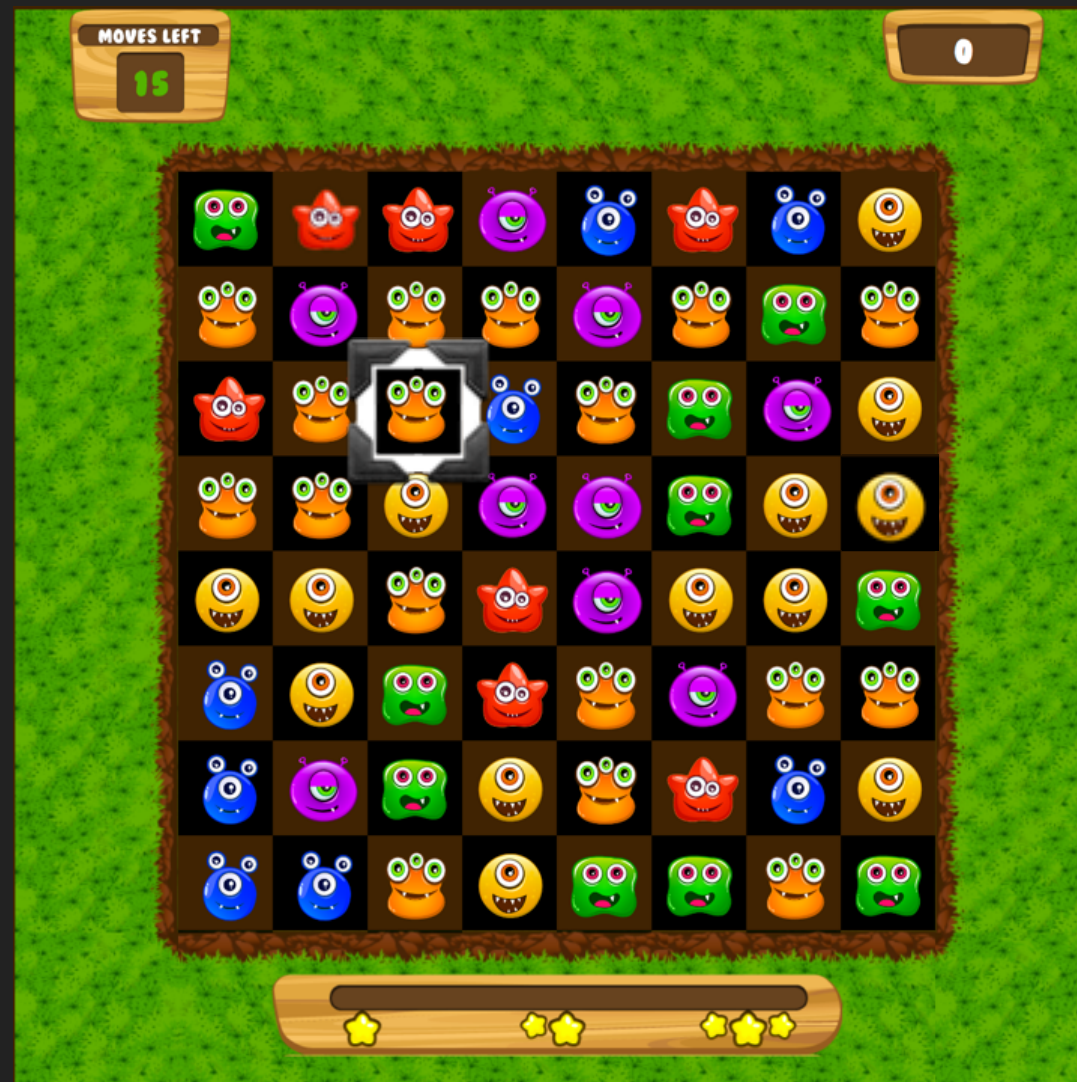
MOVES LEFT
15

0

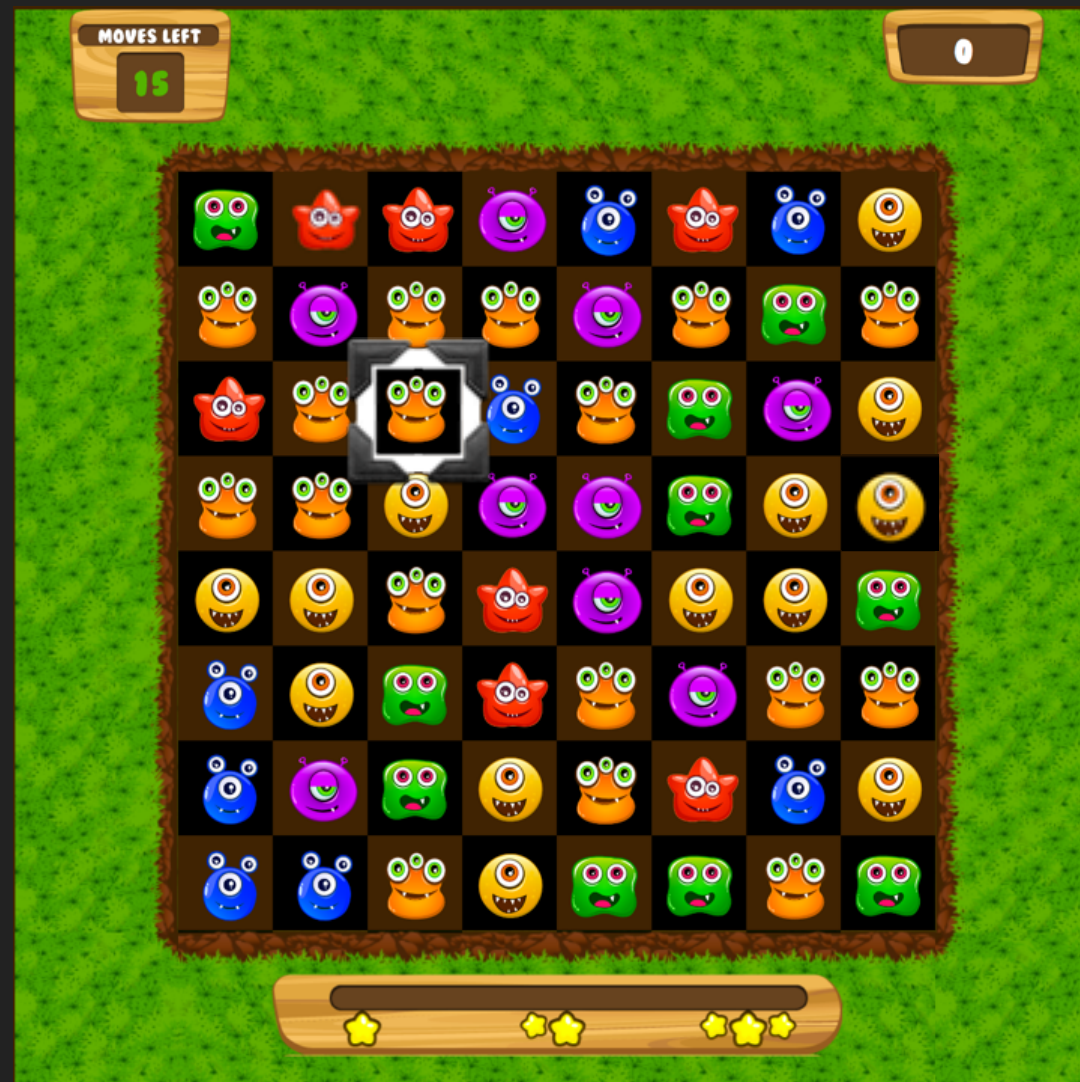




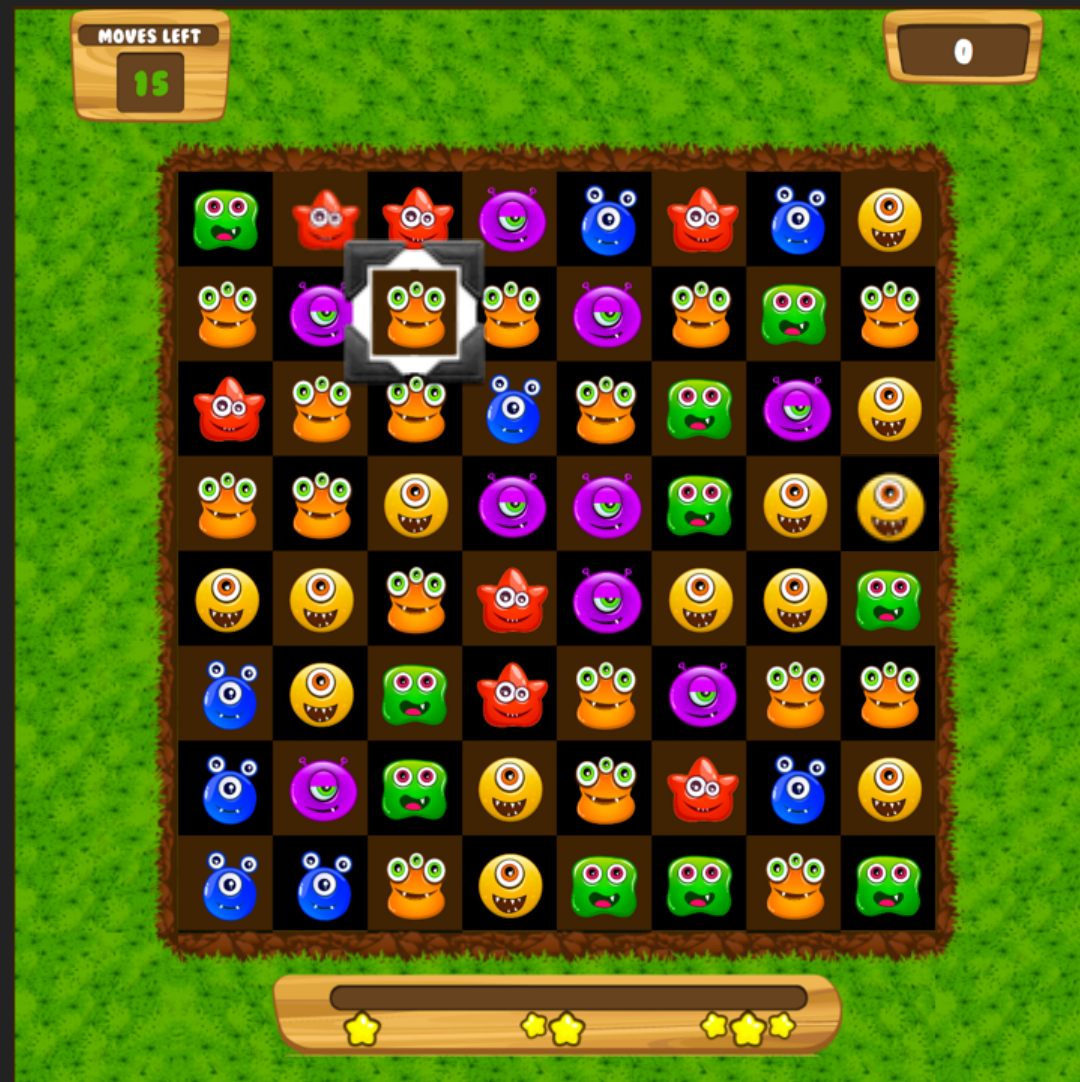
```
public static boolean isVerticalMatch(Level level, int i) {
    List<Tile> tiles = level.getTiles();
    return rowNum(level, i) >= 2
}
```

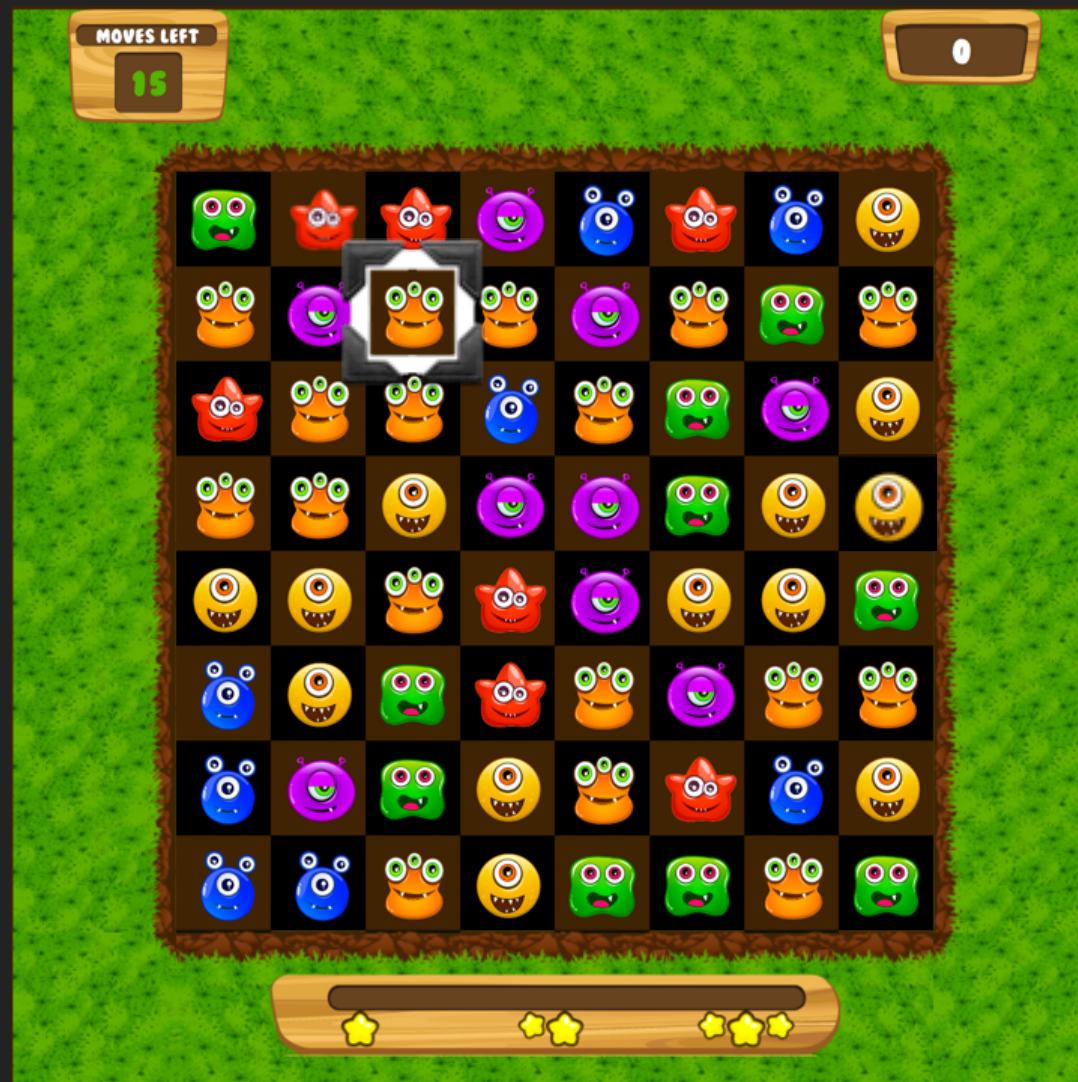
```
public static boolean isVerticalMatch(Level level, int i) {
    List<Tile> tiles = level.getTiles();
    return rowNumber(level, i) >= 2
}
```

```
public static boolean isVerticalMatch(Level level, int i) {
    List<Tile> tiles = level.getTiles();
    return rowNumbr(level, i) >= 2
        && tiles.get(i).getType() == tiles.get(i - level.getColums()).getType()
```



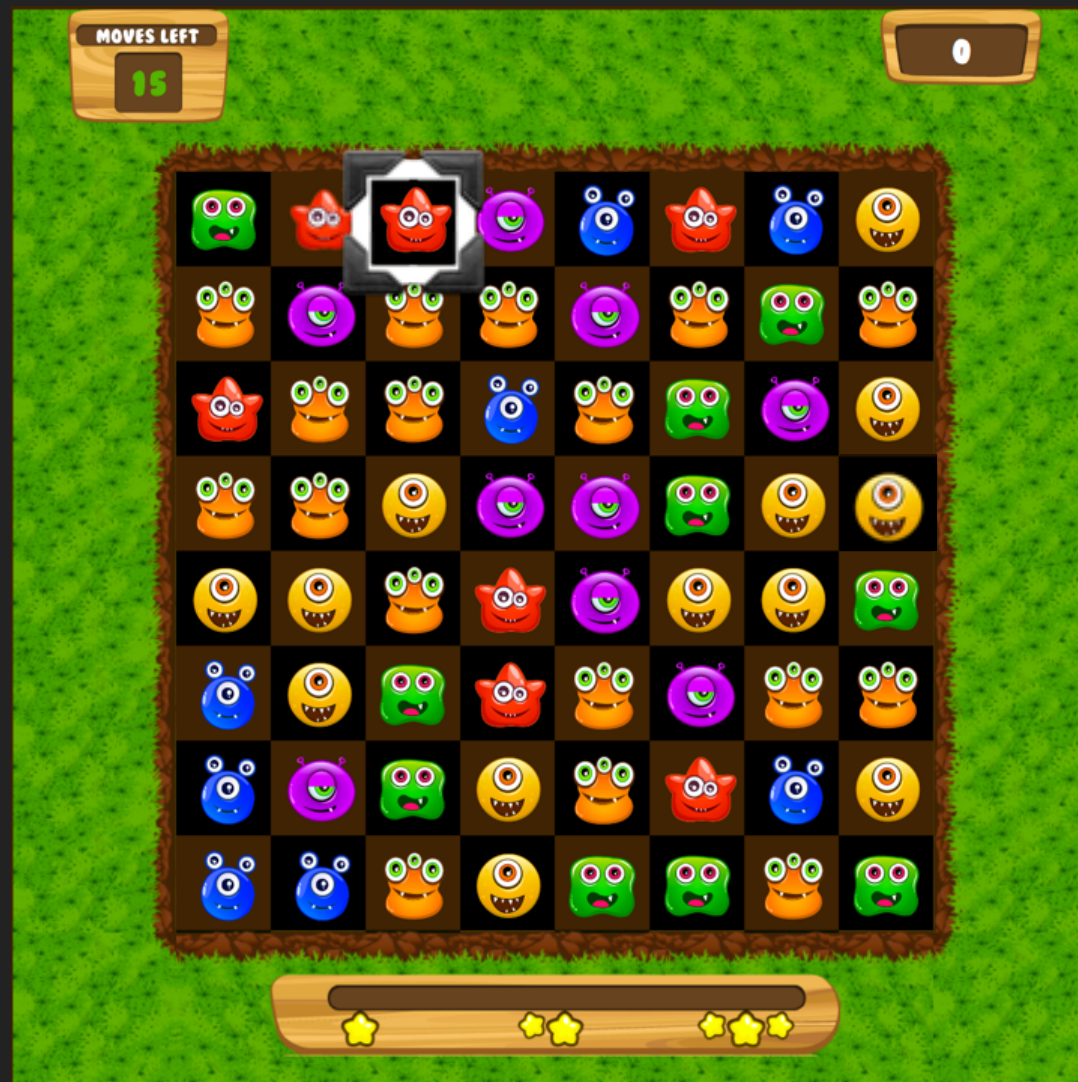
```
public static boolean isVerticalMatch(Level level, int i) {
    List<Tile> tiles = level.getTiles();
    return rowNum(level, i) >= 2
        && tiles.get(i).getType() == tiles.get(i - level.getColumns()).getType()
```

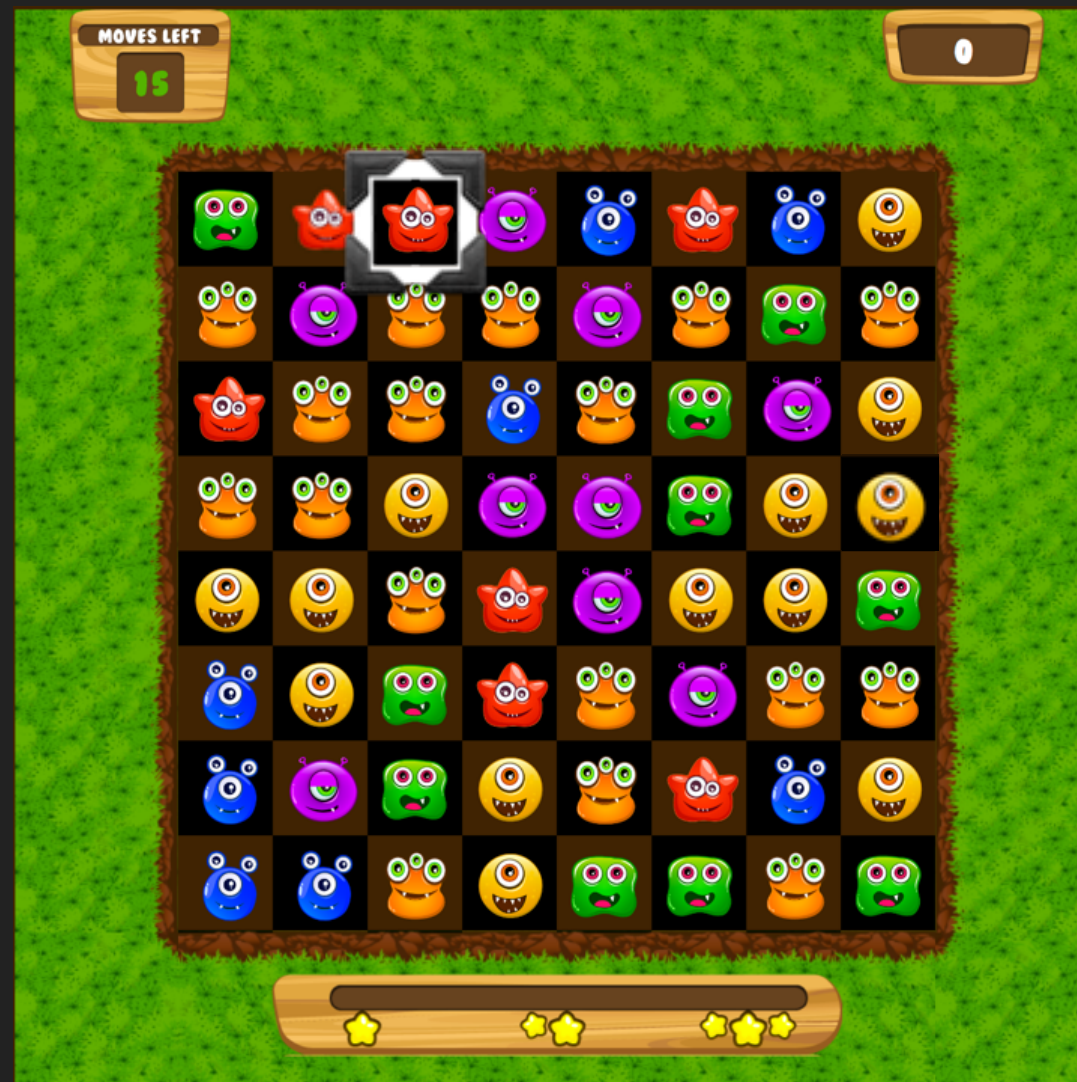
```

public static boolean isVerticalMatch(Level level, int i) {
    List<Tile> tiles = level.getTiles();
    return rowNumNumber(level, i) >= 2
        && tiles.get(i).getType() == tiles.get(i - level.getColumns()).getType()
        && tiles.get(i).getType() == tiles.get(i - 2 * level.getColumns()).getType()
}

```



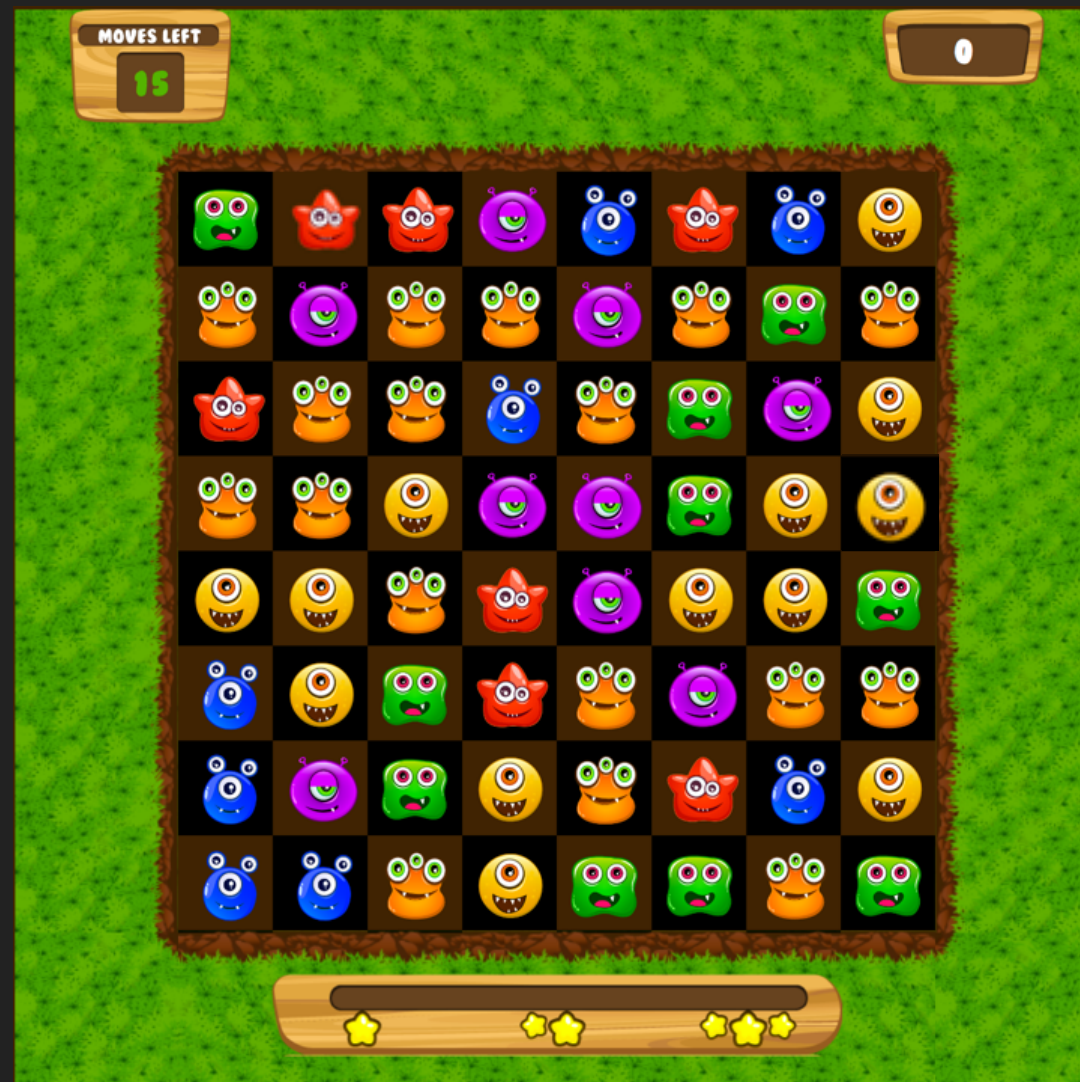
```
public static boolean isVerticalMatch(Level level, int i) {
    List<Tile> tiles = level.getTiles();
    return rowNumNumber(level, i) >= 2
        && tiles.get(i).getType() == tiles.get(i - level.getColumns()).getType()
        && tiles.get(i).getType() == tiles.get(i - 2 * level.getColumns()).getType()
}
```

```

public static boolean isVerticalMatch(Level level, int i) {
    List<Tile> tiles = level.getTiles();
    return rowNumNumber(level, i) >= 2
        && tiles.get(i).getType() == tiles.get(i - level.getColums()).getType()
        && tiles.get(i).getType() == tiles.get(i - 2 * level.getColums()).getType()
}

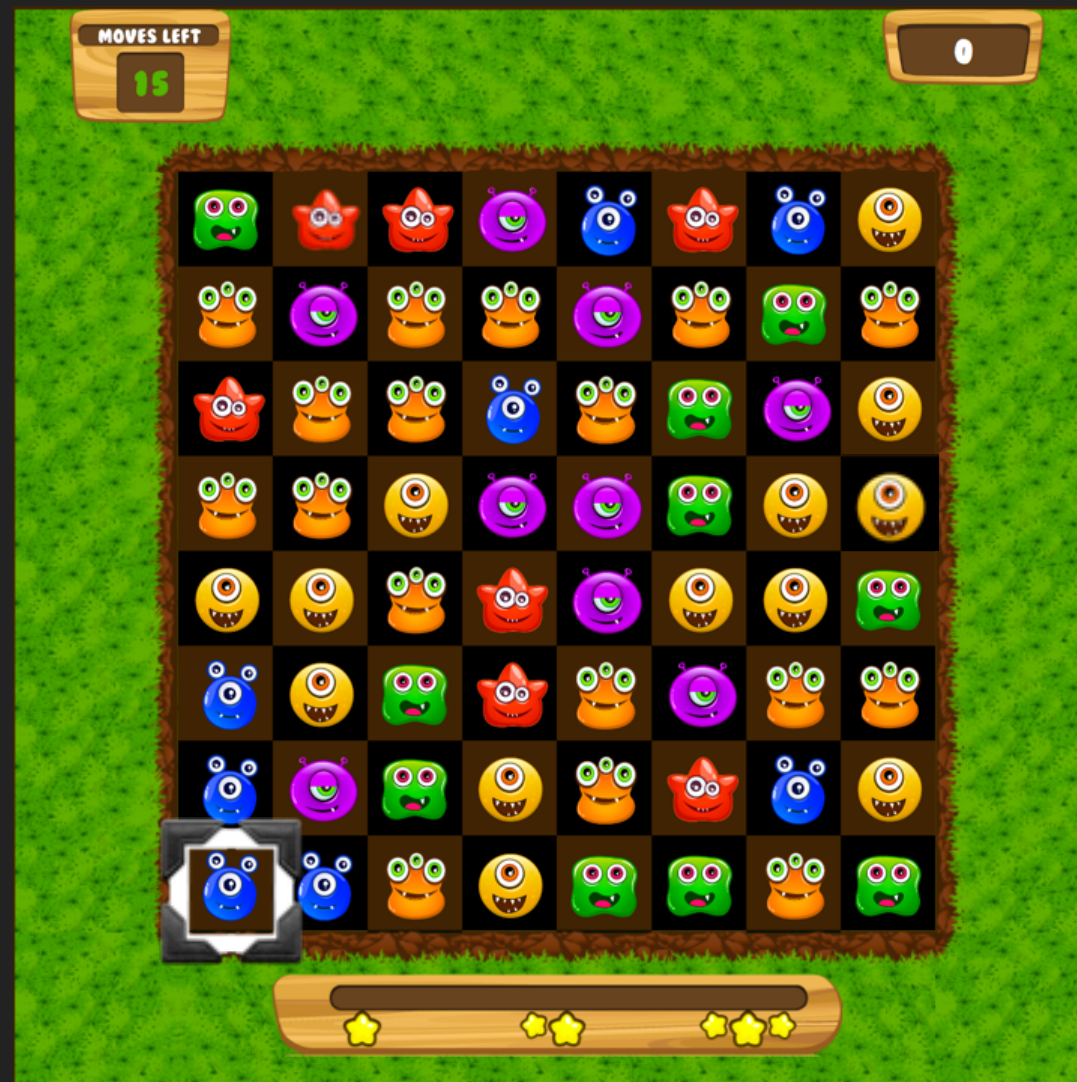
```



```

public static boolean isVerticalMatch(Level level, int i) {
    List<Tile> tiles = level.getTiles();
    return rowNumNumber(level, i) >= 2
        && tiles.get(i).getType() == tiles.get(i - level.getColumns()).getType()
        && tiles.get(i).getType() == tiles.get(i - 2 * level.getColumns()).getType()
}

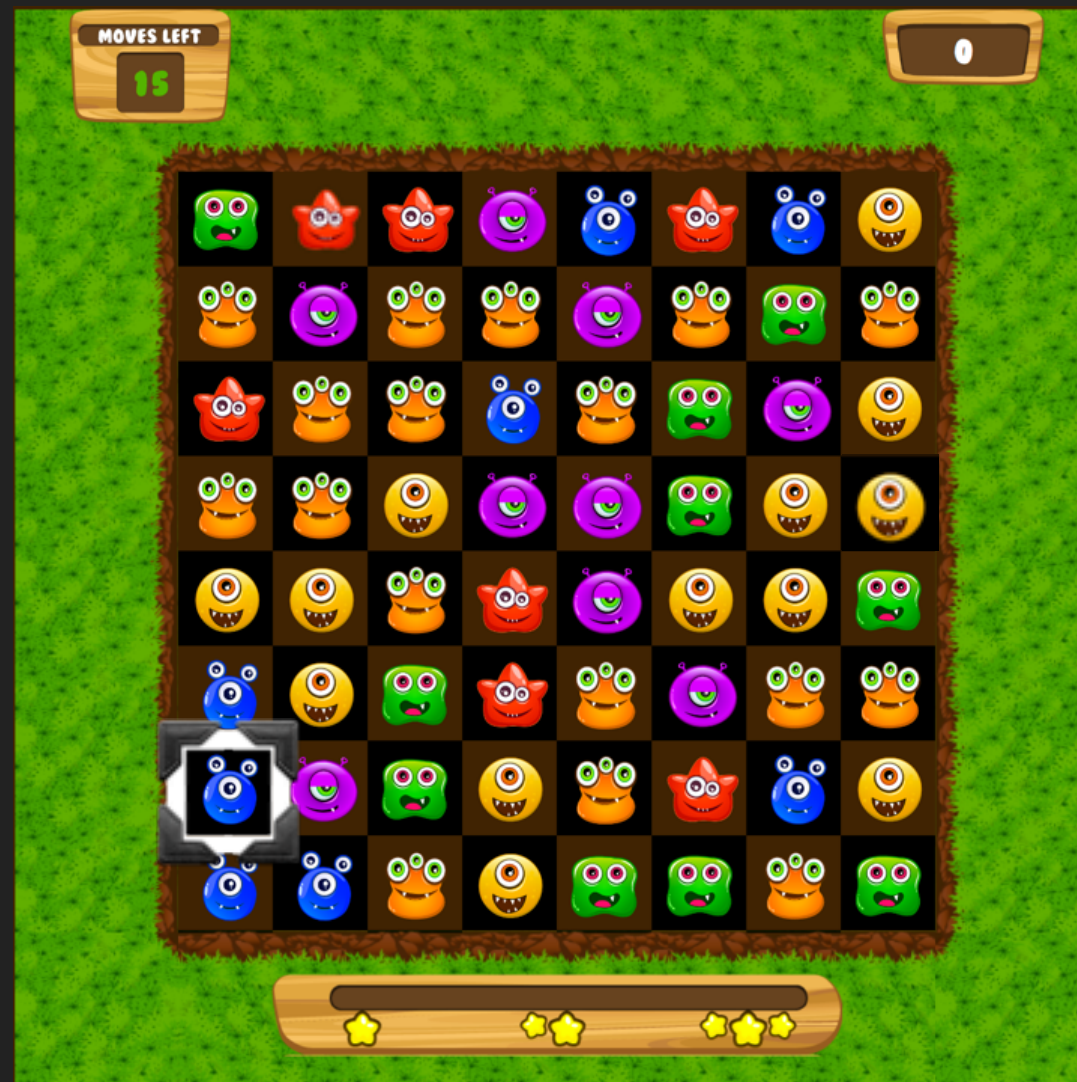
```



```

public static boolean isVerticalMatch(Level level, int i) {
    List<Tile> tiles = level.getTiles();
    return rowNumNumber(level, i) >= 2
        && tiles.get(i).getType() == tiles.get(i - level.getColums()).getType()
        && tiles.get(i).getType() == tiles.get(i - 2 * level.getColums()).getType()
}

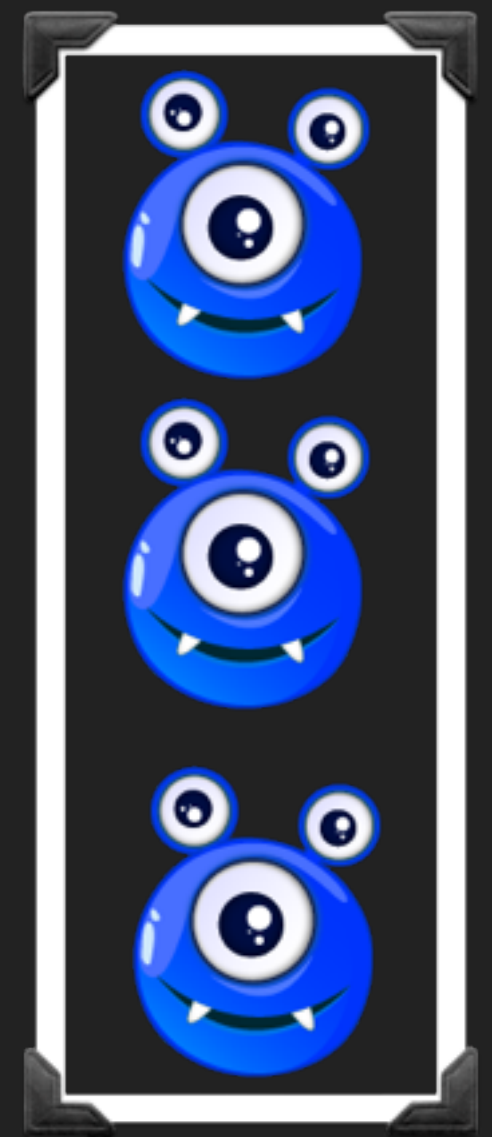
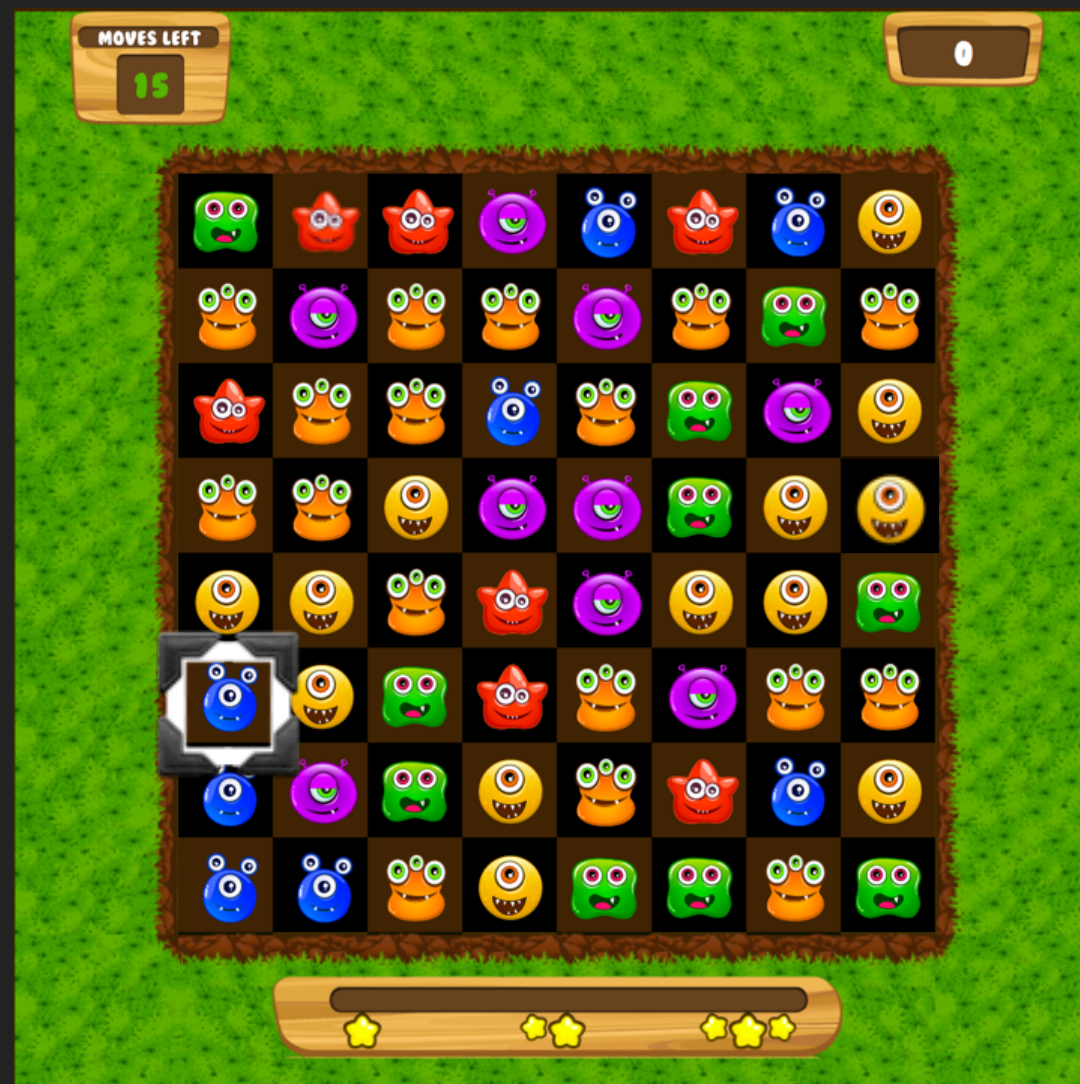
```

```

public static boolean isVerticalMatch(Level level, int i) {
    List<Tile> tiles = level.getTiles();
    return rowNumbr(level, i) >= 2
        && tiles.get(i).getType() == tiles.get(i - level.getColums()).getType()
        && tiles.get(i).getType() == tiles.get(i - 2 * level.getColums()).getType()
}

```

```

public static boolean isVerticalMatch(Level level, int i) {
    List<Tile> tiles = level.getTiles();
    return rowNum(level, i) >= 2
        && tiles.get(i).getType() == tiles.get(i - level.getColums()).getType()
        && tiles.get(i).getType() == tiles.get(i - 2 * level.getColums()).getType()
}

```


MOVES LEFT

15

0



MOVES LEFT

15

0





ACHIEVEMENT UNLOCKED!



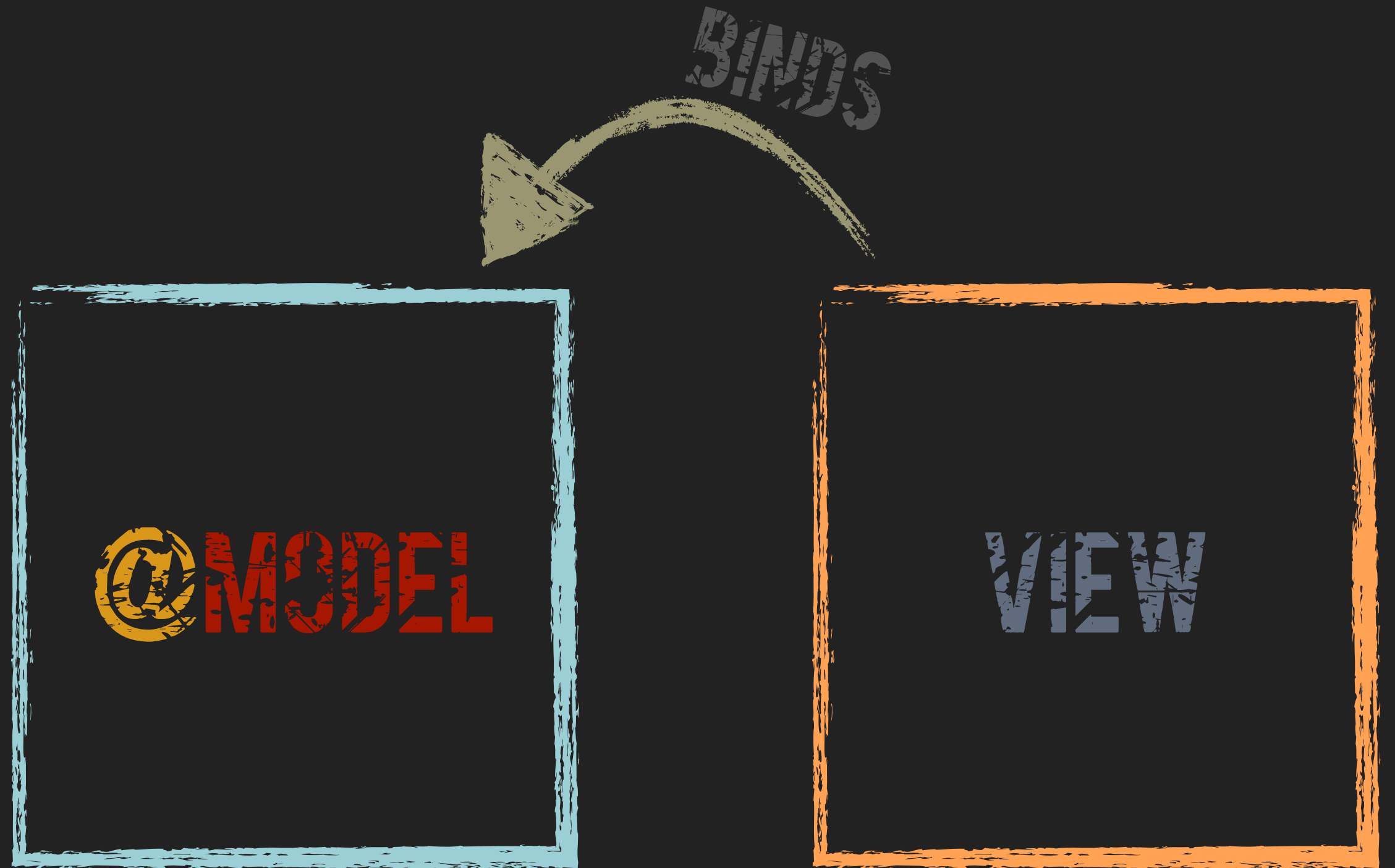
MATCH 3 CALCULATOR



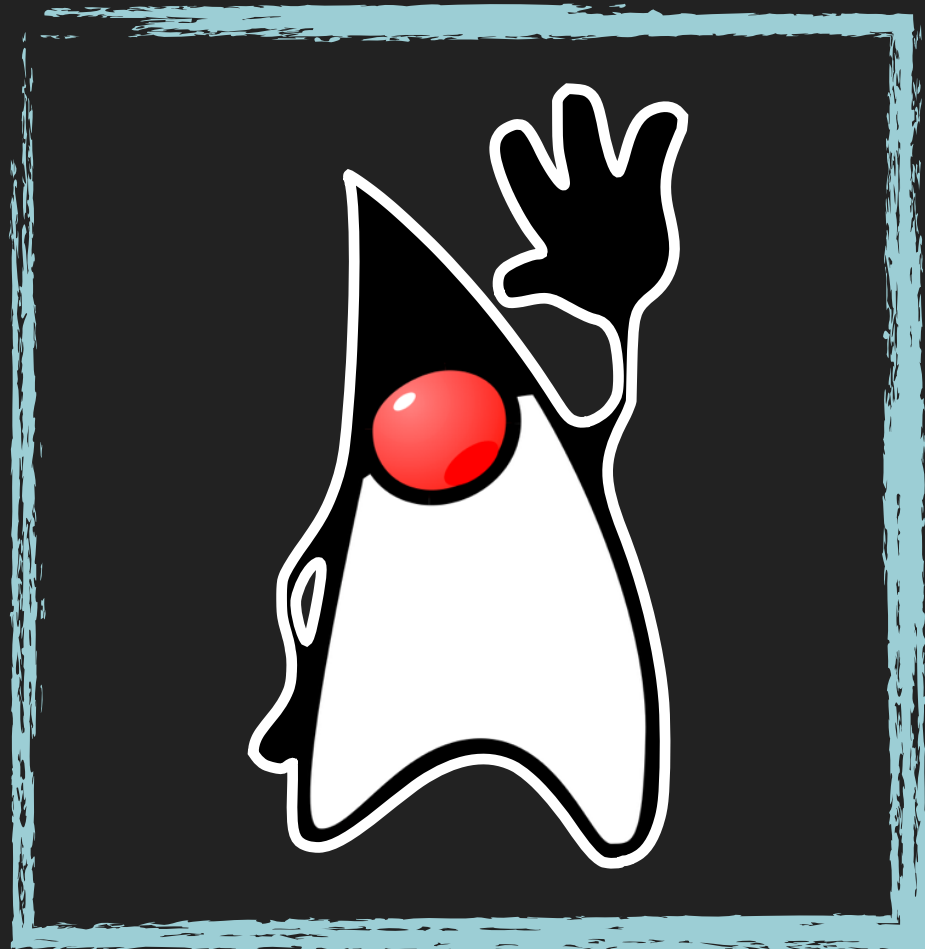
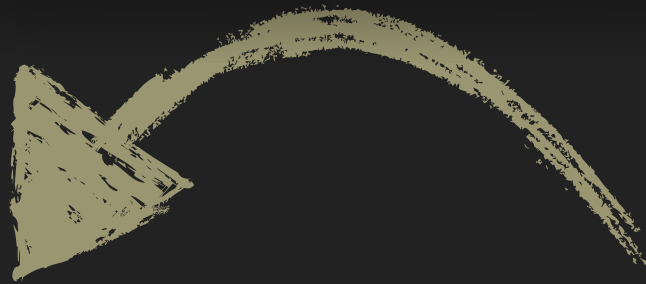








Knockout.



Knockout.

Knockout.

data-bind="<binding>: <property> [...]"

databind="<binding>: <property> [...]"

`databind="<binding>: <property> [...]"`

`<!--show how many moves are left-->`

`data-bind="<binding>: <property> [...]"`

`<!--show how many moves are left-->`

`<div id="moves-board" data-bind="text: moves"></div>`

`data-bind="<binding>: <property> [...]"`

`<!--show how many moves are left-->`

`<div id="moves-board" data-bind="text: moves"></div>`

`<!--show current score-->`

`data-bind="<binding>: <property> [...]"`

`<!--show how many moves are left-->`

`<div id="moves-board" data-bind="text: moves"></div>`

`<!--show current score-->`

`<div id="score-board" data-bind="text: score"></div>`

databind="<binding>: <property> [...]"

data-bind="<binding>: <property> [...]"

<div id="grid" data-bind="foreach: tiles">

data-bind="<binding>: <property> [...]"

```
<div id="grid" data-bind="foreach: tiles">
```

```
  <div class="tile"
```

data-bind="<binding>: <property> [...]"

```
<div id="grid" data-bind="foreach: tiles">
```

```
  <div class="tile"
```

```
    data-bind="css: 'tile-position-'+$index()+ ' tile-type-'+type()">
```

```
  </div>
```

```
</div>
```

data-bind="<binding>: <property> [...]"

```
<div id="grid" data-bind="foreach: tiles">
```

```
  <div class="tile"
```

```
    data-bind="css: 'tile-position-'+$index()+' tile-type-'+type()">
```

```
  </div>
```

```
</div>
```

```
Scope <div id="grid" >
  <div class="tile"
    data-bind="css: 'tile-position-'+$index()+' tile-type-'+type()">
  </div>
  <div class="tile"
    data-bind="css: 'tile-position-'+$index()+' tile-type-'+type()">
  </div>
  <div class="tile"
    data-bind="css: 'tile-position-'+$index()+' tile-type-'+type()">
  </div>

  [63 times...]

</div>
```


Scope `<div id="grid" >`

Tile 0



```
<div class="tile"
  data-bind="css: 'tile-position-'+$index()+ ' tile-type-'+type()">
</div>
```

```
<div class="tile"
  data-bind="css: 'tile-position-'+$index()+ ' tile-type-'+type()">
</div>
```

```
<div class="tile"
  data-bind="css: 'tile-position-'+$index()+ ' tile-type-'+type()">
</div>
```

[63 times...]

```
</div>
```

Scope `<div id="grid" >`

Tile 0



```
<div class="tile"
  data-bind="css: 'tile-position-'+$index()+' tile-type-'+type()">
</div>
```

Tile 1



```
<div class="tile"
  data-bind="css: 'tile-position-'+$index()+' tile-type-'+type()">
</div>
```

```
<div class="tile"
  data-bind="css: 'tile-position-'+$index()+' tile-type-'+type()">
</div>
```

[63 times...]

`</div>`

Scope `<div id="grid" >`

Tile 0



```
<div class="tile"
  data-bind="css: 'tile-position-'+$index()+ ' tile-type-'+type()">
</div>
```

Tile 1



```
<div class="tile"
  data-bind="css: 'tile-position-'+$index()+ ' tile-type-'+type()">
</div>
```

Tile 2



```
<div class="tile"
  data-bind="css: 'tile-position-'+$index()+ ' tile-type-'+type()">
</div>
```

[63 times...]

`</div>`

Scope `<div id="grid" >`

Tile 0



```
<div class="tile  
      tile-position-0 tile-type-1">  
</div>
```

Tile 1



```
<div class="tile  
      tile-position-1 tile-type-4">  
</div>
```

Tile 2



```
<div class="tile  
      tile-position-2 tile-type-4">  
</div>
```

[63 times...]

`</div>`

Scope `<div id="grid" >`

Tile 0



```
<div class="tile  
      tile-position-0 tile-type-1">  
</div>
```

Tile 1



```
<div class="tile  
      tile-position-1 tile-type-4">  
</div>
```

Tile 2



```
<div class="tile  
      tile-position-2 tile-type-4">  
</div>
```

`[63 times...]`

`</div>`

Scope `<div id="grid" >`

Tile 0



```
<div class="tile  
      tile-position-0 tile-type-1">  
</div>
```

Tile 1



```
<div class="tile  
      tile-position-1 tile-type-4">  
</div>
```

Tile 2



```
<div class="tile  
      tile-position-2 tile-type-4">  
</div>
```

`[63 times...]`

`</div>`

`// Update ViewModel in Java`

Scope `<div id="grid" >`

Tile 0



```
<div class="tile  
      tile-position-0 tile-type-1">  
</div>
```

Tile 1



```
<div class="tile  
      tile-position-1 tile-type-4">  
</div>
```

Tile 2



```
<div class="tile  
      tile-position-2 tile-type-4">  
</div>
```

[63 times...]

```
</div>
```

```
// Update ViewModel in Java
```

```
Tile tile0 = level.getTiles().get(0);
```

Scope `<div id="grid" >`

Tile 0



```
<div class="tile  
      tile-position-0 tile-type-1">  
</div>
```

Tile 1



```
<div class="tile  
      tile-position-1 tile-type-4">  
</div>
```

Tile 2



```
<div class="tile  
      tile-position-2 tile-type-4">  
</div>
```

[63 times...]

`</div>`

`// Update ViewModel in Java`

`Tile tile0 = level.getTiles().get(0);`

`tile0.setType(4);`

Scope `<div id="grid" >`

Tile 0



```
<div class="tile  
      tile-position-0 tile-type-4">  
</div>
```

Tile 1



```
<div class="tile  
      tile-position-1 tile-type-4">  
</div>
```

Tile 2



```
<div class="tile  
      tile-position-2 tile-type-4">  
</div>
```

[63 times...]

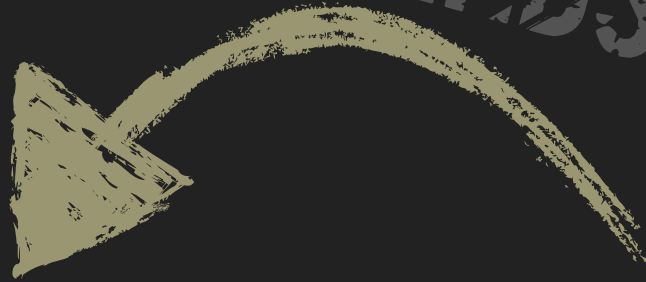
`</div>`

`// Update ViewModel in Java`

`Tile tile0 = level.getTiles().get(0);`

`tile0.setType(4);`

BINDS



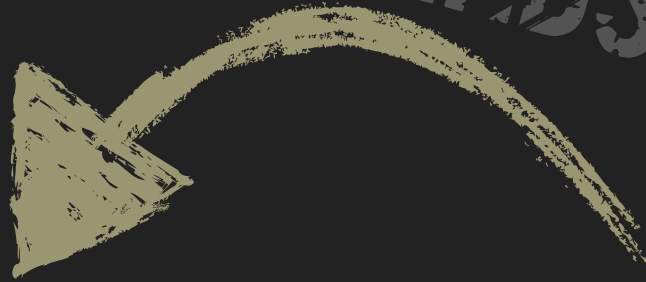
1

@MODEL



VIEW

BINDS



4

@MODEL



VIEW



ACHIEVEMENT UNLOCKED!



KNOCKOUT NOVICE



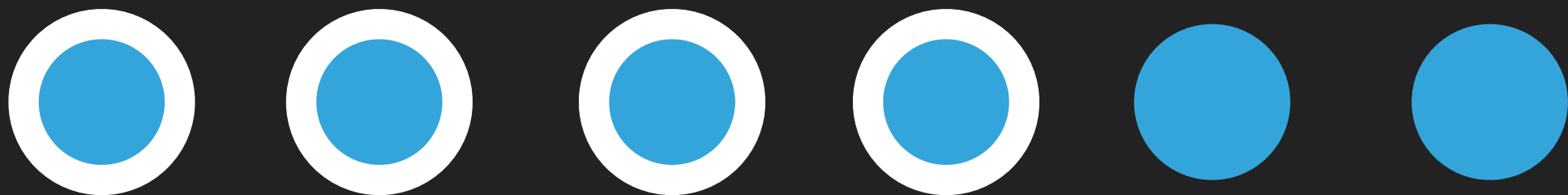


RESPONSIVE LAYOUT





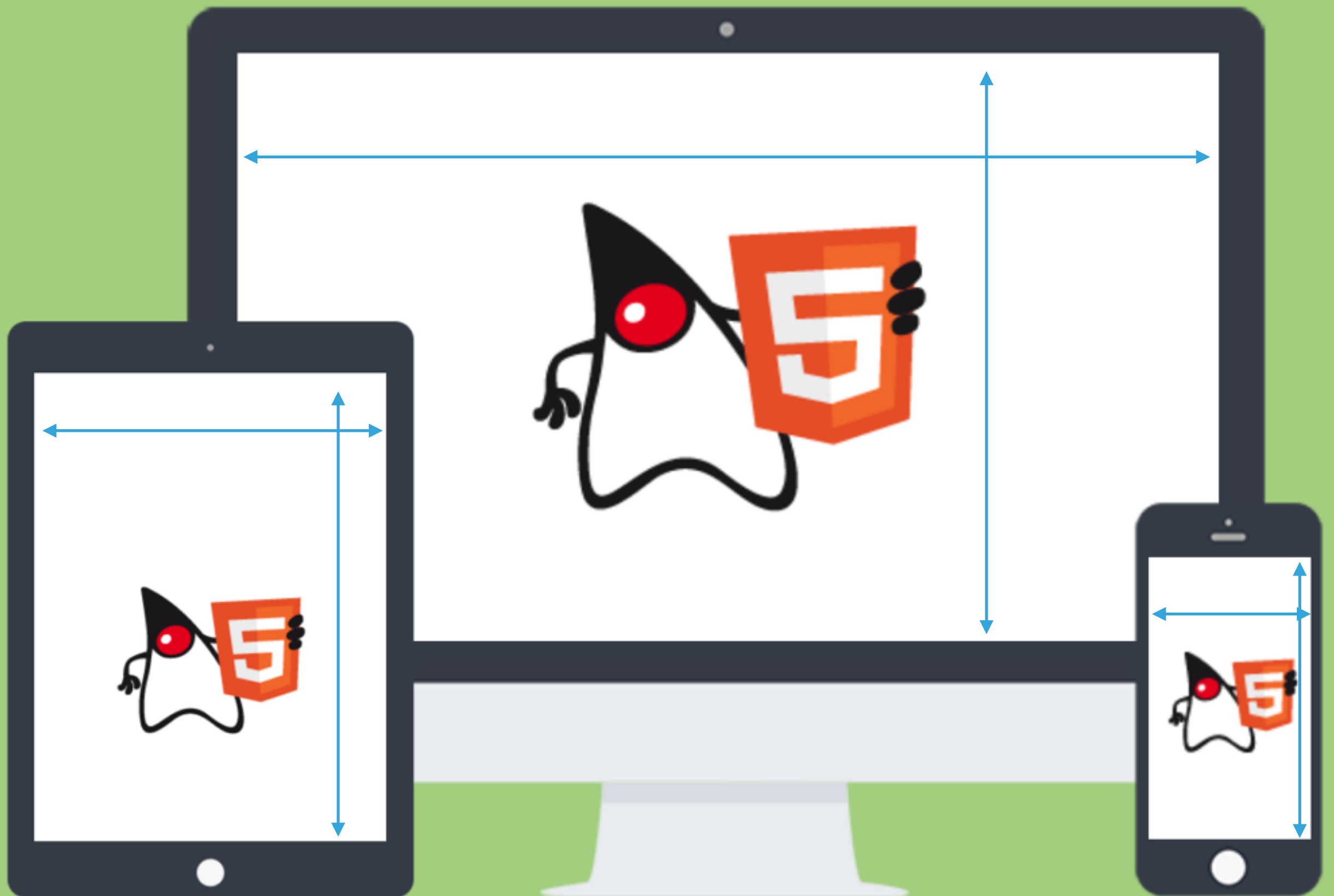
RESPONSIVE LAYOUT



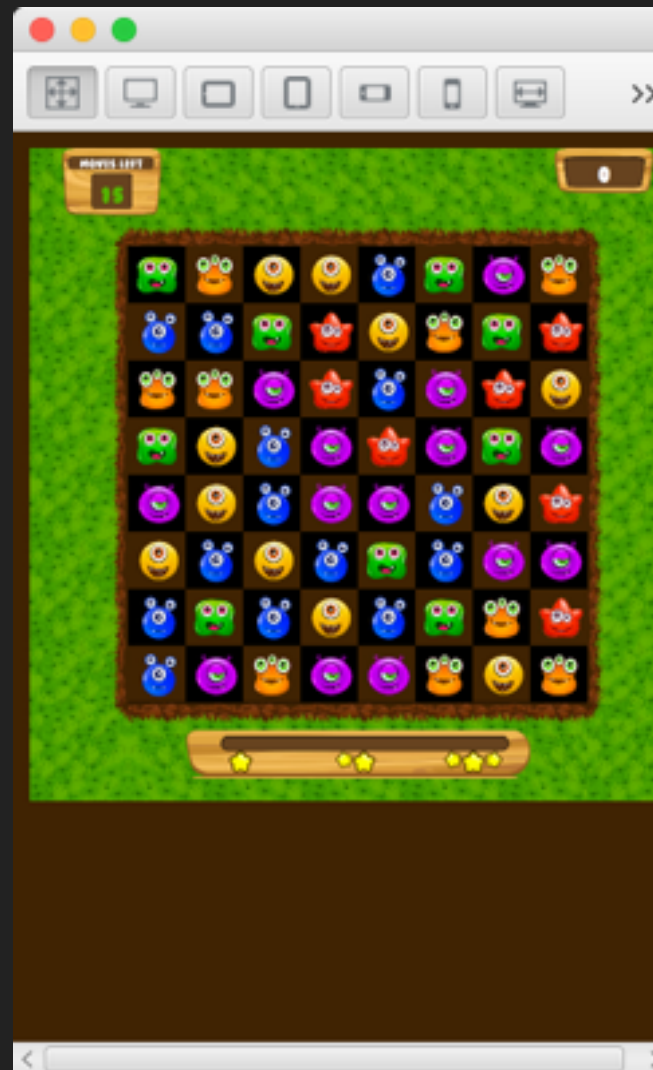
CS5



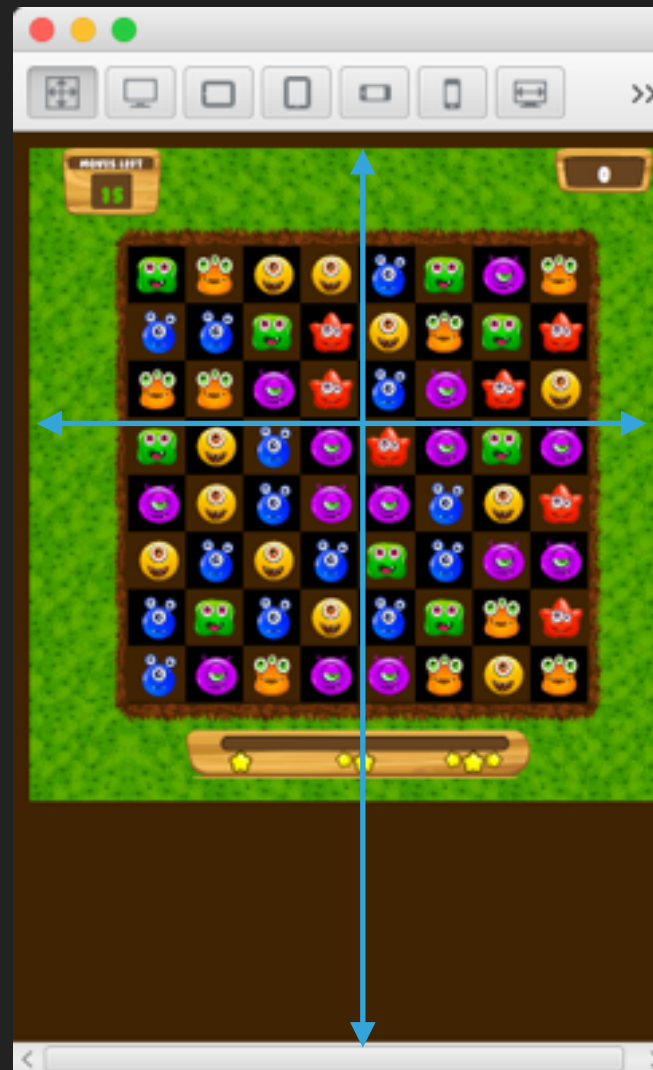




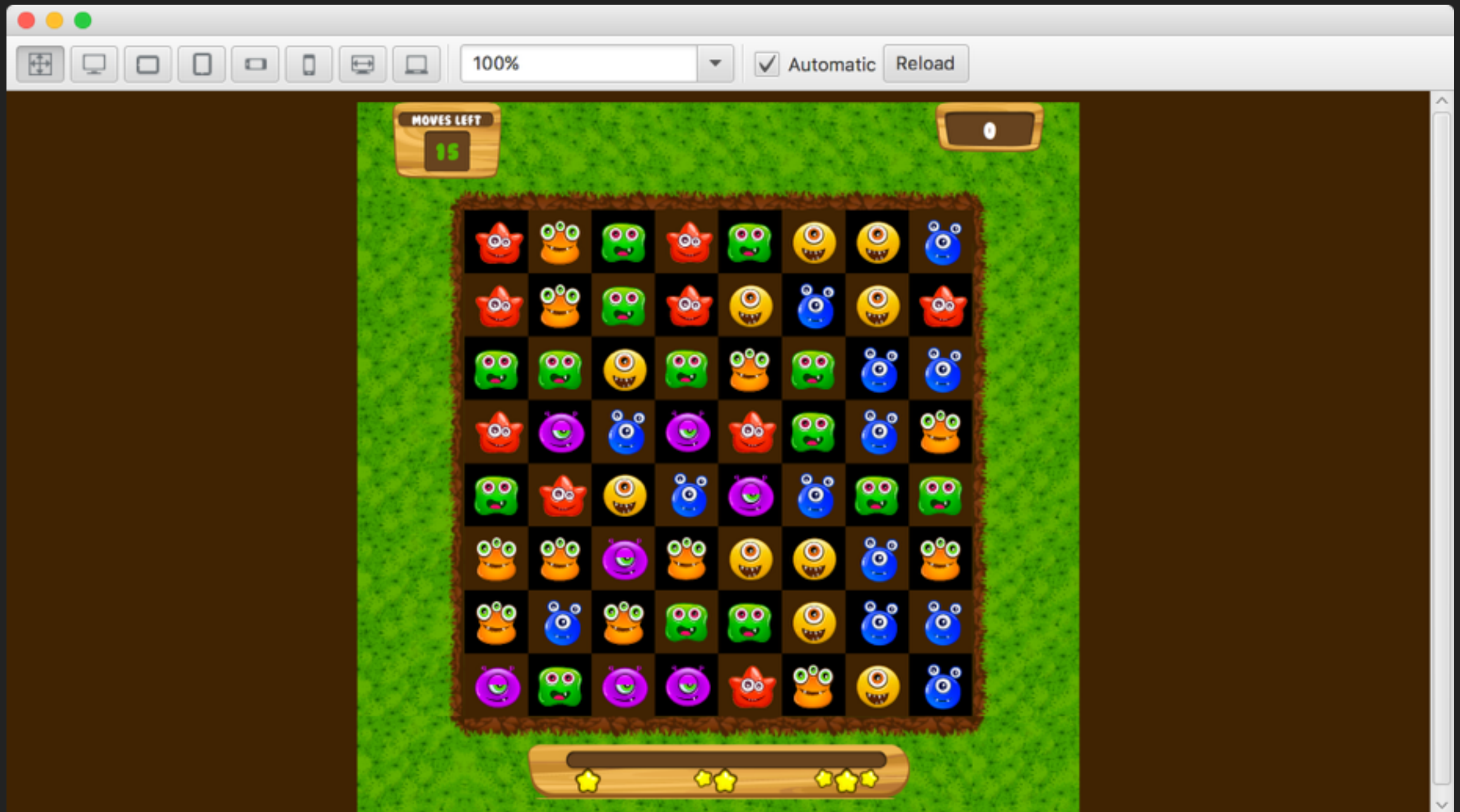
PORTRAIT



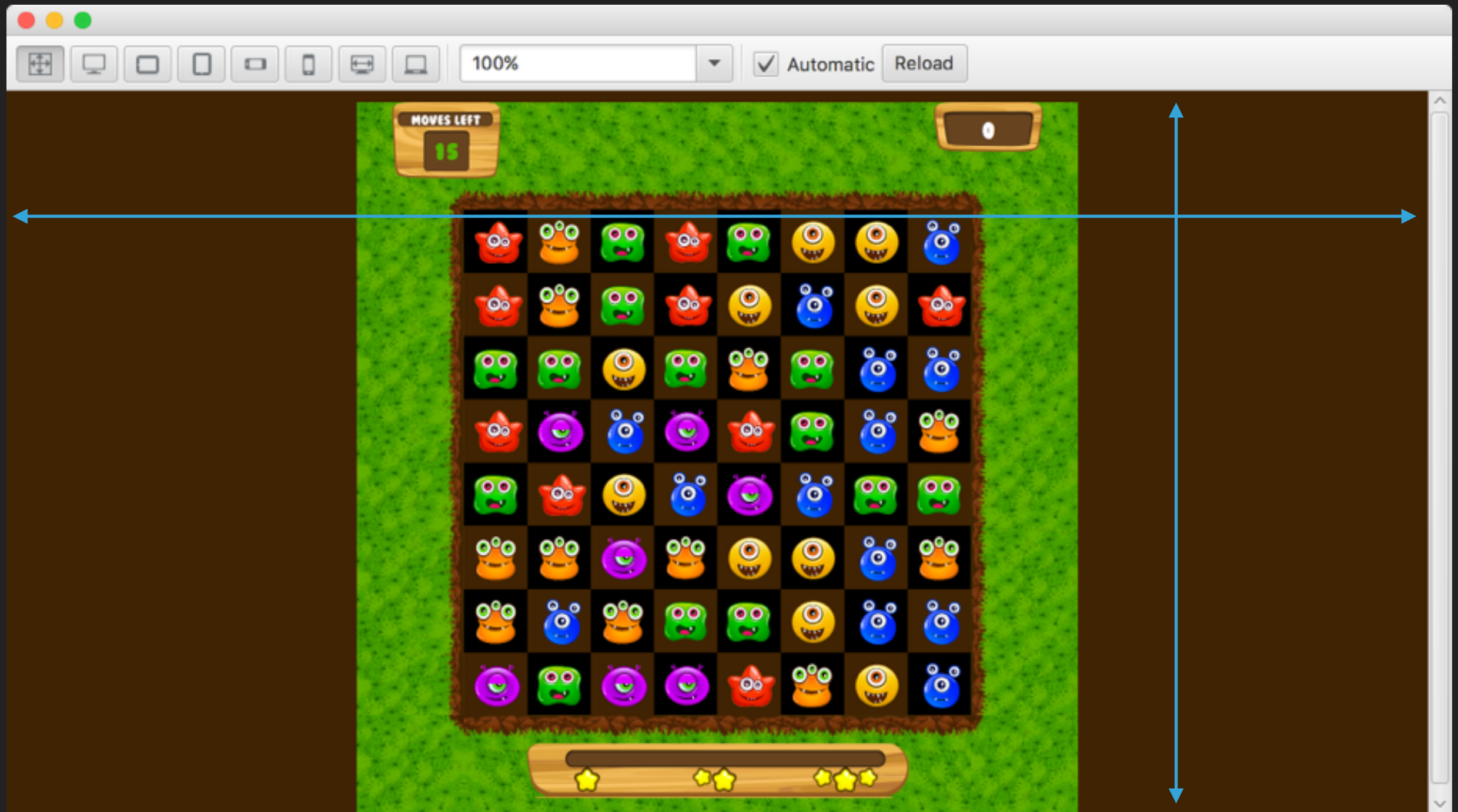
PORTRAIT



LANDSCAPE



LANDSCAPE



```
.tile {  
    height: 8.75vw;  
    width: 8.75vw;  
}
```

view width



```
.tile {
```

```
  height: 8.75vw;
```

```
  width: 8.75vw;
```

```
}
```




<http://www.dukescript.com>



Search



```
@media screen and (orientation: landscape) {
```

```
  .tile {
```

```
    height: 8.75vh;
```

```
    width: 8.75vh;
```

```
  }
```

```
}
```



<http://www.dukescript.com>



Search



← 100vw →

```
@media screen and (orientation: landscape) {
```

```
  .tile {
```

```
    height: 8.75vh;
```

```
    width: 8.75vh;
```

```
  }
```

```
}
```

↑ 100vh ↓



<http://www.dukescript.com>



Search



← 100vw →

```
@media screen and (orientation: landscape) {
```

```
  .tile {
```

```
    height: 8.75vh;
```

```
    width: 8.75vh;
```

```
  }
```

```
}
```

100vh





```
.tile.tile-position-0 {  
  top: 0vw;  
  left: 0vw; }
```

```
.tile.tile-position-1 {  
  top: 0vw;  
  left: 8.75vw; }
```

```
.tile.tile-position-2 {  
  top: 0vw;  
  left: 17.5vw; }
```

```
tile tile-position-3 {
```

```
.tile.tile-position-60 {  
  top: 61.25vw;  
  left: 35vw; }
```

```
.tile.tile-position-61 {  
  top: 61.25vw;  
  left: 43.75vw; }
```

```
.tile.tile-position-62 {  
  top: 61.25vw;  
  left: 52.5vw; }
```

```
.tile.tile-position-63 {  
  top: 61.25vw;  
  left: 61.25vw; }
```

Sass


```
$tile_size: 8.75;
```

```
$tile_size: 8.75;
```

```
@for $i from 0 through 63{
```

```
$tile_size: 8.75;
```

```
@for $i from 0 through 63{
```

```
  // Row
```



```
$tile_size: 8.75;  
@for $i from 0 through 63{  
  // Row  
  $stop: floor($i/8)*$tile_size;
```

```
$tile_size: 8.75;  
@for $i from 0 through 63{  
  // Row  
  $stop: floor($i/8)*$tile_size;  
  // Column
```

```
$tile_size: 8.75;  
@for $i from 0 through 63{  
  // Row  
  $stop: floor($i/8)*$tile_size;  
  // Column  
  $left: floor($i%8)*$tile_size;
```

```
$tile_size: 8.75;  
@for $i from 0 through 63{  
  // Row  
  $stop: floor($i/8)*$tile_size;  
  // Column  
  $left: floor($i%8)*$tile_size;
```

```
$tile_size: 8.75;  
@for $i from 0 through 63{  
  // Row  
  $stop: floor($i/8)*$tile_size;  
  // Column  
  $left: floor($i%8)*$tile_size;  
  
  .tile.tile-position-#{ $i } {
```



```
$tile_size: 8.75;
@for $i from 0 through 63{
  // Row
  $stop: floor($i/8)*$tile_size;
  // Column
  $left: floor($i%8)*$tile_size;

  .tile.tile-position-#{ $i } {
    top: $stop+vw;
```

```
$tile_size: 8.75;
@for $i from 0 through 63{
  // Row
  $top: floor($i/8)*$tile_size;
  // Column
  $left: floor($i%8)*$tile_size;

  .tile.tile-position-#{ $i } {
    top: $top+vw;
    left: $left+vw;
```

```
$tile_size: 8.75;
@for $i from 0 through 63{
  // Row
  $top: floor($i/8)*$tile_size;
  // Column
  $left: floor($i%8)*$tile_size;

  .tile.tile-position-#{ $i } {
    top: $top+vw;
    left: $left+vw;
  }
}
```

```
$tile_size: 8.75;
@for $i from 0 through 63{
  // Row
  $top: floor($i/8)*$tile_size;
  // Column
  $left: floor($i%8)*$tile_size;

  .tile.tile-position-#{ $i } {
    top: $top+vw;
    left: $left+vw;
  }
}
```

```
.tile.tile-position-0 {  
  top: 0vw;  
  left: 0vw;  
}
```

```
.tile.tile-position-1 {  
  top: 0vw;  
  left: 8.75vw;  
}
```

[... 63]



ACHIEVEMENT UNLOCKED!



CSS PRO

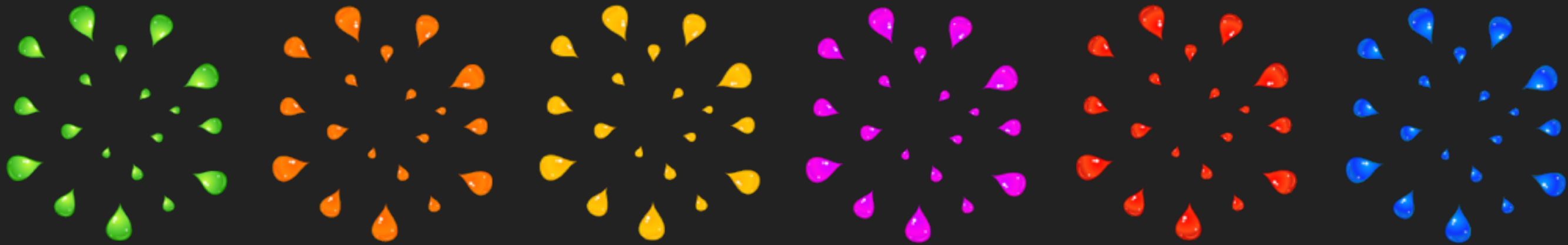




Sprites



Sprites



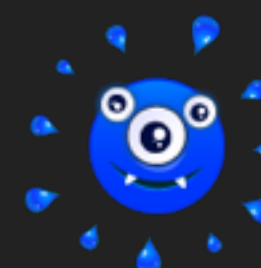
Sprites



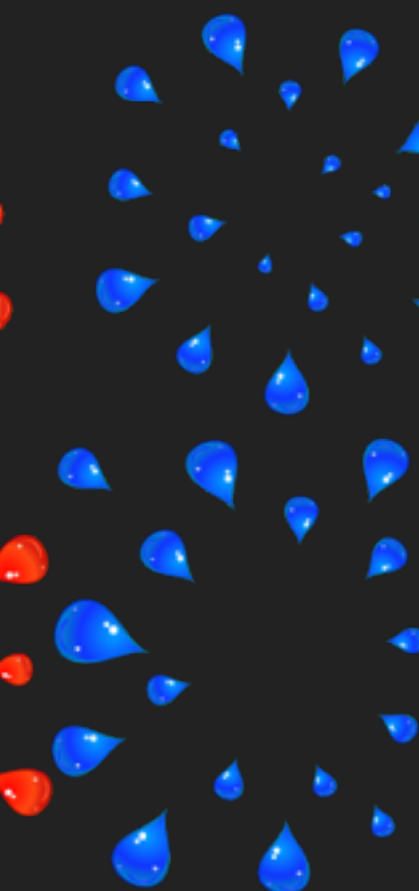
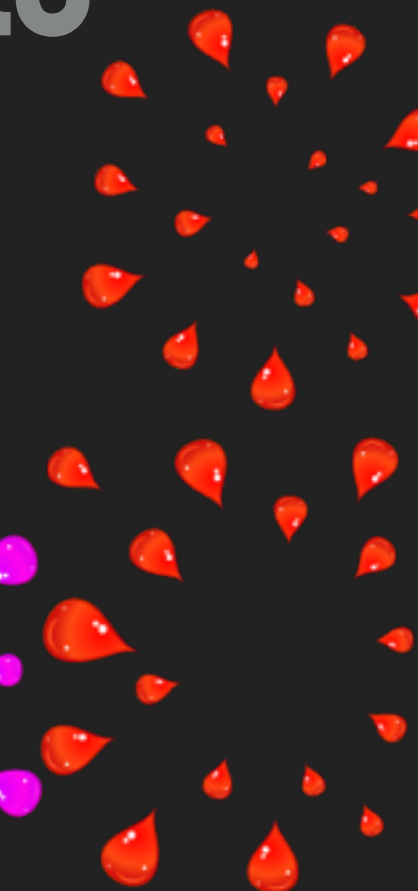
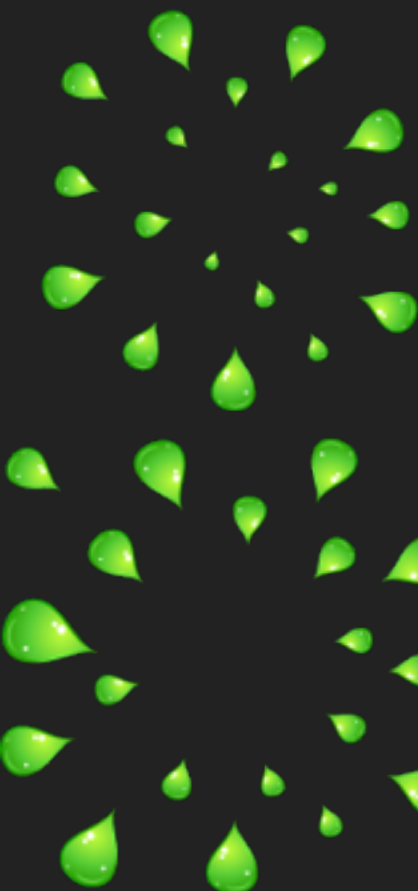
Sprites



Sprites



Spritesheets



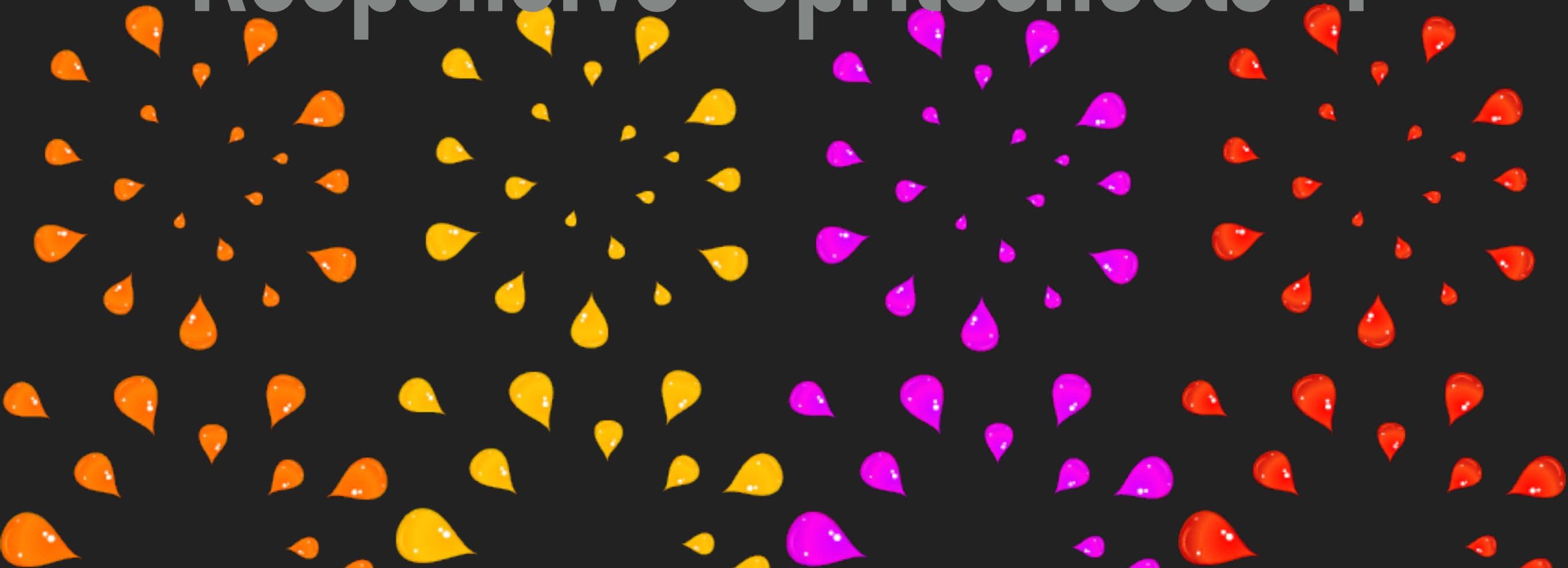


Responsive Spritesheets ?

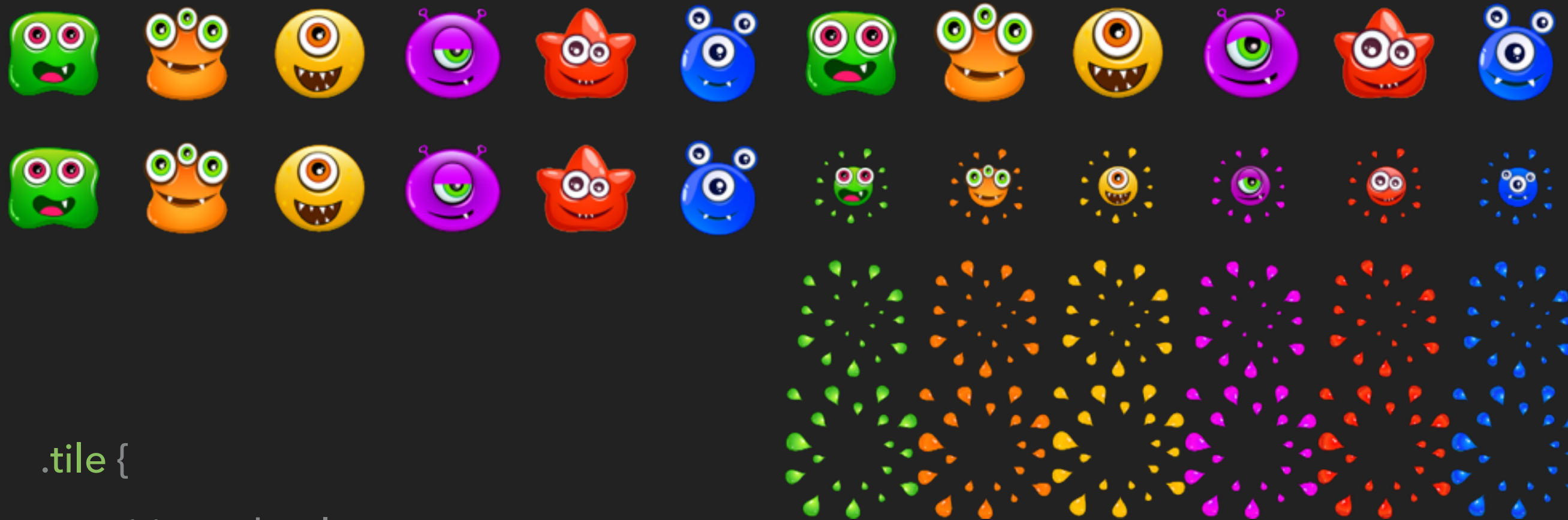




Responsive Spritesheets ?

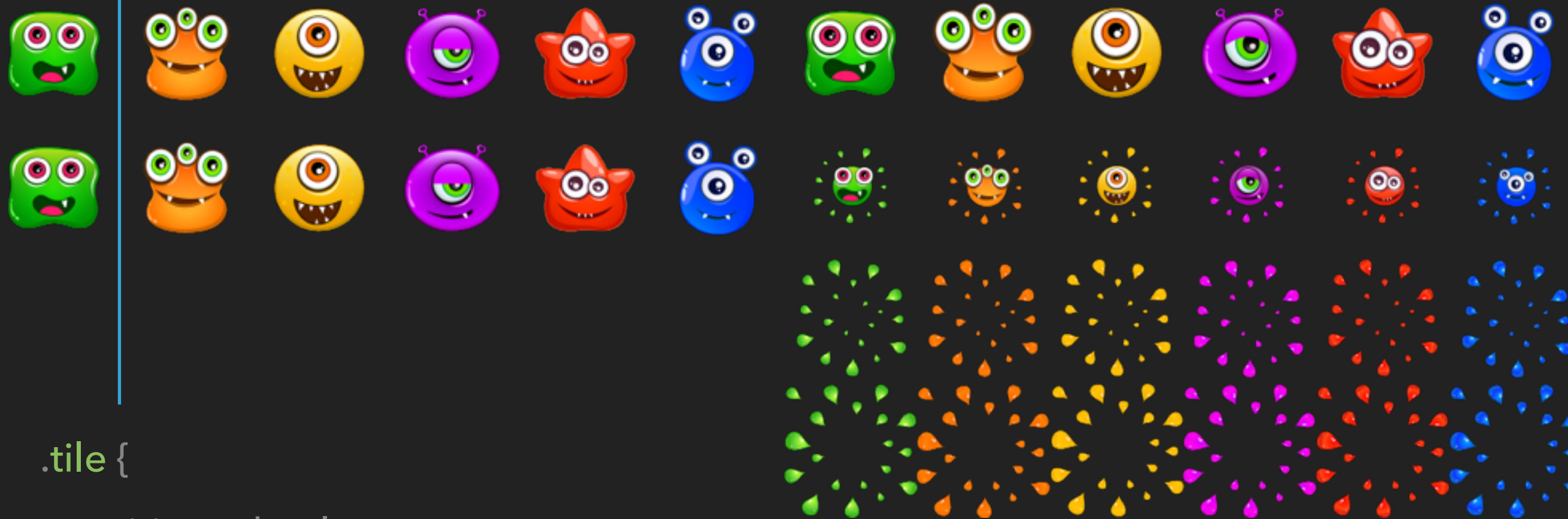


background-size



```
.tile {  
  position: absolute;  
  background: url("../assets/Monster.png");  
  background-size: 1200%;  
}
```

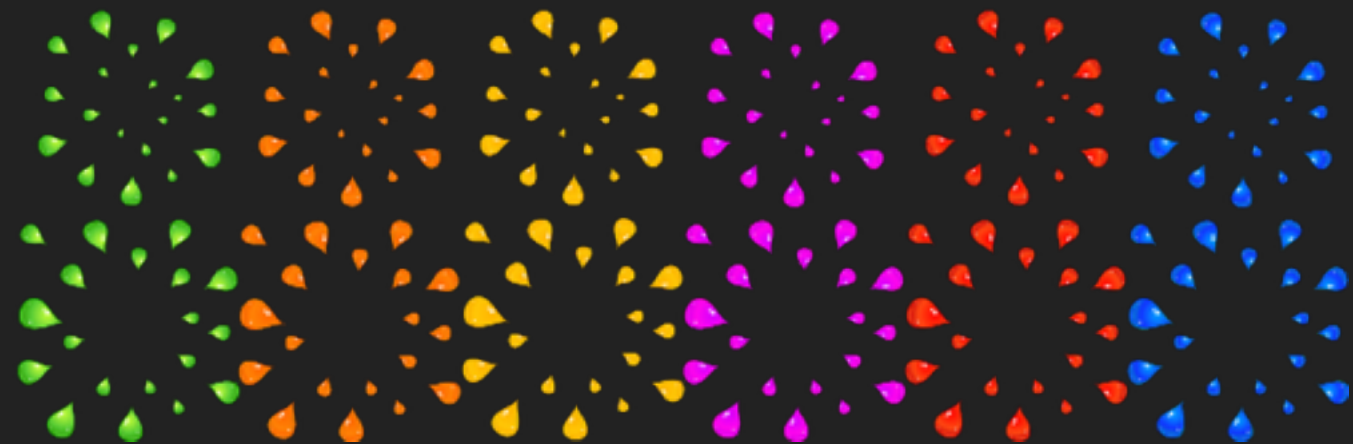
background-size



```
.tile {  
  position: absolute;  
  background: url("../assets/Monster.png");  
  background-size: 1200%;  
}
```

background-size

100%



```
.tile {
```

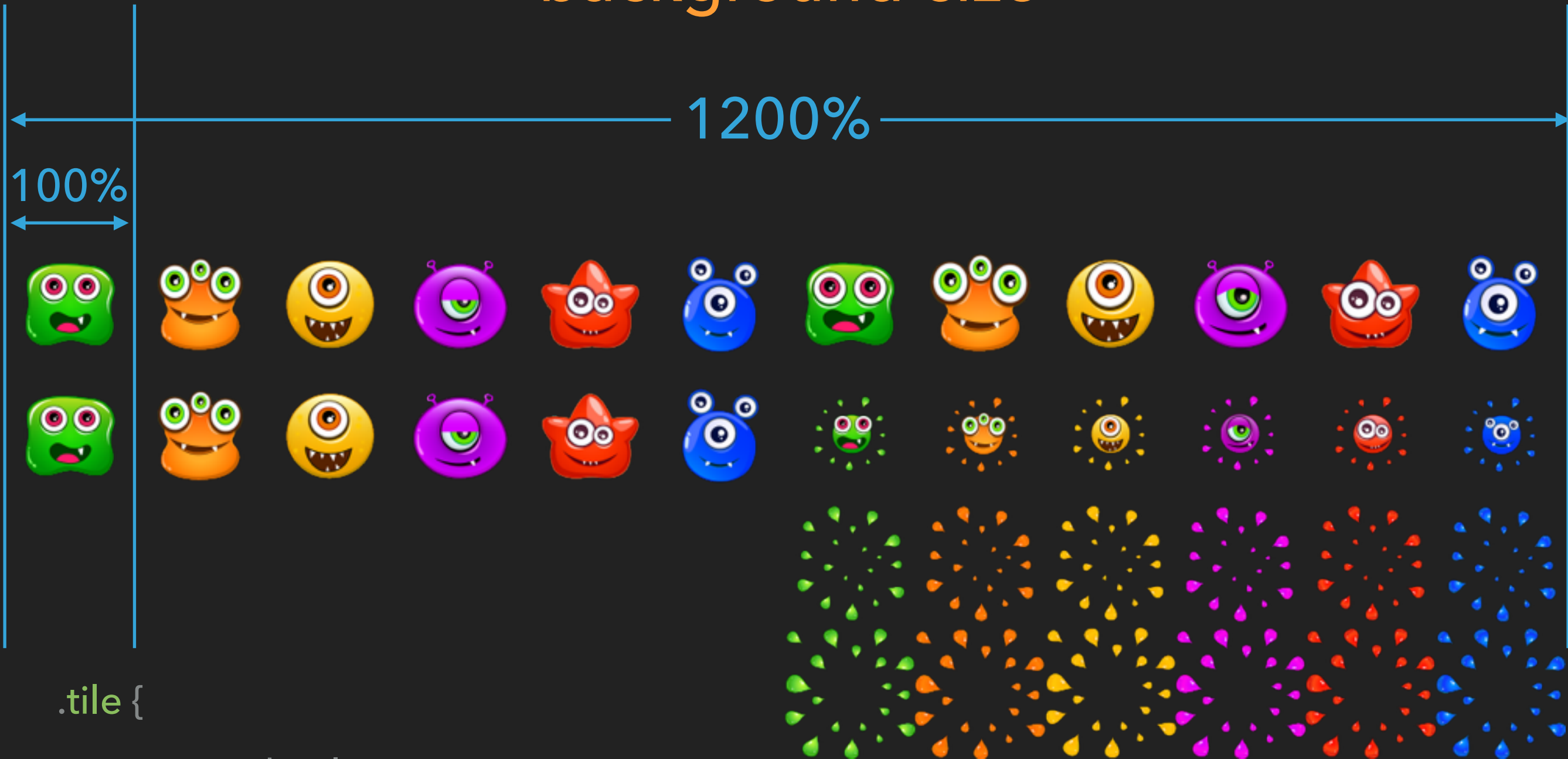
```
  position: absolute;
```

```
  background: url("../assets/Monster.png");
```

```
  background-size: 1200%;
```

```
}
```

background-size



```
.tile {
```

```
  position: absolute;
```

```
  background: url("../assets/Monster.png");
```

```
  background-size: 1200%;
```

```
}
```

```
.tile.tile-type-0 {  
  background-position: 0 0; }
```

```
.tile.tile-type-1 {  
  background-position: 9.09091% 0; }
```

```
.tile.tile-type-2 {  
  background-position: 18.18182% 0; }
```

```
[...]
```


Sass

```
@for $i from 0 through 5{  
  $xPos: $i/(11)*100;  
  .tile.tile-type-#{ $i }{  
    background-position: #{ $xPos }% 0;  
  }  
}
```



CSS Animations

```
@-webkit-keyframes explode {  
    100% { background-position-y: 400% }  
}  
  
#grid .tile.explode{  
    -webkit-animation: explode .4s;  
}
```



CSS Animations

```
@-webkit-keyframes explode {  
    100% { background-position-y: 400% }  
}  
  
#grid .tile.explode{  
    -webkit-animation: explode .4s;  
}
```



Steps

```
@-webkit-keyframes explode {
```

```
  100% { background-position-y: 400% }
```

```
}
```

```
#grid .tile.explode{
```

```
  -webkit-animation: explode .4s steps(4);
```

```
}
```




Steps

```
@-webkit-keyframes explode {
```

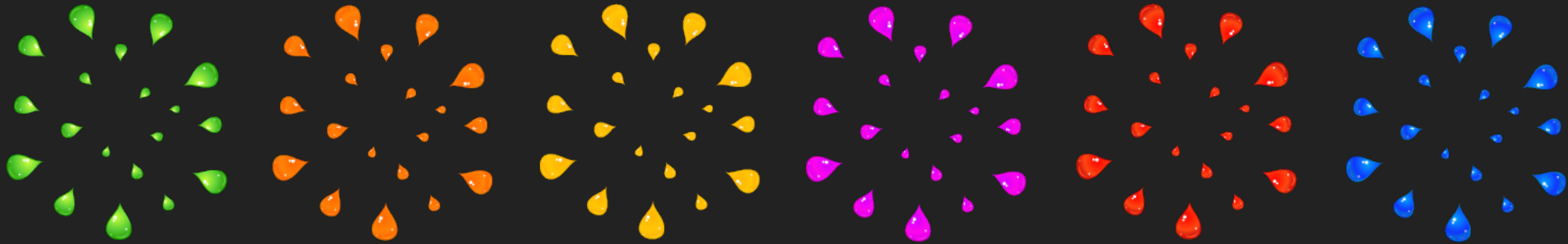
```
  100% { background-position-y: 400% }
```

```
}
```

```
#grid .tile.explode{
```

```
  -webkit-animation: explode .4s steps(4);
```

```
}
```



Steps

```
@-webkit-keyframes explode {
```

```
  100% { background-position-y: 400% }
```

```
}
```

```
#grid .tile.explode{
```

```
  -webkit-animation: explode .4s steps(4);
```

```
}
```



Steps

```
@-webkit-keyframes explode {
```

```
  100% { background-position-y: 400% }
```

```
}
```

```
#grid .tile.explode{
```

```
  -webkit-animation: explode .4s steps(4);
```

```
}
```



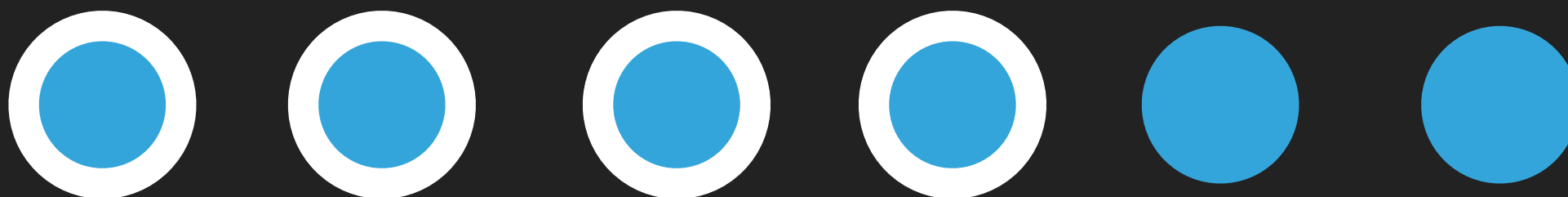
ACHIEVEMENT UNLOCKED!



CSS WIZARD

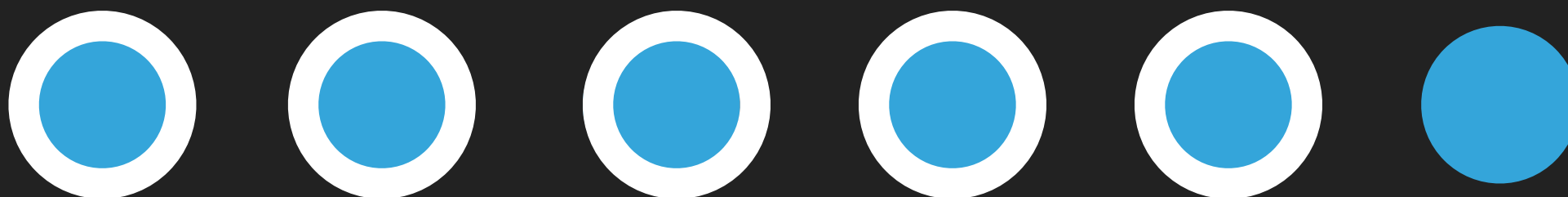


DIRTY TRICKS





DIRTY TRICKS



```
<div id="grid" data-bind="foreach: tiles,  
                           css: {play: play}">  
  <div class="tile"  
    data-bind="moveable: $parent.move,  
               css: 'tile-position-' +  
                   $index() + ' tile-type-' +  
                   type(),  
               swap: swap,  
               drop: drop,  
               explode: explode">  
  
  </div>  
</div>
```

```
css: {play: play}
```

```
#grid.play .tile{  
  -webkit-animation: play 3s steps(2) infinite;  
}
```

```
moveable: $parent.move
```

Knockout.

Custom Bindings

Knockout.

Custom Bindings

Knockout.

`data-bind="<custom-binding>: <property>"`

Knockout.

Custom Bindings

Knockout.

Custom Bindings

Knockout.

```
data-bind="movable: $parent.move"
```



```
ko.bindingHandlers.yourBindingName = {
  init: function(element, valueAccessor, allBindings, viewModel, bindingContext) {
    // This will be called when the binding is first applied to an element
    // Set up any initial state, event handlers, etc. here
  },
  update: function(element, valueAccessor, allBindings, viewModel, bindingContext) {
    // This will be called once when the binding is first applied to an element,
    // and again whenever any observables/computed that are accessed change
    // Update the DOM element based on the supplied values here.
  }
};
```

Javascript

```
ko.bindingHandlers['moveable'] = {  
  init: function (element, valueAccessor, allBindingsAccessor, data,  
context) {  
    var startPosition = {};  
    startPosition.element = element;  
    var handler = valueAccessor().bind(data);  
  
    var mousehandler = function (event) {  
      //...  
    };  
    element.addEventListener('mousedown', mousehandler);  
    element.addEventListener('mouseout', mousehandler);  
    element.addEventListener('touchstart', mousehandler);  
    element.addEventListener('touchmove', mousehandler);  
  }  
};
```

Javascript

```
//...
if (mouseout) {
    var deltaX = startPosition.x - newX;
    var deltaY = startPosition.y - newY;

    if (Math.abs(deltaX) > Math.abs(deltaY) && deltaX > 0) {
        event.dir = "left";
        handler(data, event);
    } else if (Math.abs(deltaX) > Math.abs(deltaY) && deltaX < 0) {
        event.dir = "right";
        handler(data, event);
    } else if (Math.abs(deltaY) > Math.abs(deltaX) && deltaY > 0) {
        event.dir = "up";
        handler(data, event);
    } else if (Math.abs(deltaY) > Math.abs(deltaX) && deltaY < 0) {
        event.dir = "down";
        handler(data, event);
    }
}
//...
}
```

Java

@Function

```
public static void move(Level level, Tile data, String dir, int pos) {  
    List<Tile> tiles = level.getTiles();  
    int newPos = -1;  
    if (dir.equals("up") && rowNumber(level, pos) > 0) {  
        newPos = pos - 8;  
    } else if (dir.equals("down") && rowNumber(level, pos) < 7) {  
        newPos = pos + 8;  
    } else if (dir.equals("left") && columnNumber(level, pos) > 0) {  
        newPos = pos - 1;  
    } else if (dir.equals("right") && columnNumber(level, pos) < 7) {  
        newPos = pos + 1;  
    }  
}
```

Animations and Functions

Remove after Animation

```
@-webkit-keyframes explode {  
    100% { background-position-y: 400% }  
}  
  
#grid .tile.explode{  
    -webkit-animation: explode .4s steps(4);  
}
```

Remove after Animation

```
@-webkit-keyframes explode {  
    100% { background-position-y: 400% }  
}  
  
#grid .tile.explode{  
    -webkit-animation: explode .4s steps(4);  
}
```

???

explode: explode

```
ko.bindingHandlers['explode'] = {  
  update: function (element, valueAccessor, allBindings, data, bindingContext) {  
    var value = valueAccessor();  
    var tile = data;  
    var parent = bindingContext.$parent;  
    var remove = function () {  
      // ...  
    };  
    var explode = ko.unwrap(value);  
    if (explode) {  
      $(element).addClass("explode");  
      setTimeout(remove, 210);  
    }  
  }  
};
```

explode: explode

```
var remove = function () {  
  tile.type(-1);  
  $(element).removeClass("explode");  
  value(false);  
  parent.remove();  
};
```

Java

```
@Function
public static void remove(Level l) {
    l.setDeleted(l.getDeleted() - 1);
    if (l.getDeleted() == 0) {
        fillUpTiles(l, l.getTiles());
    }
}
```




ACHIEVEMENT UNLOCKED!



PUPPET MASTER





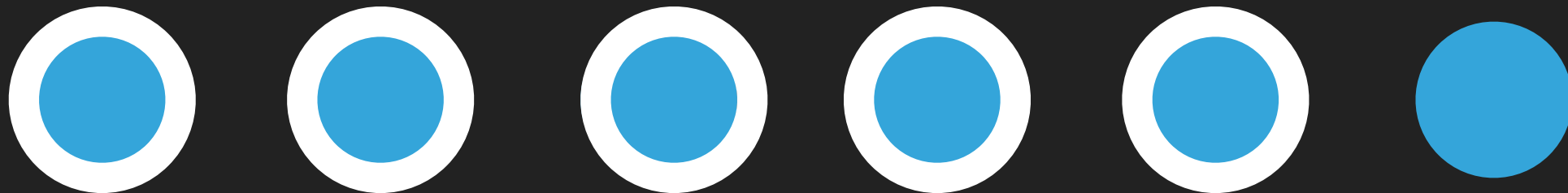
ACHIEVEMENT UNLOCKED!



PUPPET MASTER

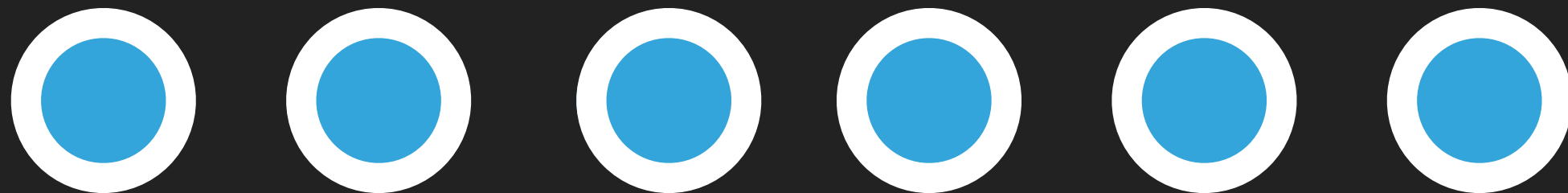


APP STORES





APP STORES



GOOGLE

- ▶ Developer Account:

- ▶ <https://play.google.com/apps/publish/signup/>

- ▶ 25\$ (single payment)

- ▶ Signing

- ▶ <http://developer.android.com/tools/publishing/app-signing.html#signing-manually>

- ▶ `keytool -genkey -v -keystore my-release-key.keystore`

- `-alias alias_name -keyalg RSA -keysize 2048`

- `-validity 10000`

GOOGLE

Use Maven jarsigner to sign, or sign manually:

```
jarsigner -verbose -sigalg SHA1withRSA
```

```
-digestalg SHA1 -keystore ../ks.keystore
```

```
match3-android-1.0-SNAPSHOT.apk alias
```

GOOGLE

Zipalign:

```
zipalign -v 4 match3-android-1.0-SNAPSHOT.apk  
match3.apk
```

Upload

fill out forms

upload images

published after 1 day

APPLE

- ▶ iTunes Developer Account:
 - ▶ <https://developer.apple.com>
 - ▶ 99\$ (annual)
 - ▶ Mac with XCode

APPLE

- ▶ Info.plist.xml
- ▶ Configure in RoboVM
- ▶ https://developer.apple.com/library/ios/documentation/iPhone/Conceptual/iPhoneOSProgrammingGuide/ExpectedAppBehaviors/ExpectedAppBehaviors.html#//apple_ref/doc/uid/TP40007072-CH3-SW9

APPLE

- ▶ Create App Record in iTunes Connect
- ▶ Create Signing identity and provisioning file using XCode
- ▶ <https://developer.apple.com/library/ios/documentation/IDEs/Conceptual/AppDistributionGuide/SubmittingYourApp/SubmittingYourApp.html>
- ▶ Add reference to iosProvisioningProfile in pom.xml
- ▶ RoboVM create IPA

Upload

fill out forms

upload (many) images

Rejected!

Don't mention you're cross-platform...

The term "Android" in your description
doesn't help

The terms "beta", "demo", or "test"
don't help either

Some complaints are hard to understand
"Apps with placeholder text will be rejected"

APPLE

But after a couple of weeks

- ▶ <http://wiki.apidesign.org/wiki/AppStore>
- ▶ <https://itunes.apple.com/us/app/fair-minesweeper/id903688146?mt=8>

VICTORY

YOU WON!



17190

SHARE WITH YOUR FRIENDS



FRIENDS RANKING



12587



14256



16985





CAST (IN ORDER OF APPEARANCE)

CAST (IN ORDER OF APPEARANCE)

www.dukescript.com

www.knockoutjs.com

www.netbeans.org

www.sass-lang.com

www.jquery.com

www.jqueryui.com

<https://github.com/dukescript/match3.git>

GAME OVER!



THANK YOU!