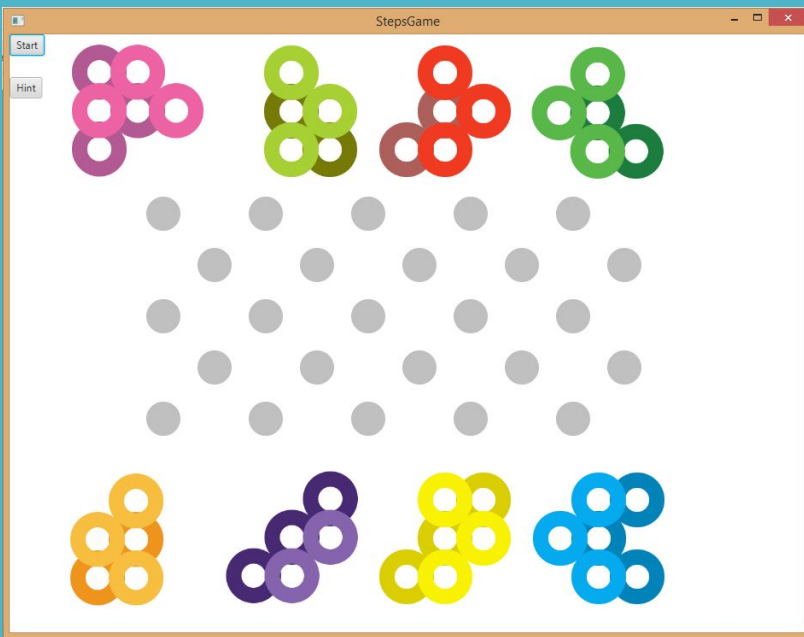


IQSteps

By Ziwei Liu, Tingyu Pan,
Taylor-James Thomas



Design Approach

- Easy to translate graphical interface into logic data
- All graphical objects contain their own information
- Logical components as conditions for the modification of graphical elements

Problems

- Logic: Representing pieces and checking for validity of placements
- Integration: Combining UI and logic components

Logic – The board

`Int[5][10]`

0 – open for placement

1 – occupied

Logic - Pieces



1	1	1
1	1	0
1	1	1

Logic – Check 1, occupied?



x-1, y+1	x, y+1	x+1, y+1
x-1, y	x , y	x+1, y
x-1, y-1	x, y-1	x+1, y-1

Logic – Check 2, update the board



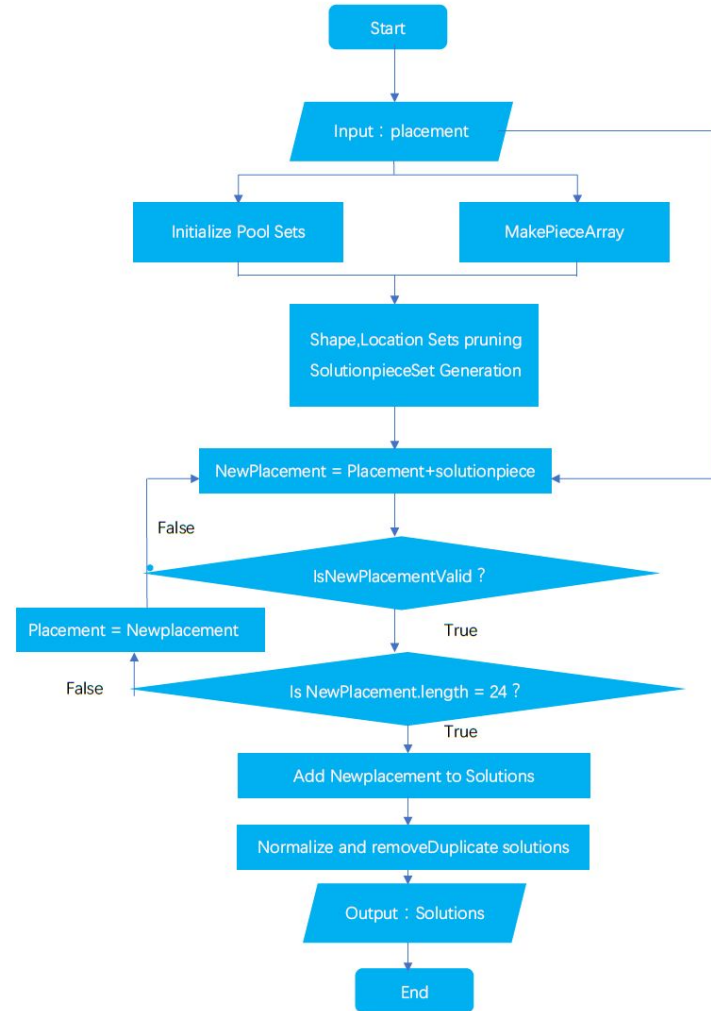
1	1	1
1	1	1
1	1	1

Logic – Check 3, surroundings



0	0	1	0	0
0	1	1	1	0
1	1	1	0	0
0	1	1	1	0
0	0	1	0	0

Logic - Task 9



Integration – UI and logic

- Generate values from the UI which can be interpreted by logic components
- Store values in the objects which generated them for ease of identification and access

Integration – UI and logic



EA?

Placements on the board?

Integration – UI and logic



EA

Instance fields:

Char piece = 'E';

String pieceStr = piece + 'A';

Integration – UI and logic



piece = 'E'

pieceStr = "EAb"



piece = 'B'

pieceStr = "BAJ"



piece = 'C'

pieceStr = "CAk"

Integration – UI and logic



piece = 'E'

pieceStr = "EAb"



piece = 'B'

pieceStr = "BAJ"



piece = 'C'

pieceStr = "CAk"

"EAbBAJCAk"

Features - Rotation



A-D

setRotate()



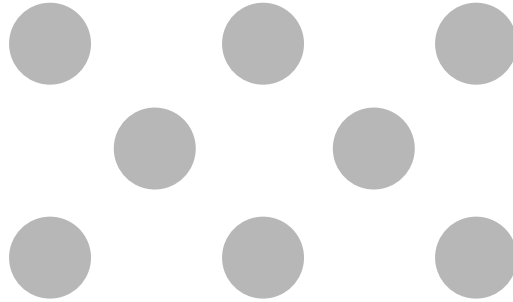
setImage()



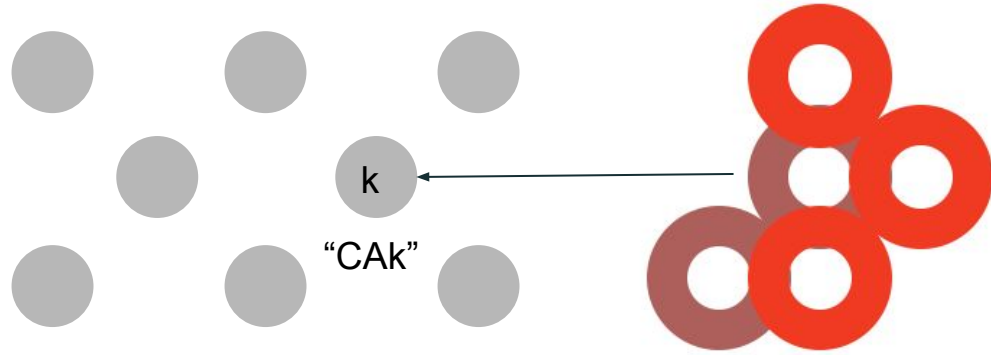
E-H

setRotate()

Features - Snapping



Features - Snapping



```
String testPlacements = currentPlacements + "CAk";
```

```
isPlacementSequenceValid(testPlacements);
```

Features - Hints

```
String sols[] = getSolutions(currentPlacements);
```



```
Set<String> place = StepsGame.getViablePiecePlacements(currentPlacements, sols[0]);
```



Find the piece object with the same “piece” field as the first letter in one of the strings in “place”, play animation on that piece