Comp 6710 Presentation

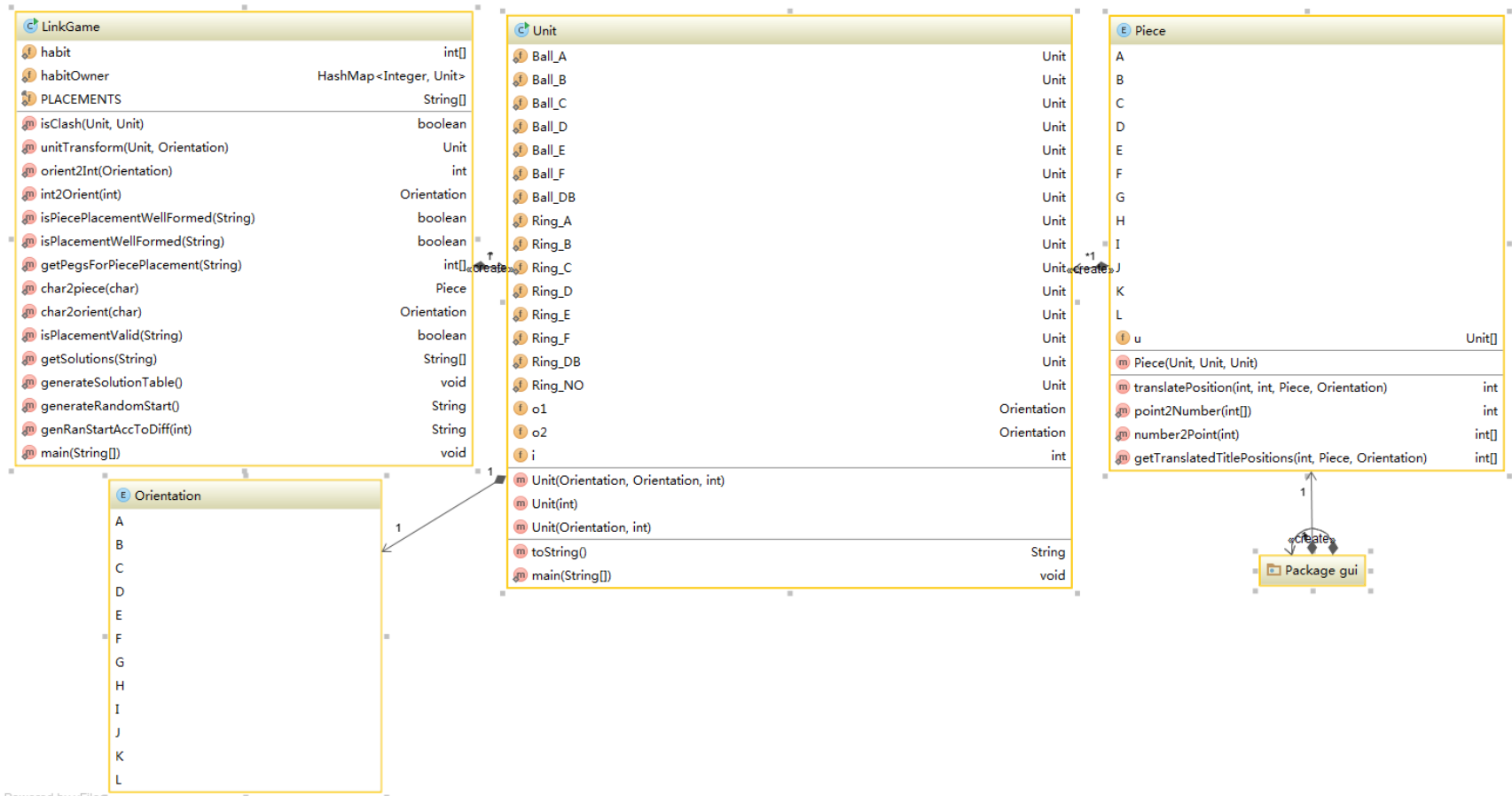## Group 11ThuH

Honggu Lin U6135394

Boya Na U5918786

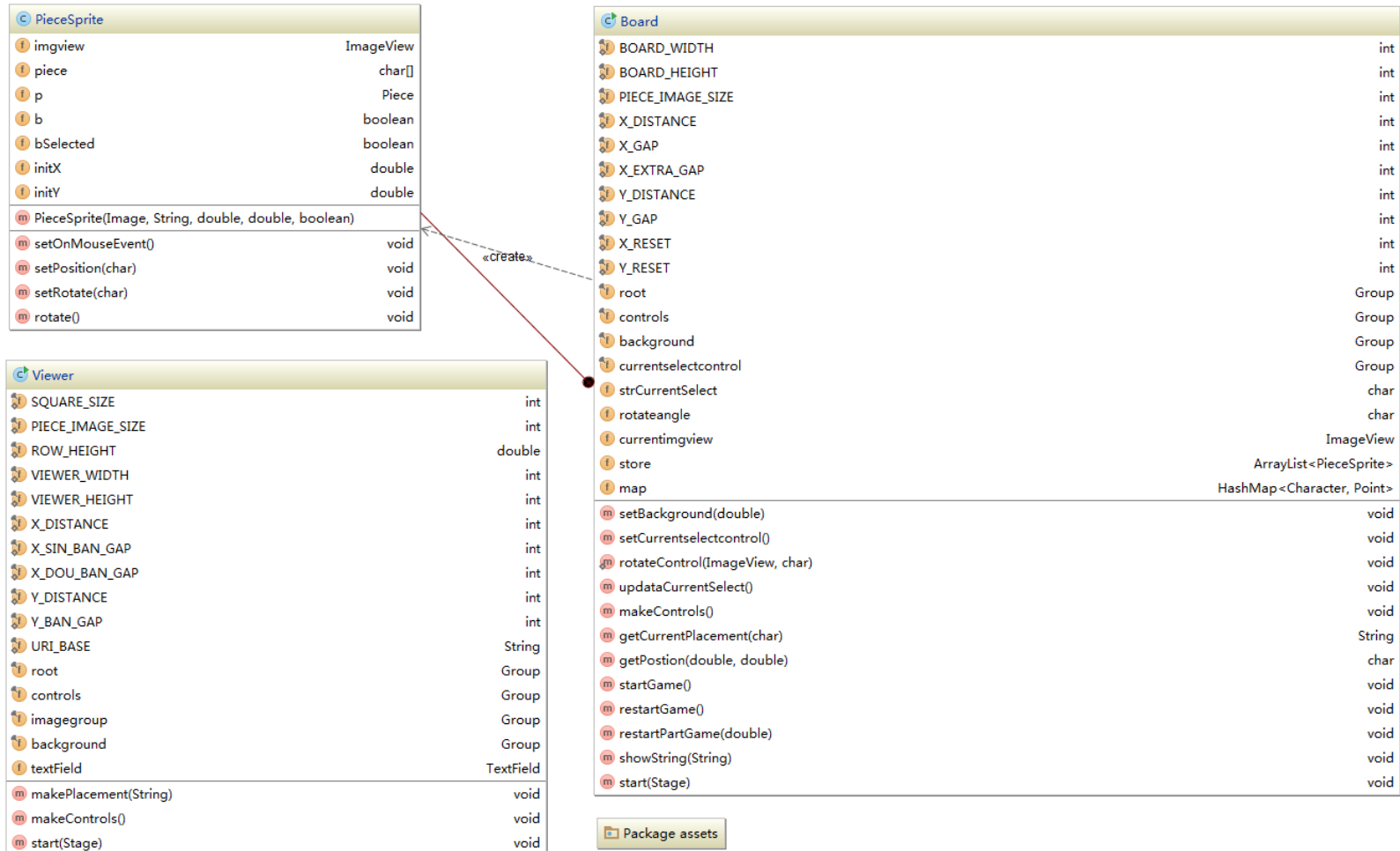**Yuxi Chen U6106850**

Introduction

- **Analyze requirement of development**

- **Design model using UML**

- **Design tests**

- **Implement model and control layer**

- **Implement visual layer by JavaFX**

- **Debug and advance user experience**

Class Diagram

LinkGame

habit	int[]
habitOwner	HashMap<Integer, Unit>
PLACEMENTS	String[]
isClash(Unit, Unit)	boolean
unitTransform(Unit, Orientation)	Unit
orient2Int(Orientation)	int
int2Orient(int)	Orientation
isPiecePlacementWellFormed(String)	boolean
isPlacementWellFormed(String)	boolean
getPegsForPiecePlacement(String)	int[]
char2piece(char)	Piece
char2orient(char)	Orientation
isPlacementValid(String)	boolean
getSolutions(String)	String[]
generateSolutionTable()	void
generateRandomStart()	String
genRanStartAccToDiff(int)	String
main(String[])	void

Orientation

A
B
C
D
E
F
G
H
I
J
K
L

Unit

Ball_A	Unit
Ball_B	Unit
Ball_C	Unit
Ball_D	Unit
Ball_E	Unit
Ball_F	Unit
Ball_DB	Unit
Ring_A	Unit
Ring_B	Unit
Ring_C	Unit
Ring_D	Unit
Ring_E	Unit
Ring_F	Unit
Ring_DB	Unit
Ring_NO	Unit
o1	Orientation
o2	Orientation
i	int
Unit(Orientation, Orientation, int)	
Unit(int)	
Unit(Orientation, int)	
toString()	String
main(String[])	void

Piece

A
B
C
D
E
F
G
H
I
J
K
L

u	Unit[]
Piece(Unit, Unit, Unit)	
translatePosition(int, int, Piece, Orientation)	int
point2Number(int[])	int
number2Point(int)	int[]
getTranslatedTitlePositions(int, Piece, Orientation)	int[]

Package gui

«create»

Powered by yFiles

Class Diagram

PieceSprite

f imgview	ImageView
f piece	char[]
f p	Piece
f b	boolean
f bSelected	boolean
f initX	double
f initY	double
m PieceSprite(Image, String, double, double, boolean)	
m setOnMouseEvent()	void
m setPosition(char)	void
m setRotate(char)	void
m rotate()	void

«create»

Board

f BOARD_WIDTH	int
f BOARD_HEIGHT	int
f PIECE_IMAGE_SIZE	int
f X_DISTANCE	int
f X_GAP	int
f X_EXTRA_GAP	int
f Y_DISTANCE	int
f Y_GAP	int
f X_RESET	int
f Y_RESET	int
f root	Group
f controls	Group
f background	Group
f currentselectcontrol	Group
f strCurrentSelect	char
f rotateangle	char
f currentimgview	ImageView
f store	ArrayList<PieceSprite>
f map	HashMap<Character, Point>
m setBackground(double)	void
m setCurrentselectcontrol()	void
m rotateControl(ImageView, char)	void
m updataCurrentSelect()	void
m makeControls()	void
m getCurrentPlacement(char)	String
m getPostion(double, double)	char
m startGame()	void
m restartGame()	void
m restartPartGame(double)	void
m showString(String)	void
m start(Stage)	void

Viewer

f SQUARE_SIZE	int
f PIECE_IMAGE_SIZE	int
f ROW_HEIGHT	double
f VIEWER_WIDTH	int
f VIEWER_HEIGHT	int
f X_DISTANCE	int
f X_SIN_BAN_GAP	int
f X_DOU_BAN_GAP	int
f Y_DISTANCE	int
f Y_BAN_GAP	int
f URI_BASE	String
f root	Group
f controls	Group
f imagegroup	Group
f background	Group
f textField	TextField
m makePlacement(String)	void
m makeControls()	void
m start(Stage)	void

Powered by yFiles

Package assets

MODEL ------ Data Structure

- **Piece (A,B,C,D,E,F,G,H,I,J,K,L)**

- **Unit**

  - **Ball (A,B,C,D,F,DB)**

  - **Ring (A,B,C,D,E,F,DB,NO)**

- **Orientation (A,B,C,D,E,F,G,H,I,J,K,L)**

MODEL ------ Algorithm

- **getSolutions**

    using Depth First Search

    to find the solutions of the game

CONTROL ----- Conversion

- **unitTransform**

- **orient2Int/int2Orient**

- **char2piece/char2orient**

- **translatePosition**

- **point2Number/number2Point**

- **getTranslatedTitlePositions**

CONTROL ----- Validity Check

- **isPiecePlacementWellFormed**

- **isPlacementWellFormed**

- **getPegsForPiecePlacement**

- **isPlacementValid**

- **isClash**

VISUAL
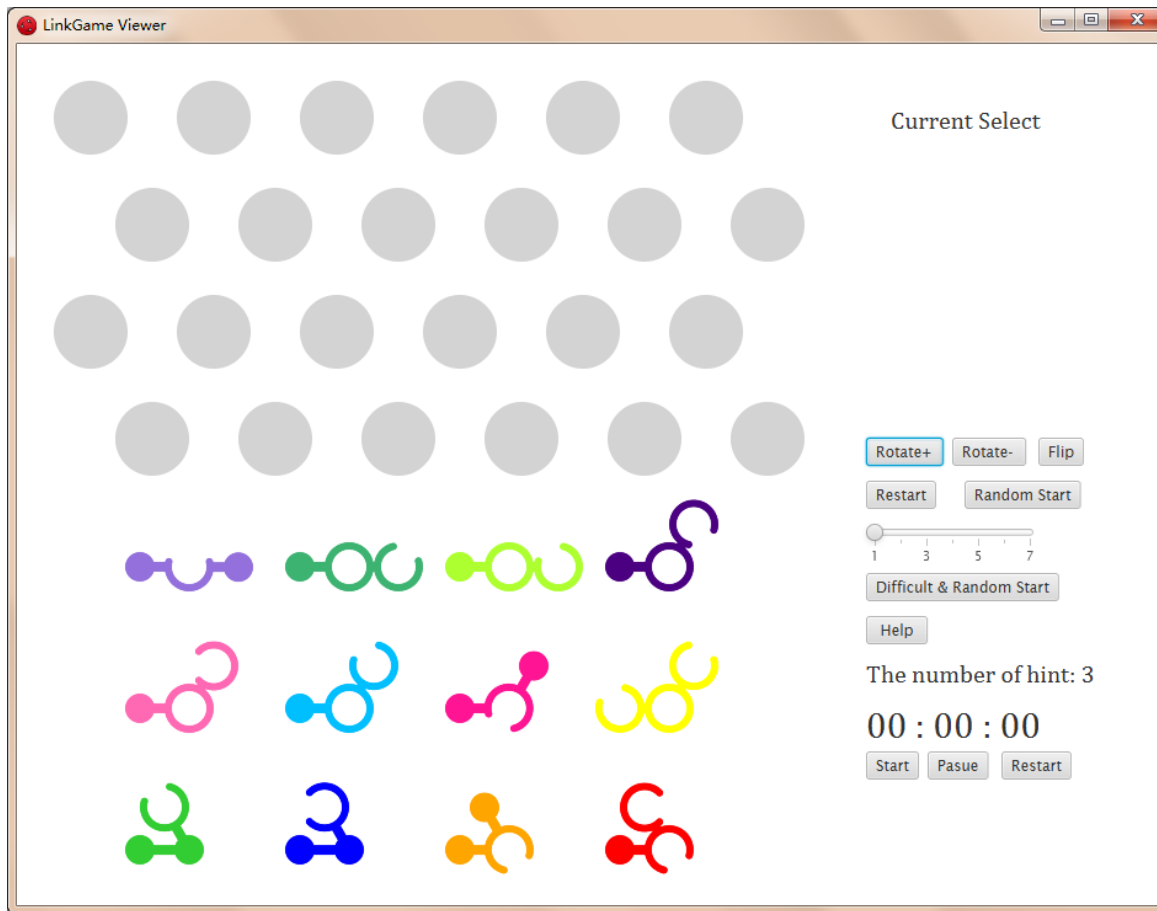
Button Rotate+/-, Flip：Control the orientation.

Button Restart: Restart the game.

Button Random Start: Start Randomly.

Button Diff&Ran Start: Start due to difficulty setting in above line.

VISUAL
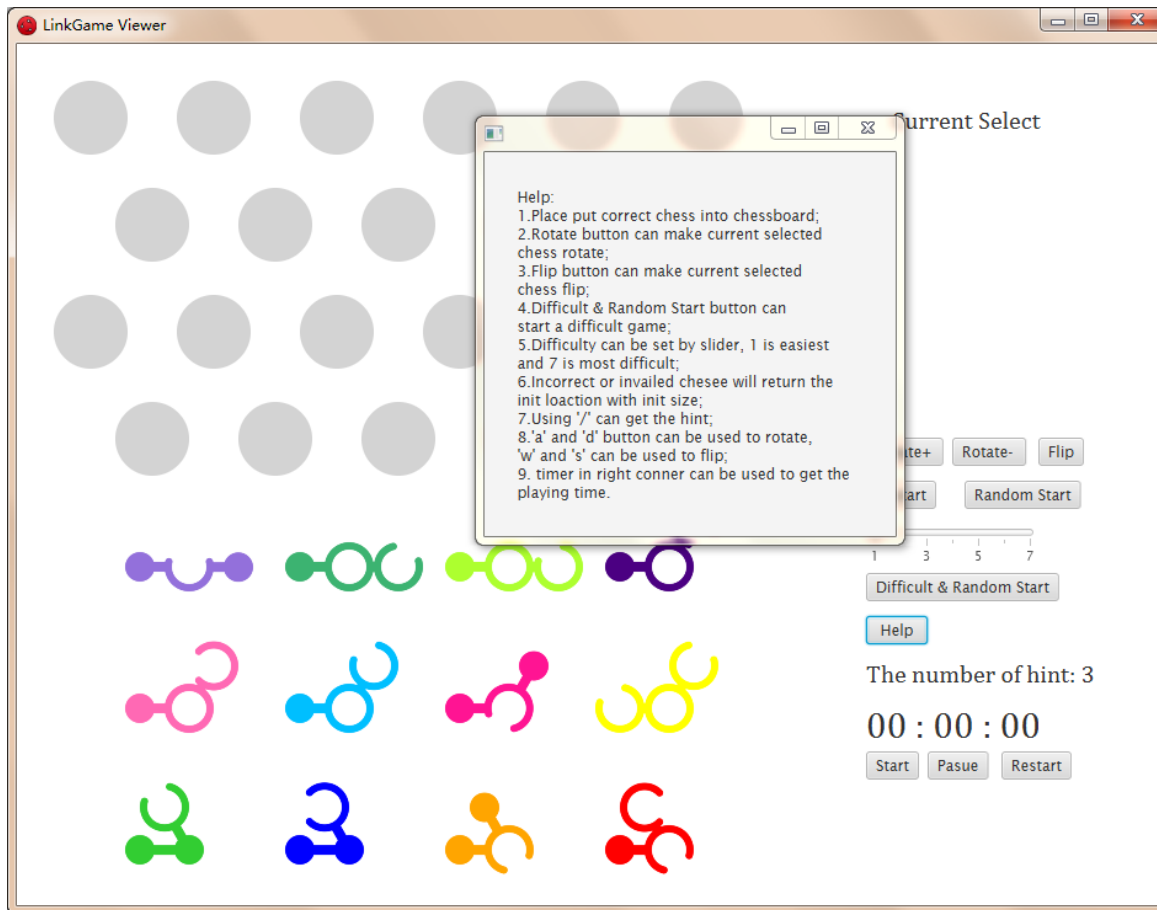
Button Help:
Show Help

We give out three hints, and you can see the number of remained ones.

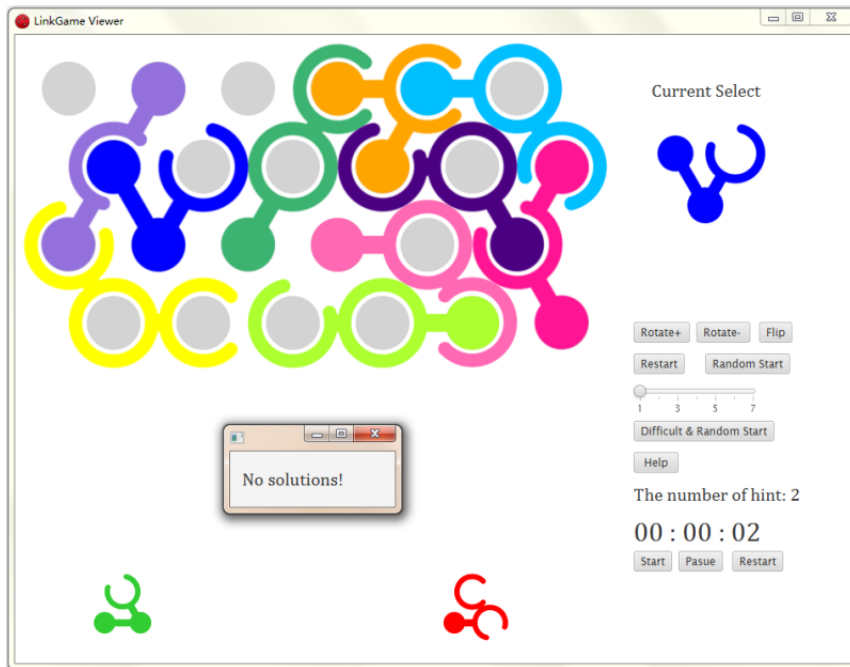**At left bottom there is a timer for helping accounting the time. You can pause at any time you want.**

LinkGame Viewer
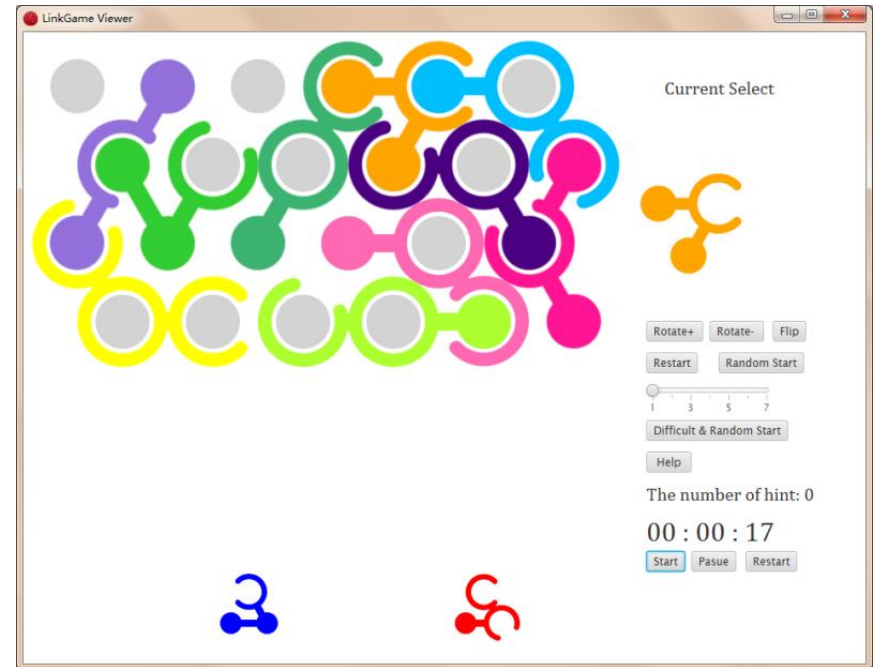
Current Select

Help:
1.Place put correct chess into chessboard;
2.Rotate button can make current selected chess rotate;
3.Flip button can make current selected chess flip;
4.Difficult & Random Start button can start a difficult game;
5.Difficulty can be set by slider, 1 is easiest and 7 is most difficult;
6.Incorrect or invailed chesee will return the init loaction with init size;
7.Using '/' can get the hint;
8.'a' and 'd' button can be used to rotate, 'w' and 's' can be used to flip;
9. timer in right conner can be used to get the playing time.

te+   Rotate-   Flip

art   Random Start

1   3   5   7

Difficult & Random Start

Help

The number of hint: 3

00 : 00 : 00

Start   Pasue   Restart

VISUAL

Start

Rotate

VISUAL

Use Hint after a Wrong move

Use out 3 Hints

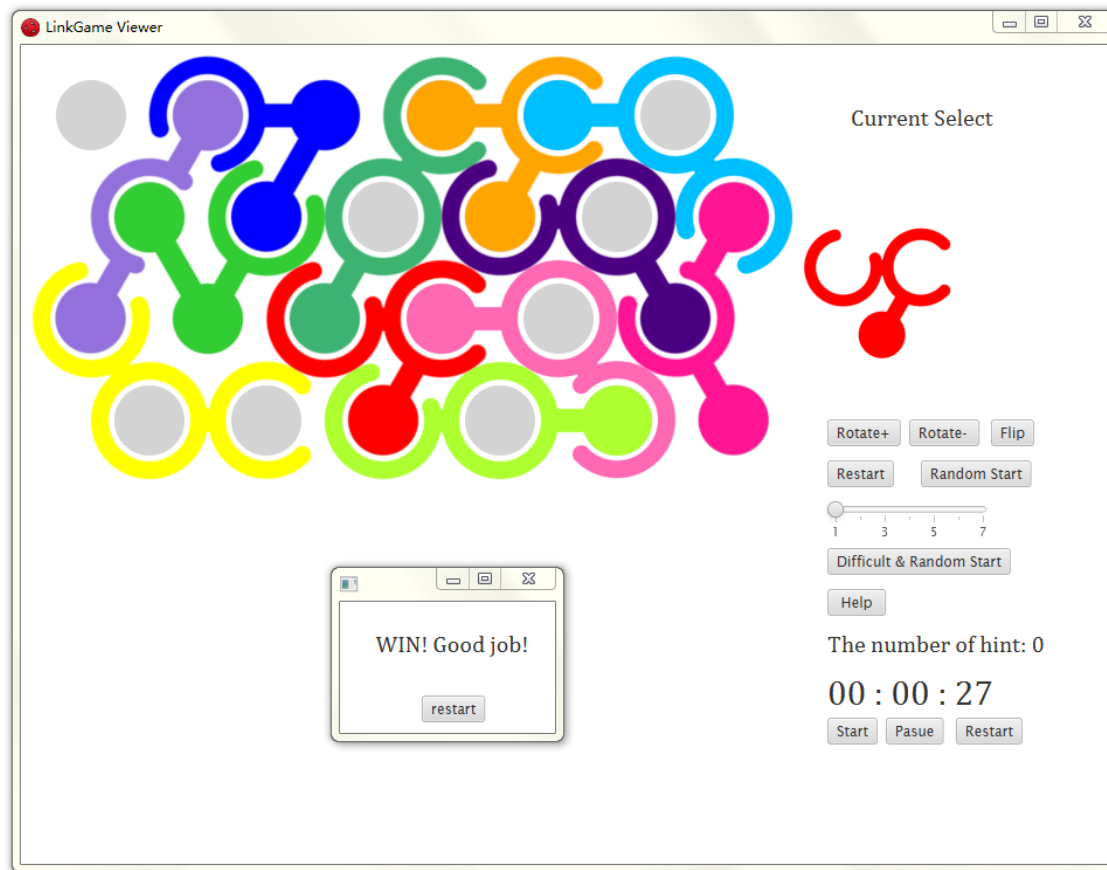# VISUAL

Succeed in Solving!

More Features

- **Hint (by step)**

- **Icon**

- **Timer**

- **Winning Notice**

Thank you