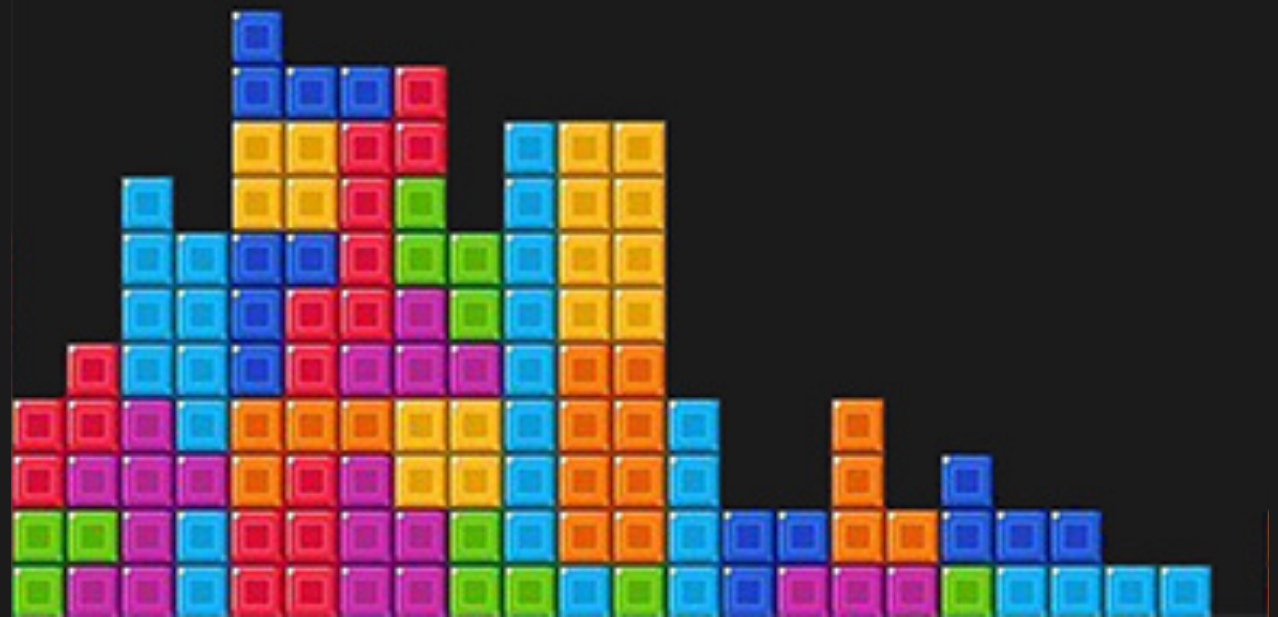
Arcade Games

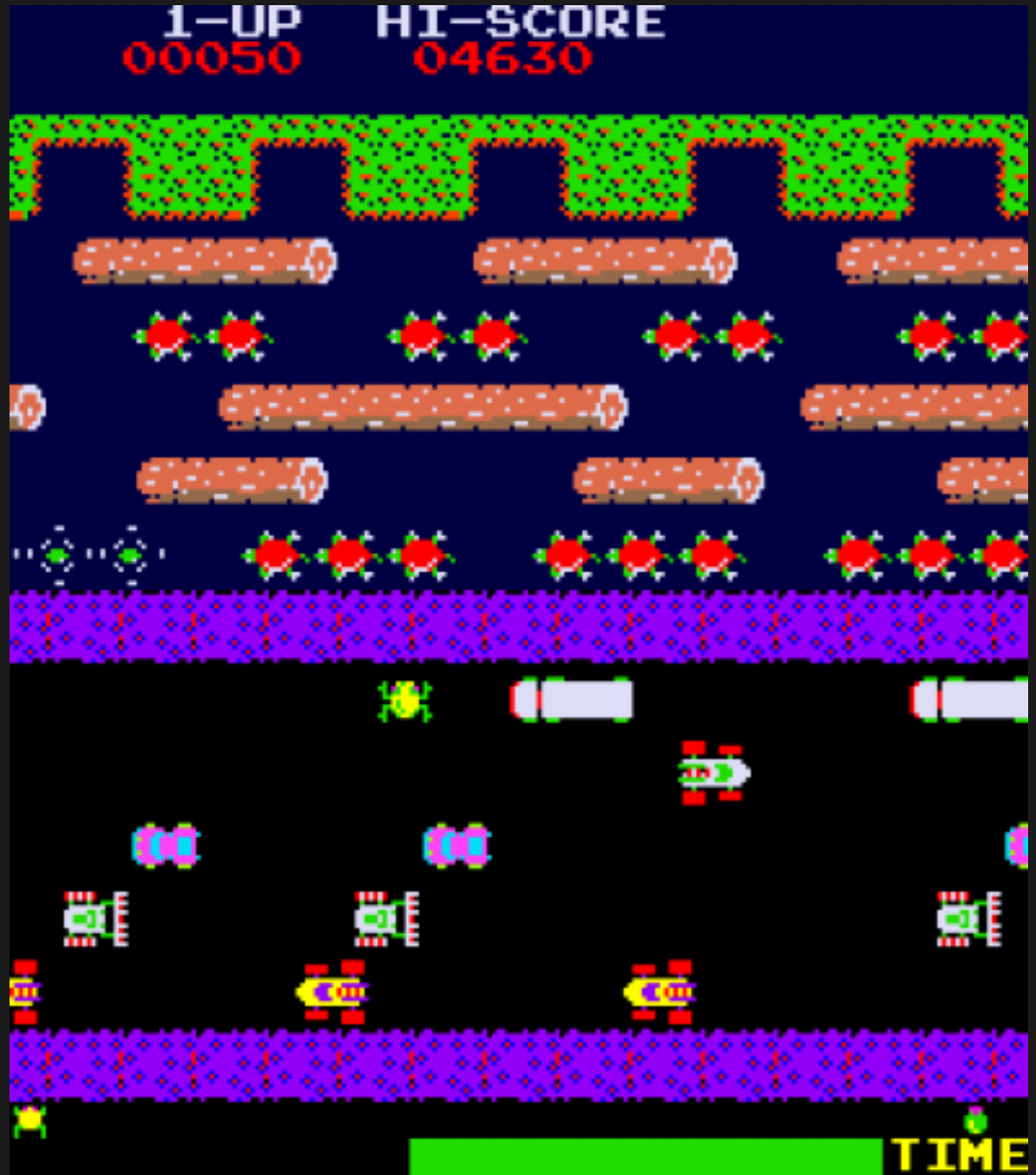
Miffy Chen

CS 622 Adv. Java

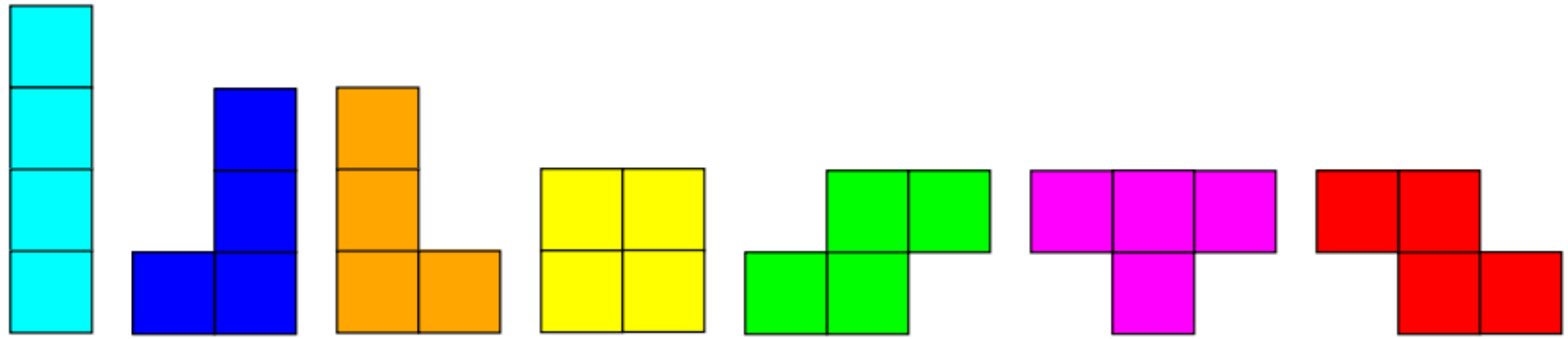
Games

- Tetris (Java Swing)
- Frogger (JavaFX)

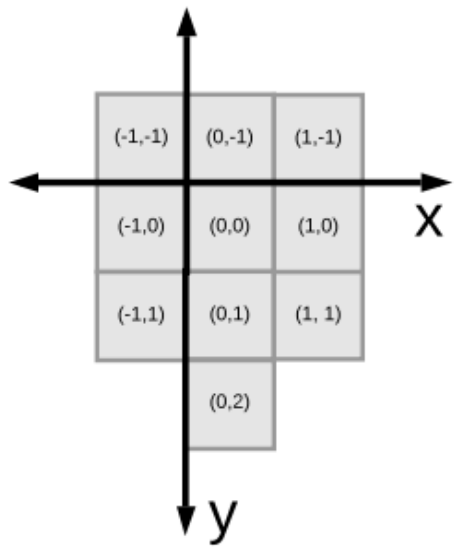




public enum Tetrominoes



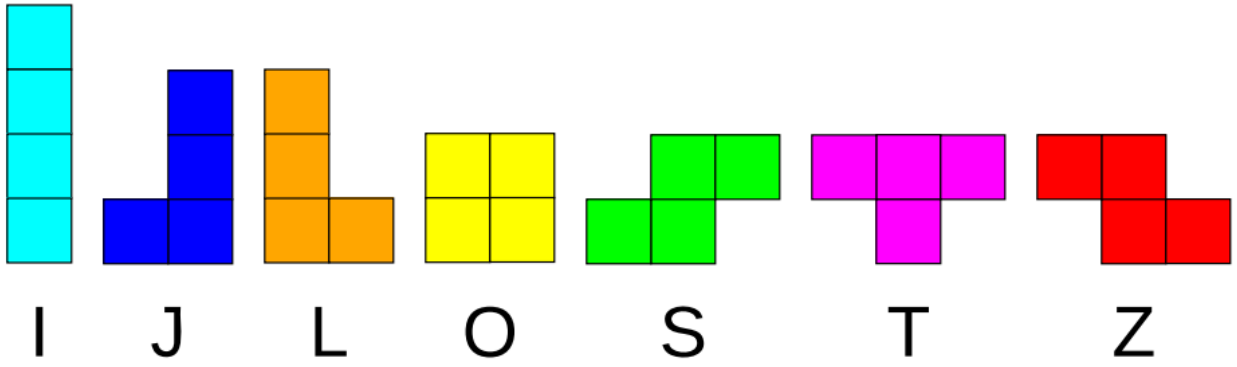

```
Tetrominoes [nameOfShape]Shape = NoShape, IShape,
        JShape, LShape, OShape, SShape, TShape, ZShape;

int[][] coords = new int[][] { {x1, y1}, {x2, y2}, {x3, y3}, {x4, y4} };

Color color = new Color (r, g, b);
```

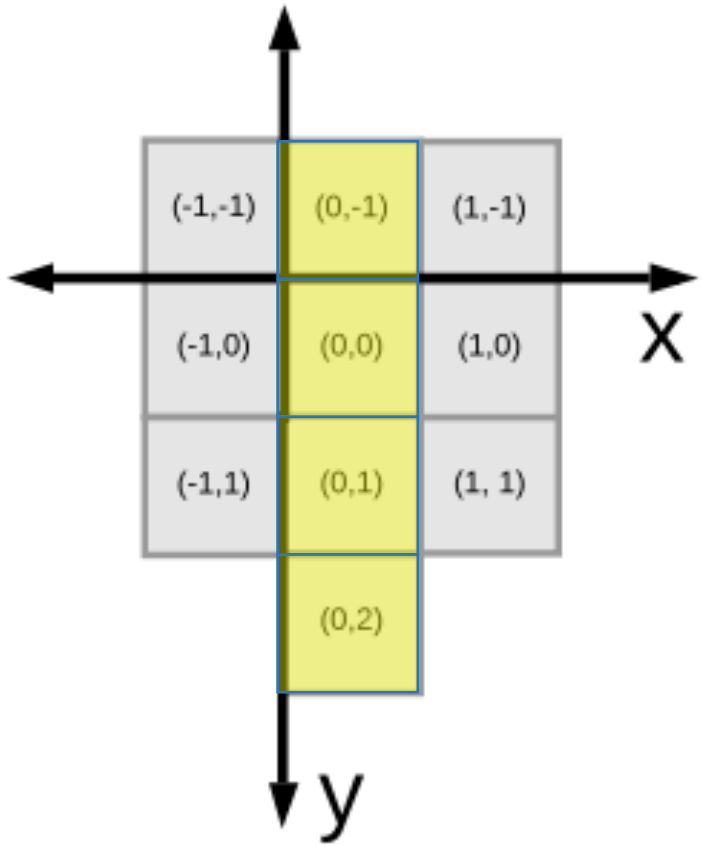


NoShape: default shape

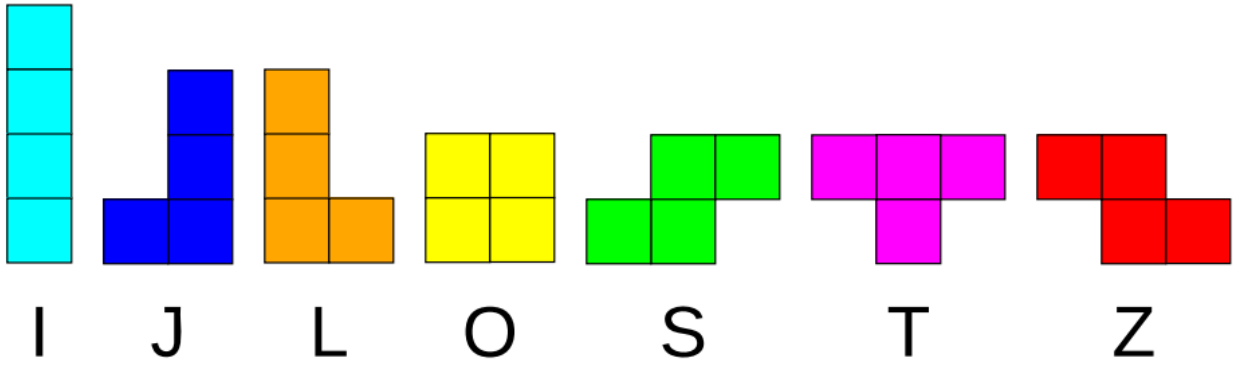
```
.getCoords() = { {0, 0}, {0, 0}, {0, 0}, {0, 0} };
.getColor()  = { 0, 0, 0 };
```

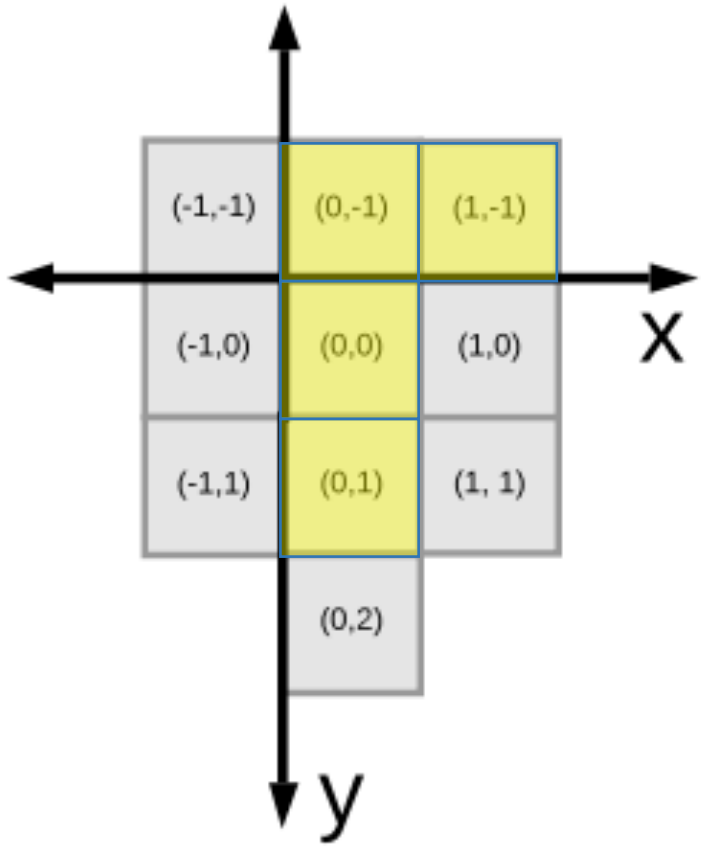
[Click here for link to LucidChart]



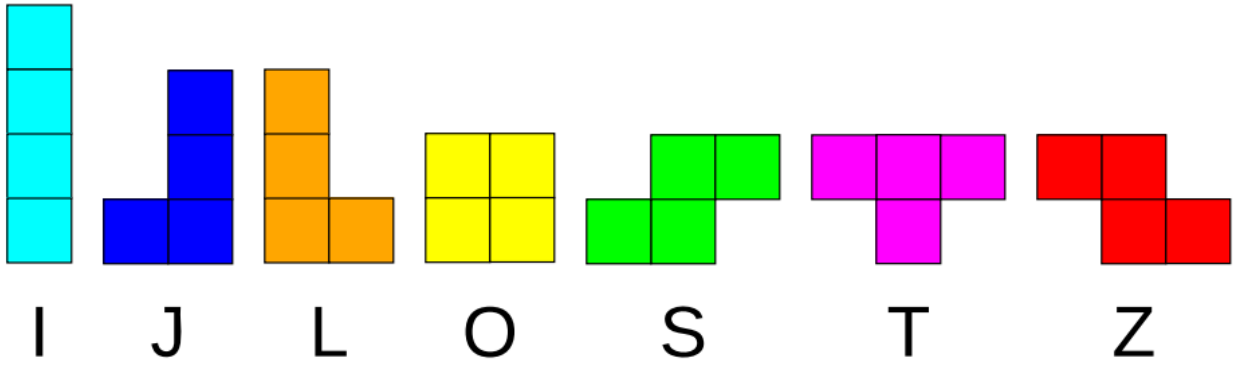


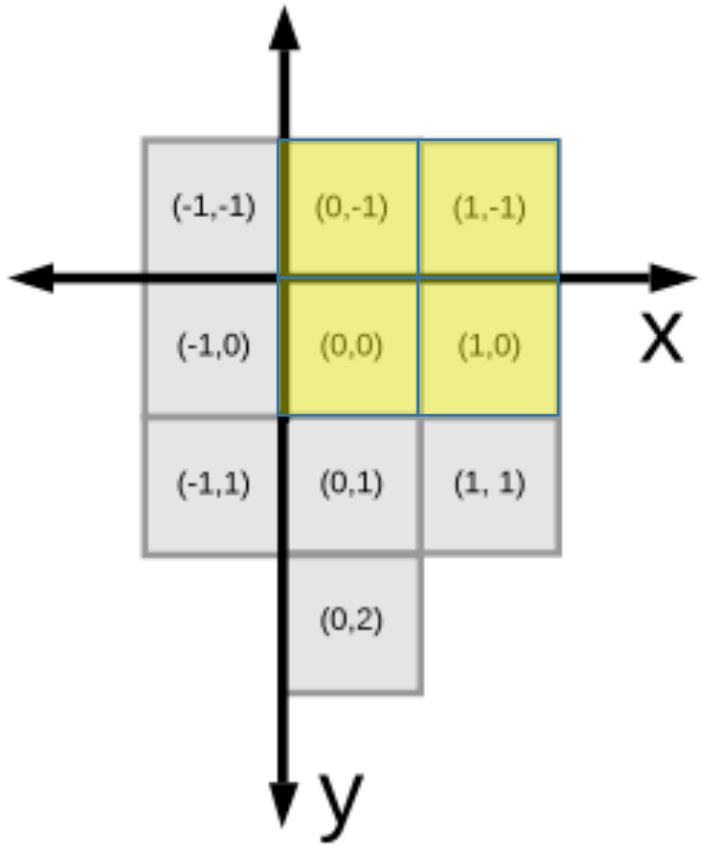

```java
package tetris;

import java.awt.Color;

public enum Tetrominoes {

    // No shape
    NoShape(new int[][]{{0, 0}, {0, 0}, {0, 0}, {0, 0}},
            new Color(0, 0, 0)),
    // I = cyan
    IShape(new int[][]{{0, -1}, {0, 0}, {0, 1}, {0, 2}},
            new Color(10, 186, 181)),
    // J = blue
    JShape(new int[][]{{1, -1}, {0, -1}, {0, 0}, {0, 1}},
            new Color(40, 100, 240)),
    // L = orange
    LShape(new int[][]{{-1, -1}, {0, -1}, {0, 0}, {0, 1}},
            new Color(240, 140, 40)),
    // O (box) = yellow
    OShape(new int[][]{{0, 0}, {1, 0}, {0, 1}, {1, 1}},
            new Color(240, 240, 40)),
    // S = green
    SShape(new int[][]{{0, -1}, {0, 0}, {1, 0}, {1, 1}},
            new Color(100, 240, 100)),
    // T = magenta bright pink
    TShape(new int[][]{{-1, 0}, {0, 0}, {1, 0}, {0, 1}},
            new Color(200, 100, 200)),
    // Z = red
    ZShape(new int[][]{{0, -1}, {0, 0}, {-1, 0}, {-1, 1}},
            new Color(240, 40, 40)),;

    public int[][] coords;
    public Color color;

    private Tetrominoes(int[][] coords, Color color) {
        this.coords = coords;
        this.color = color;
    }
}
```



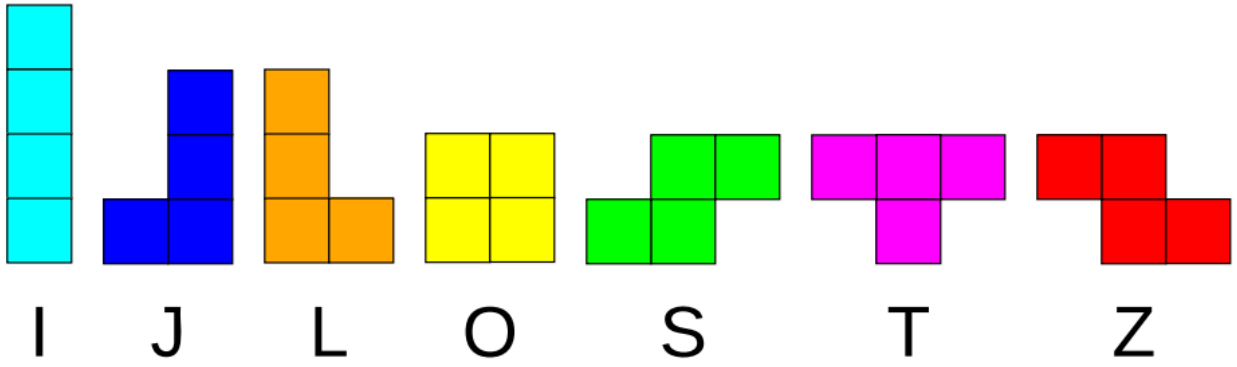


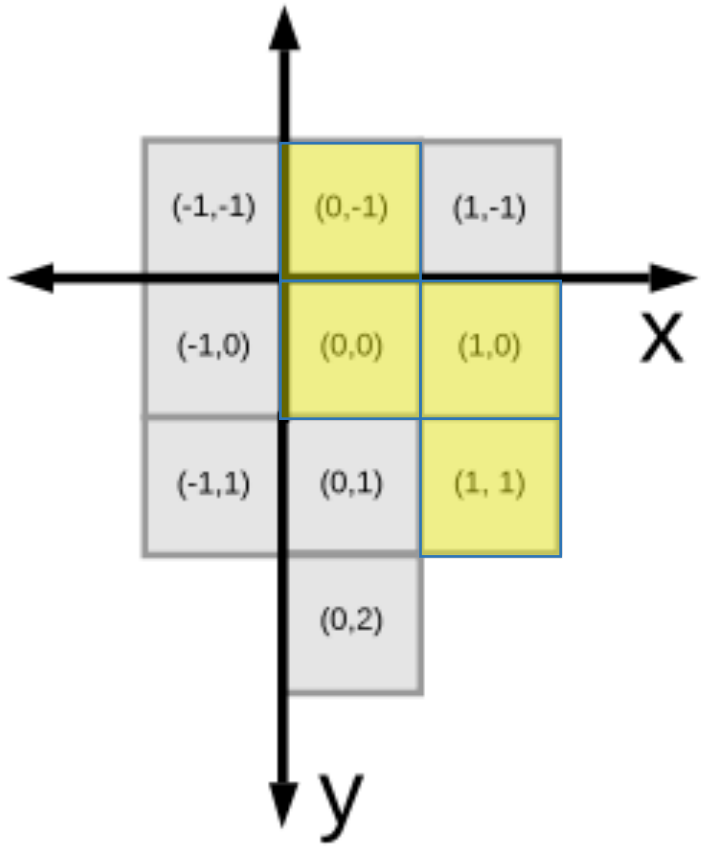

```java
package tetris;

import java.awt.Color;

public enum Tetrominoes {

    // No shape
    NoShape(new int[][]{{0, 0}, {0, 0}, {0, 0}, {0, 0}},
            new Color(0, 0, 0)),
    // I = cyan
    IShape(new int[][]{{0, -1}, {0, 0}, {0, 1}, {0, 2}},
            new Color(10, 186, 181)),
    // J = blue
    JShape(new int[][]{{1, -1}, {0, -1}, {0, 0}, {0, 1}},
            new Color(40, 100, 240)),
    // L = orange
    LShape(new int[][]{{-1, -1}, {0, -1}, {0, 0}, {0, 1}},
            new Color(240, 140, 40)),
    // O (box) = yellow
    OShape(new int[][]{{0, 0}, {1, 0}, {0, 1}, {1, 1}},
            new Color(240, 240, 40)),
    // S = green
    SShape(new int[][]{{0, -1}, {0, 0}, {1, 0}, {1, 1}},
            new Color(100, 240, 100)),
    // T = magenta bright pink
    TShape(new int[][]{{-1, 0}, {0, 0}, {1, 0}, {0, 1}},
            new Color(200, 100, 200)),
    // Z = red
    ZShape(new int[][]{{0, -1}, {0, 0}, {-1, 0}, {-1, 1}},
            new Color(240, 40, 40)),;

    public int[][] coords;
    public Color color;

    private Tetrominoes(int[][] coords, Color color) {
        this.coords = coords;
        this.color = color;
    }
}
```






```java
package tetris;

import java.awt.Color;

public enum Tetrominoes {

    // No shape
    NoShape(new int[][]{{0, 0}, {0, 0}, {0, 0}, {0, 0}},
            new Color(0, 0, 0)),
    // I = cyan
    IShape(new int[][]{{0, -1}, {0, 0}, {0, 1}, {0, 2}},
            new Color(10, 186, 181)),
    // J = blue
    JShape(new int[][]{{1, -1}, {0, -1}, {0, 0}, {0, 1}},
            new Color(40, 100, 240)),
    // L = orange
    LShape(new int[][]{{-1, -1}, {0, -1}, {0, 0}, {0, 1}},
            new Color(240, 140, 40)),
    // O (box) = yellow
    OShape(new int[][]{{0, 0}, {1, 0}, {0, 1}, {1, 1}},
            new Color(240, 240, 40)),
    // S = green
    SShape(new int[][]{{0, -1}, {0, 0}, {1, 0}, {1, 1}},
            new Color(100, 240, 100)),
    // T = magenta bright pink
    TShape(new int[][]{{-1, 0}, {0, 0}, {1, 0}, {0, 1}},
            new Color(200, 100, 200)),
    // Z = red
    ZShape(new int[][]{{0, -1}, {0, 0}, {-1, 0}, {-1, 1}},
            new Color(240, 40, 40)),;

    public int[][] coords;
    public Color color;

    private Tetrominoes(int[][] coords, Color color) {
        this.coords = coords;
        this.color = color;
    }
}
```






```java
package tetris;

import java.awt.Color;

public enum Tetrominoes {

    // No shape
    NoShape(new int[][]{{0, 0}, {0, 0}, {0, 0}, {0, 0}},
            new Color(0, 0, 0)),
    // I = cyan
    IShape(new int[][]{{0, -1}, {0, 0}, {0, 1}, {0, 2}},
            new Color(10, 186, 181)),
    // J = blue
    JShape(new int[][]{{1, -1}, {0, -1}, {0, 0}, {0, 1}},
            new Color(40, 100, 240)),
    // L = orange
    LShape(new int[][]{{-1, -1}, {0, -1}, {0, 0}, {0, 1}},
            new Color(240, 140, 40)),
    // O (box) = yellow
    OShape(new int[][]{{0, 0}, {1, 0}, {0, 1}, {1, 1}},
            new Color(240, 240, 40)),
    // S = green
    SShape(new int[][]{{0, -1}, {0, 0}, {1, 0}, {1, 1}},
            new Color(100, 240, 100)),
    // T = magenta bright pink
    TShape(new int[][]{{-1, 0}, {0, 0}, {1, 0}, {0, 1}},
            new Color(200, 100, 200)),
    // Z = red
    ZShape(new int[][]{{0, -1}, {0, 0}, {-1, 0}, {-1, 1}},
            new Color(240, 40, 40)),;

    public int[][] coords;
    public Color color;

    private Tetrominoes(int[][] coords, Color color) {
        this.coords = coords;
        this.color = color;
    }
}
```

```
/**
 * Called by Board as the current piece of Tetromino falling.
 * Has variables storing information on the current piece on the Board.
 * Contains methods for manipulating coordinates of a Tetromino
 */
```

public class Shape

```
 /**
  * coords of a shape is made out of 4 square blocks,
  * stored in each row of int[4][2] coords,
  * with each column as (x,y) coordinates of each block
  */
 private Tetrominoes currShape;
 private int[][] coords;


 // Default Constructor
 public Shape() {
     // 4 squre blocks on each row, 2 coords for (x, y)
     this.coords = new int[4][2];
     // default enum shape is NoShape, then override it
     setShape(Tetrominoes.NoShape);
 }

 // override enum Tetromino passed in
 // by using Random to pick a new Tetrominoes enum
 // store coords from enum picked into this.coords
 private void setShape(Tetrominoes shape) { }
```

```
// x coordinate of block 1 of 4 of a Tetrominoes shape
public int x(int block) {
    return this.coords[block][0];
}


// y coordinate of block 1 of 4 of a Tetrominoes shape
public int y(int block) {
    return this.coords[block][1];
}

// return Tetrominoes representing the current Shape
public Tetrominoes getTetrominoShape() {
    return currShape;
}


// manipulates coords of Shape calling on this method,
// to rotate it in the direction chosen.
// and return new Shape with rotated coords,
// but keep the Tetromino enum the same
public Shape rotateLeft() / rotateRight() { }
```

[Click here for link to LucidChart]

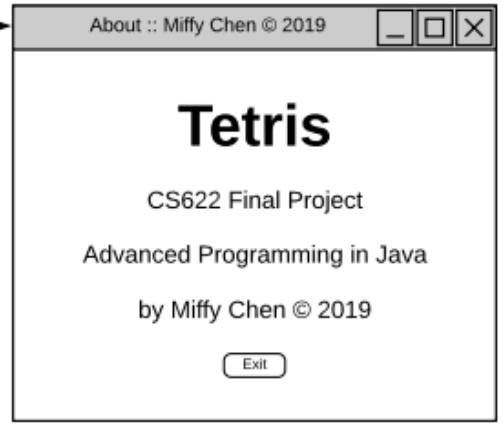
Miffy Chen 2019/03/19 Final Project Proposal

Mock Ups

[Click here for link to LucidChart]

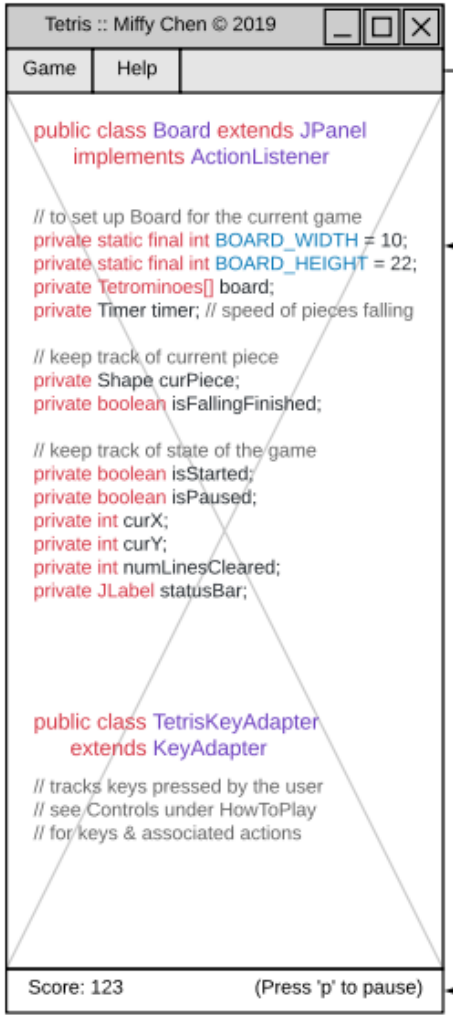
Miffy Chen 2019/03/19
Final Project Proposal

public class About extends JFrame



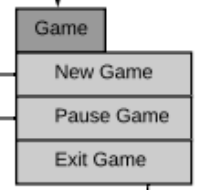
opens

public class Tetris extends JFrame



opens

// disposes current Tetris opened
this.dispose()
// run a new instance of Tetris
new Tetris().setVisible(true);



Board.pause();

statusBar updates whether Board is paused

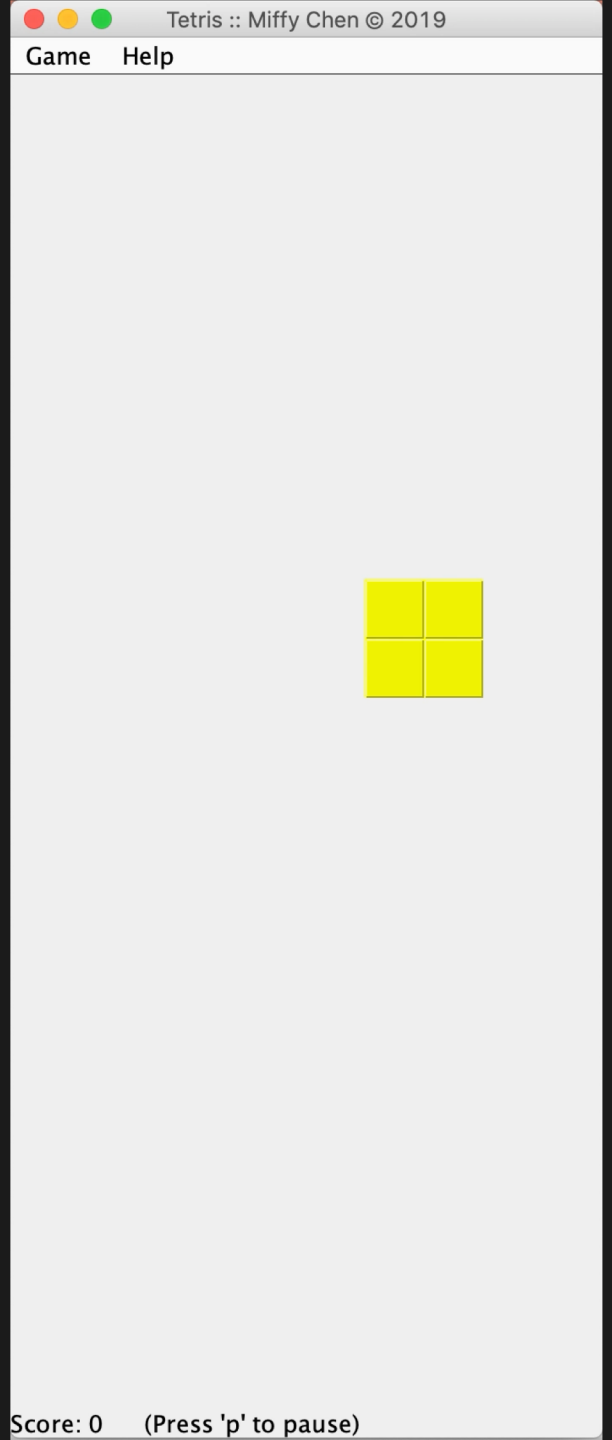
this.dispose();

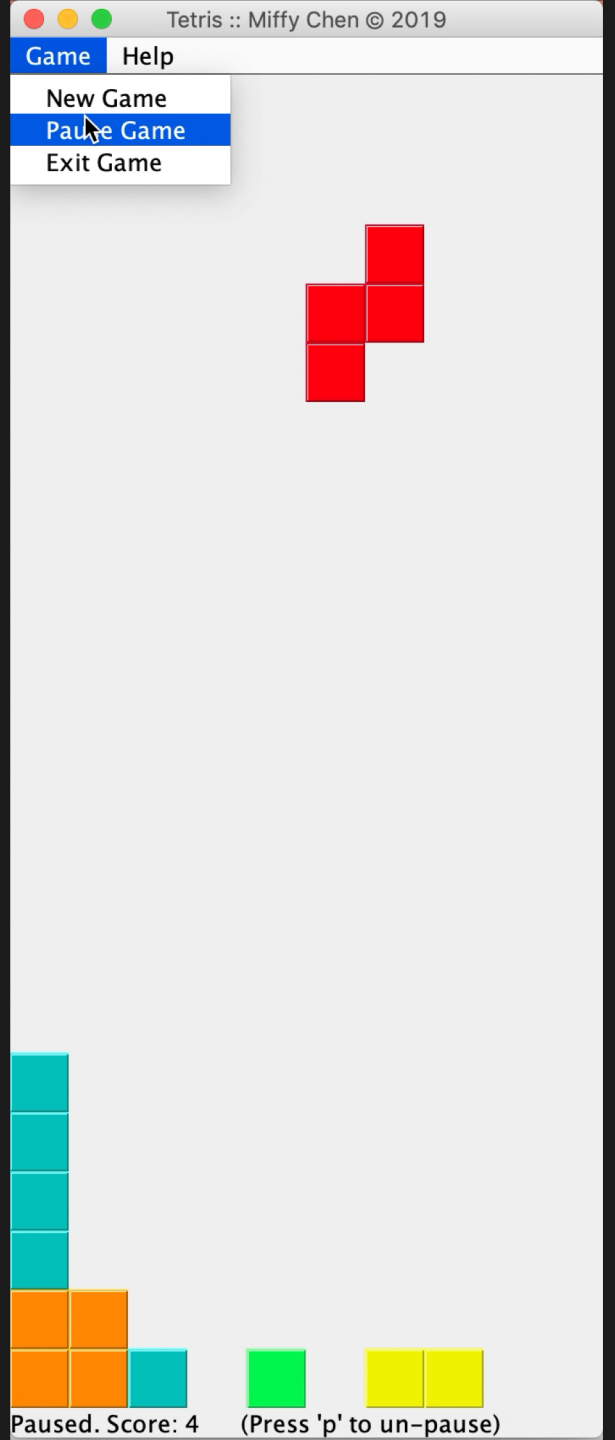
Terminates Program

(JLabel Board.statusBar)

Paused. Score: 123 (Press 'p' to un-pause)

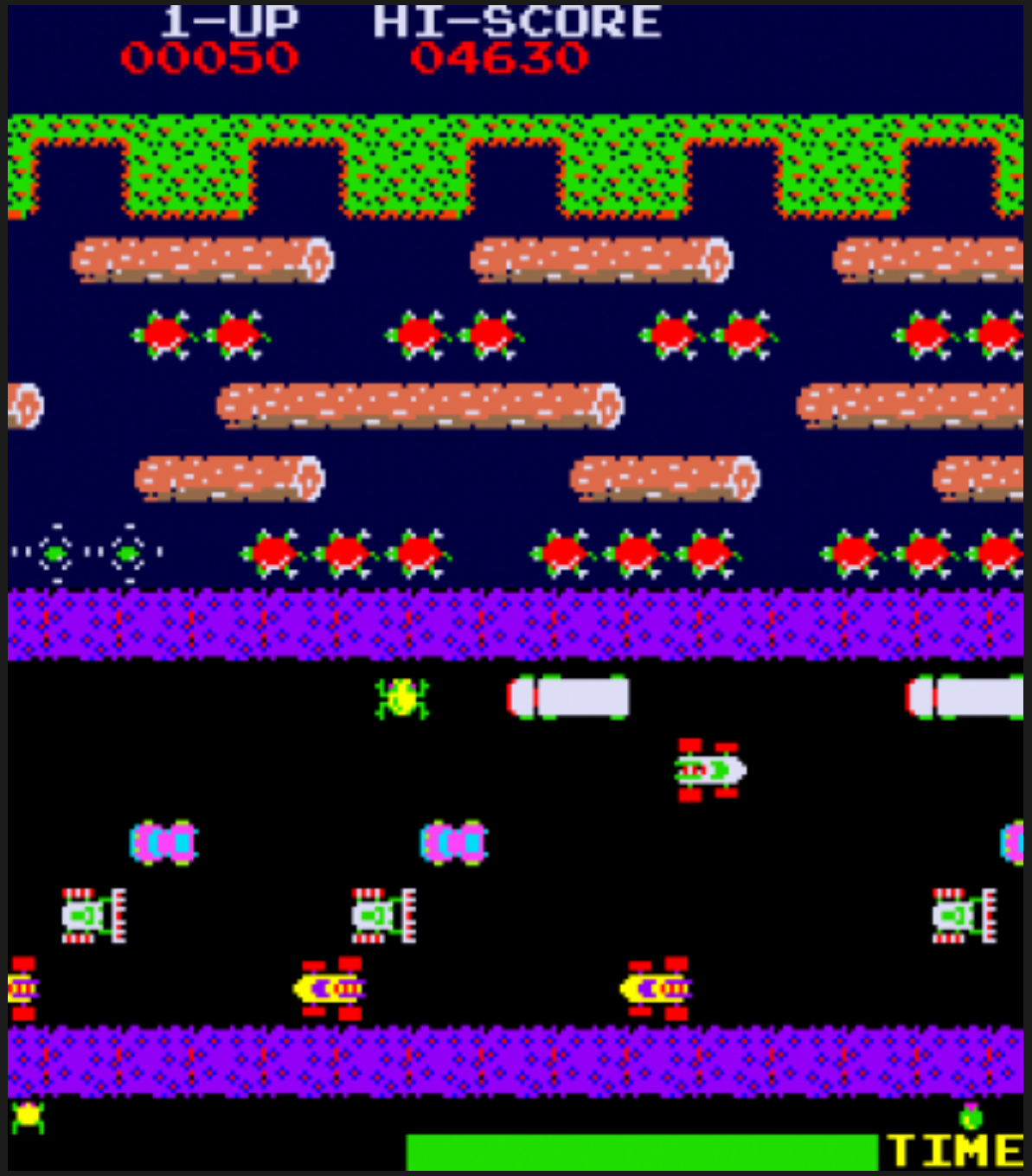
public class HowToPlay extends JFrame


Tetris :: Miffy Chen © 2019
Game
Help
Score: 0
(Press 'p' to pause)


Tetris :: Miffy Chen © 2019
Game
Help
New Game
Pause Game
Exit Game
Paused. Score: 4
(Press 'p' to un-pause)


How To Play
How To Play
Tetriminos, shapes made up of 4 blocks of squares blocks,
will fall in a random sequence from the top of the board all the way down.
Move the shapes sideways, rotate, or drop them to form a horizontal line
with no gaps to clear the line.
Players are awarded 1 point per lines cleared.
Controls
Function
Key
Move Left
←
Move Right
→
Rotate Piece
↑
Move Down
↓
Drop Piece
[space]
Pause Game
p
Back


About
Tetris
CS622 Final Project
Advance Programming in Java
by Miffy Chen © 2019
Exit



FROGGER

- 1981 Konami version

Objective: cross the roads/pond to his home

Scoring:

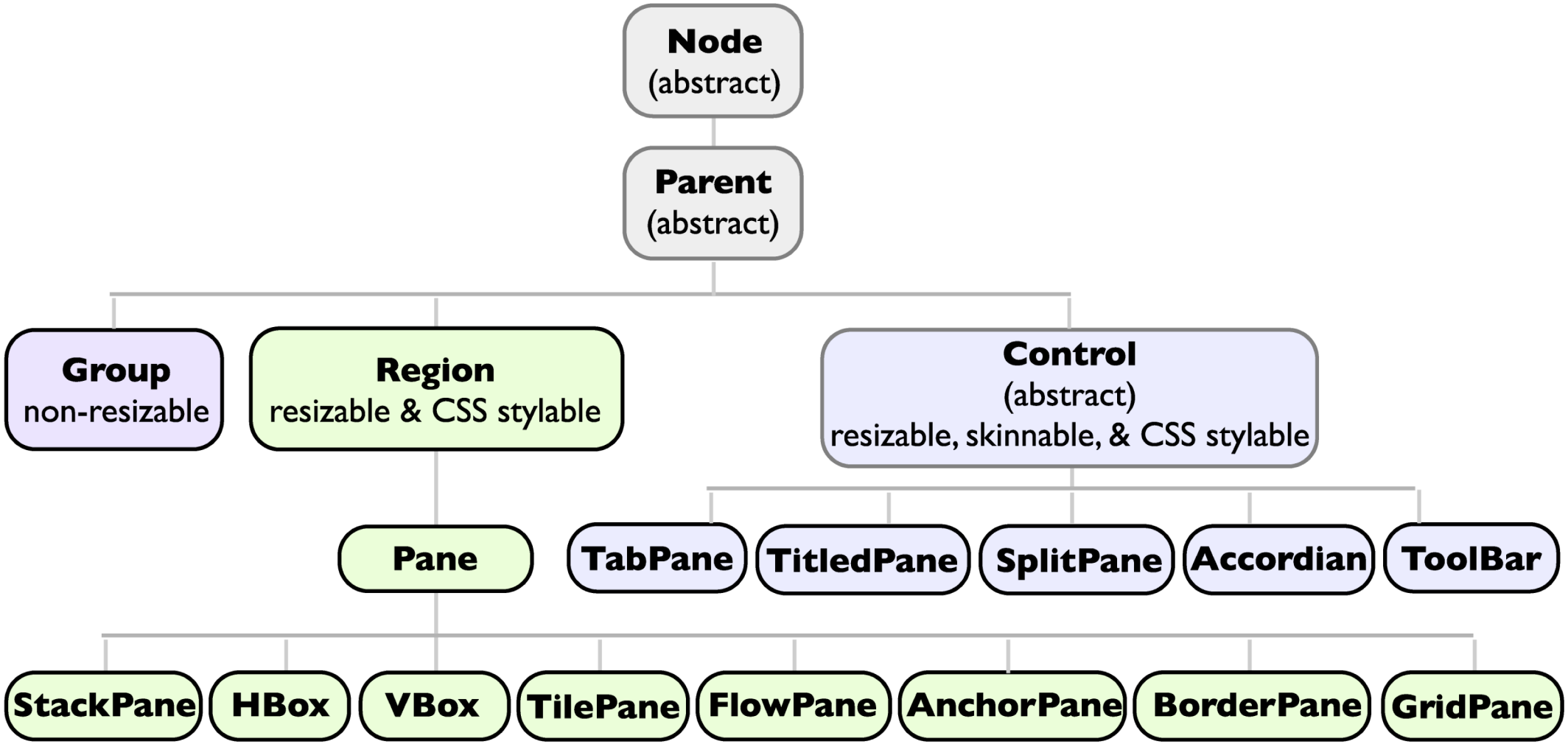
- Each forward step taken += 10
- Get home += 60
- If dies -= 50 (only if score is >= 50)

Obstacles:

- Purple = safe space (@ 50 point mark)
- Turtle sinks down == death
- Edge of screen: Logs / Turtles, pushes back

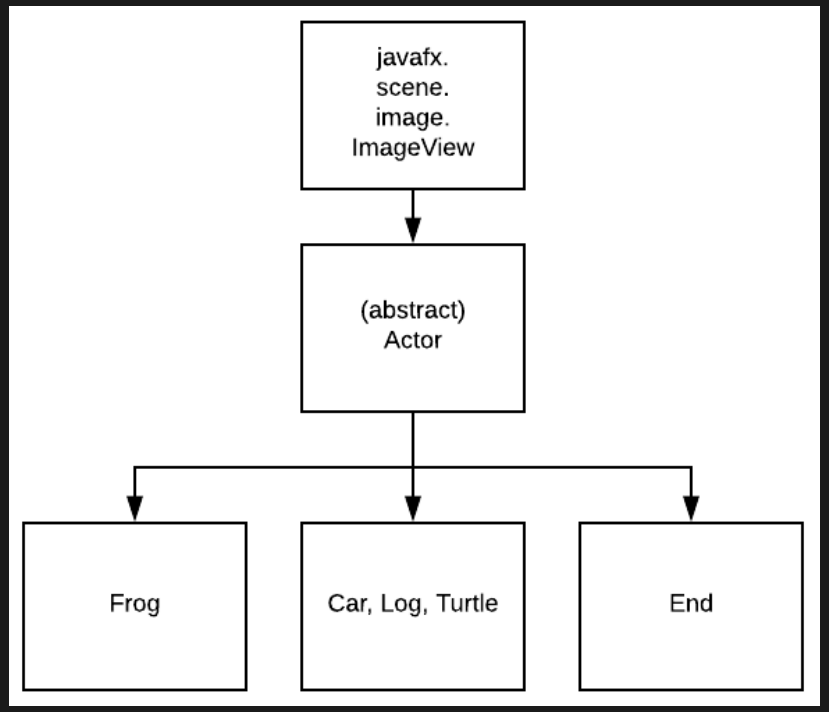
https://youtu.be/WNrz9_Fe-Us?t=80

JavaFX 2.0 Layout Classes



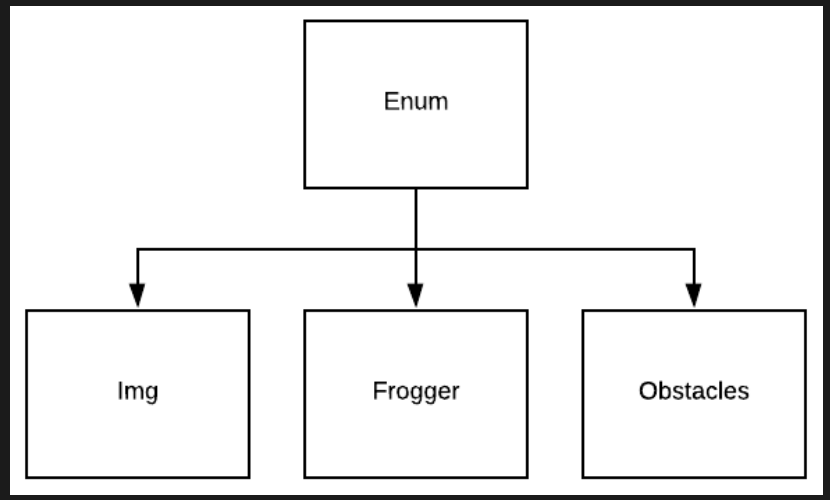
JavaFX	Classes
• Application	Main
• Pane	(abs)World
• (abs)World	GameStage
• ImageView	(abs)Actor
• (abs)Actor:	Frog, Obstacles (Car, Log, Turtle) (made into enums)

FROGGER



Actor extends ImageView

- added to GameStage

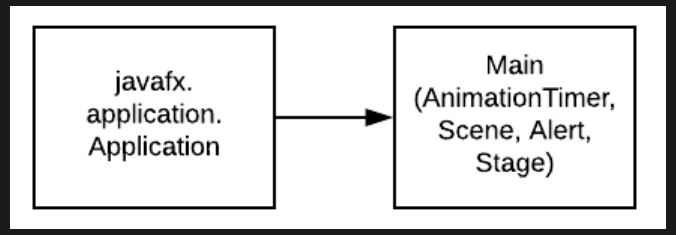


Actors utilize Enums:

- Data of each Image
- Get x/y coordinates
- Returns Image / ImageView

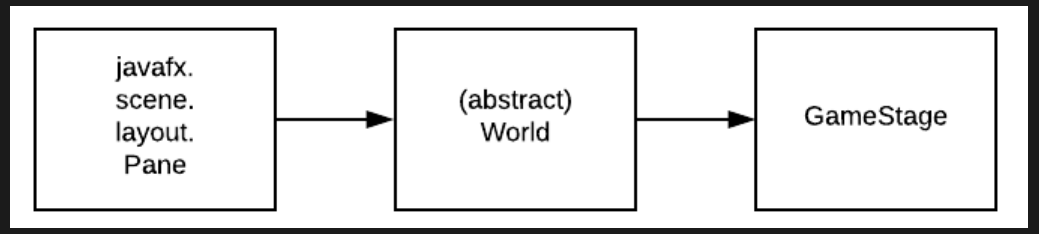
FROGGER

Main extends Application

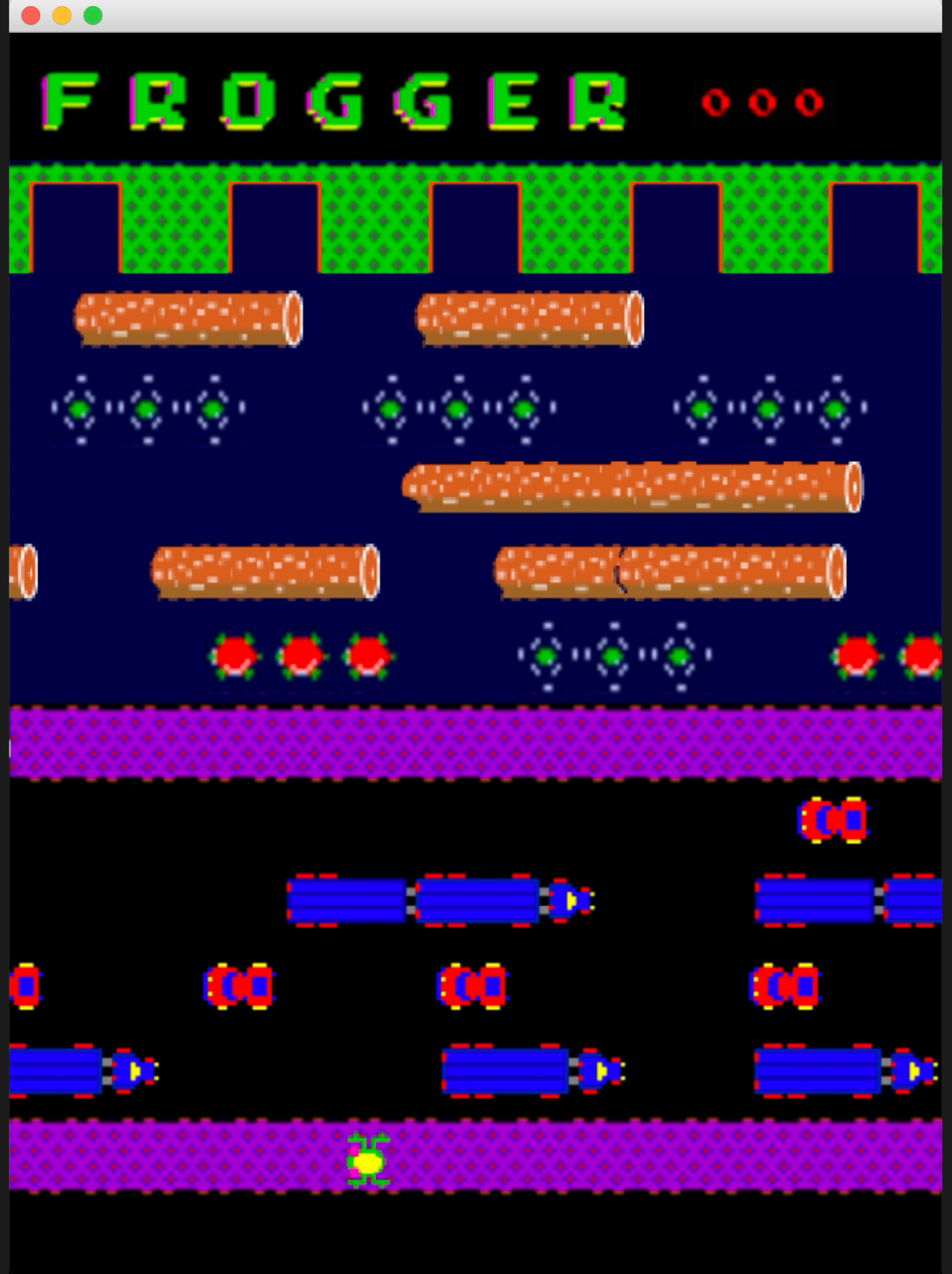


- AnimationTimer
- new Scene (GameStage, xSize, ySize)
- GameStage.add(Actor/Image);
- GameStage.playMusic() / .stopMusic();

abstract World extends Pane



- abstract void act (long now)
- World.add(Actor[extends ImageView])
- World.add(Image)
- World.start(): timer
- World.stop(): timer


FROGGER


FROGGER
Player Name
Please enter your name:
Name:
Miffy
Cancel
OK

Future Improvements

Tetris

- High Scores
- “Next” Preview
- “Hold” Queue
- Timer Display

Frogger

- High Scores
- Lives Limit
- Time Limit

Thank You!!