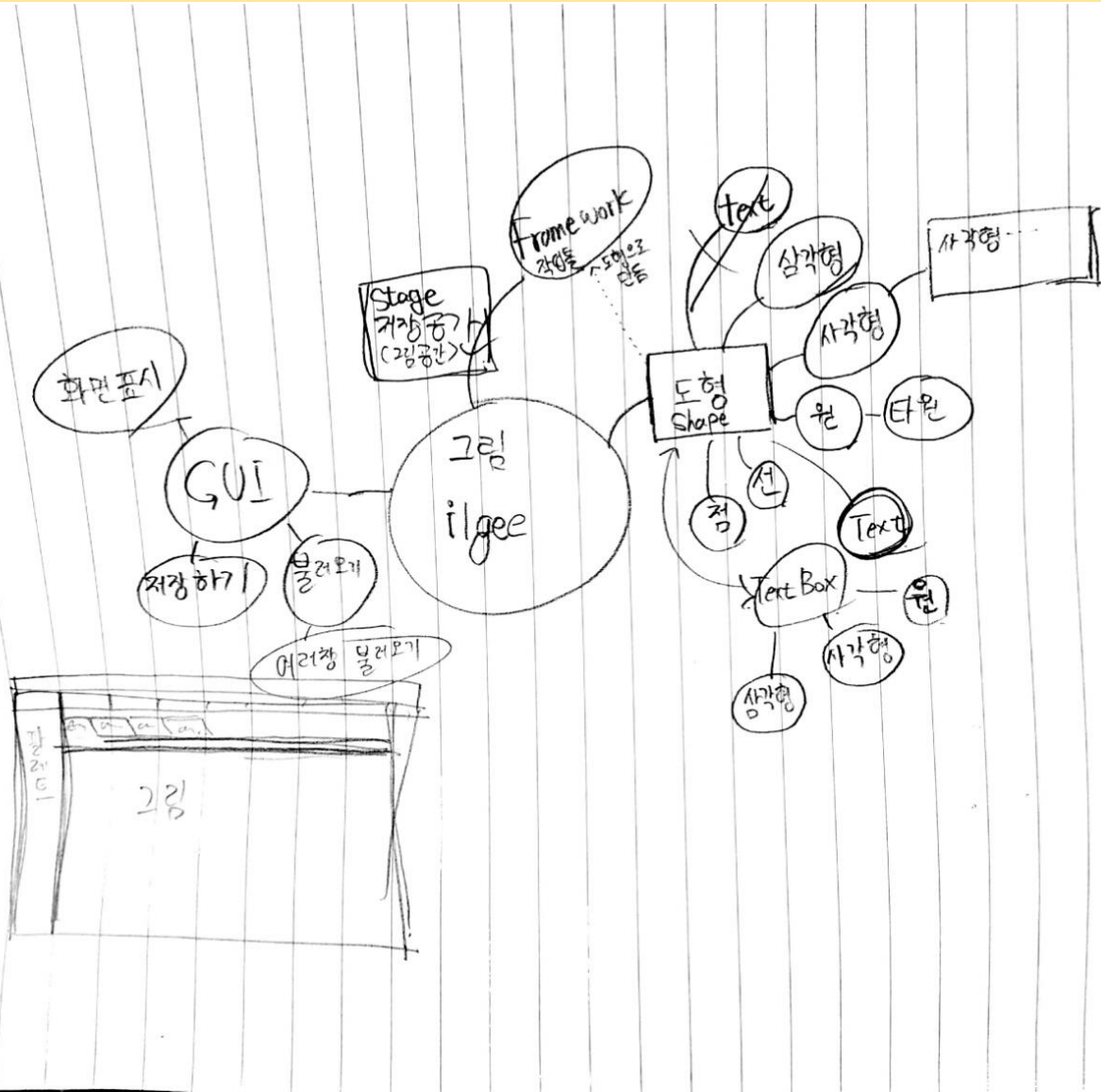


# 애니메이션 일기 UML 및 설계 자료

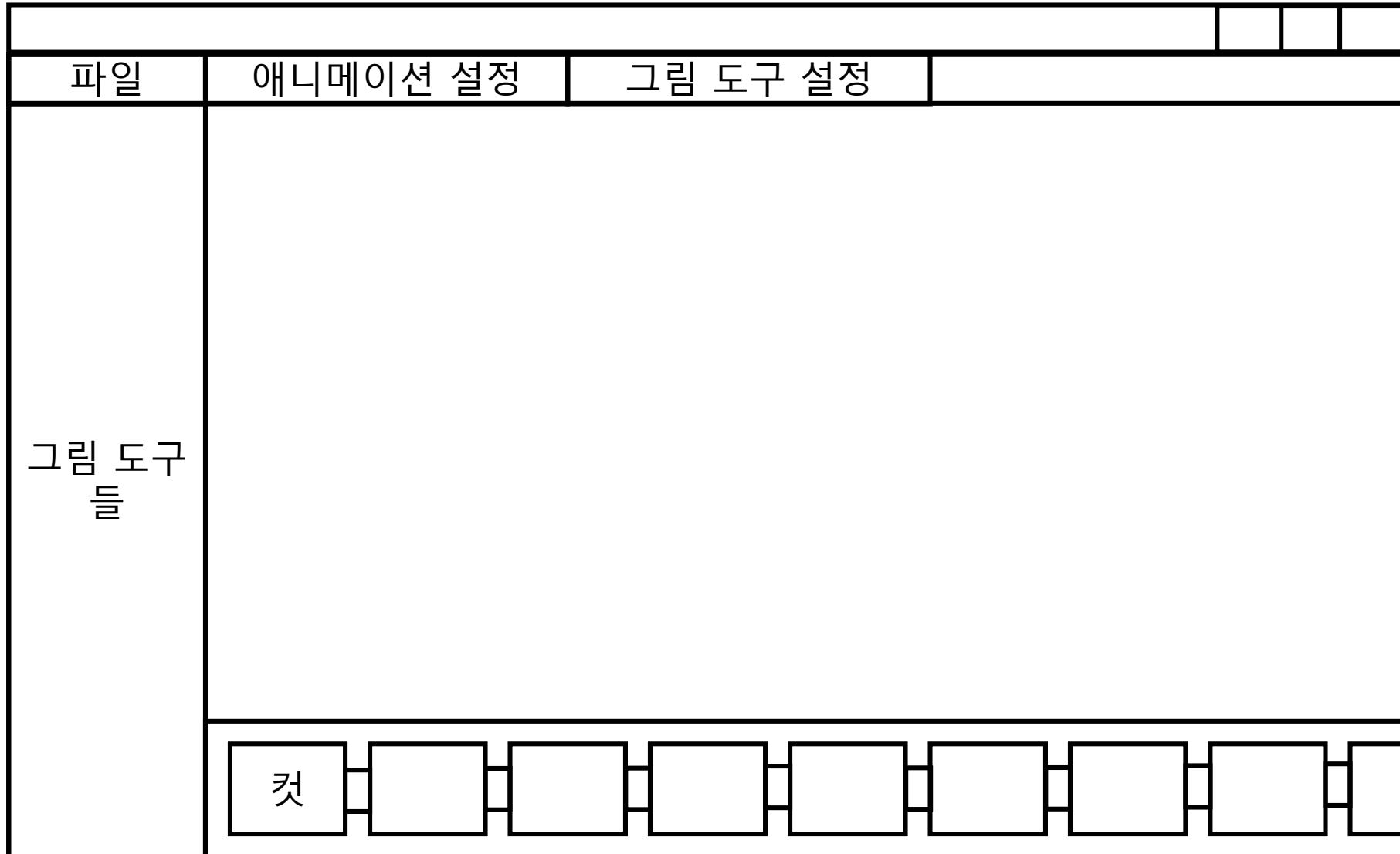
20151012 최현준

# 1. 설계 전 마인드맵



- 요소 위주로 표현함
- 도형
- GUI
- 템플릿 및 작업 공간
- 전체적인 완성 모습을 생각하며 마인드맵을 제작함
- 사람의 추억을 하나의 프레임으로 표현하기는 어렵다고 생각함
- 편의성
- 접근성

# 1. 설계 전 마인드맵



## 2. 기능 생각 및 정리, 제약 조건



필수 기능 1. 그림과 글이 동시에 나와야 한다.



필수 기능 2. 저장 및 불러 오기가 가능해야 한다.

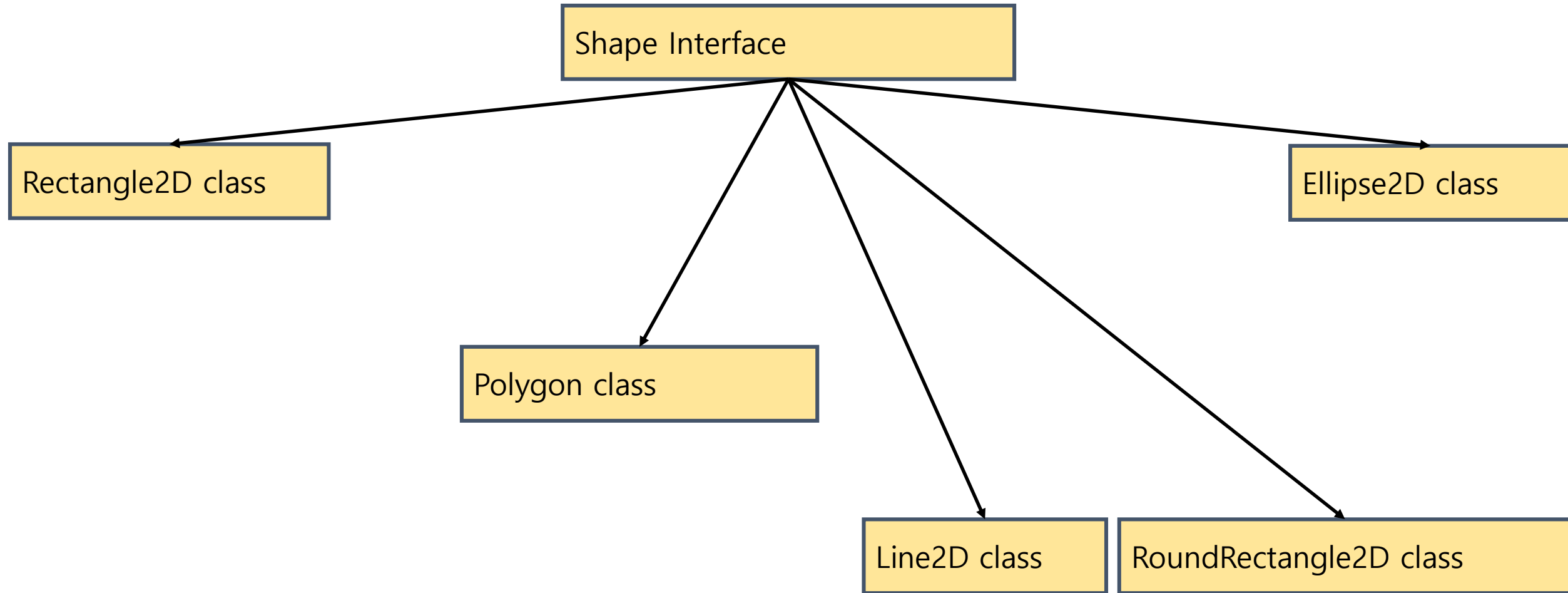
## 2. 기능 생각 및 정리, 제약 조건

- 생각한 기능 1. PPT 처럼 미리 템플릿을 만들어서 작성 가능하게 한다.
- 생각한 기능 2. 그림판과는 다르게 각각의 요소를 선택 가능하도록 만든다.
- 생각한 기능 3. 슬라이드를 이용해서 여러 장의 그림을 보여 줄 수 있도록 한다.
- 생각한 기능 4. 드래그 앤 드랍 방식의 이미지 추가 및 불러오기를 가능하게 한다.

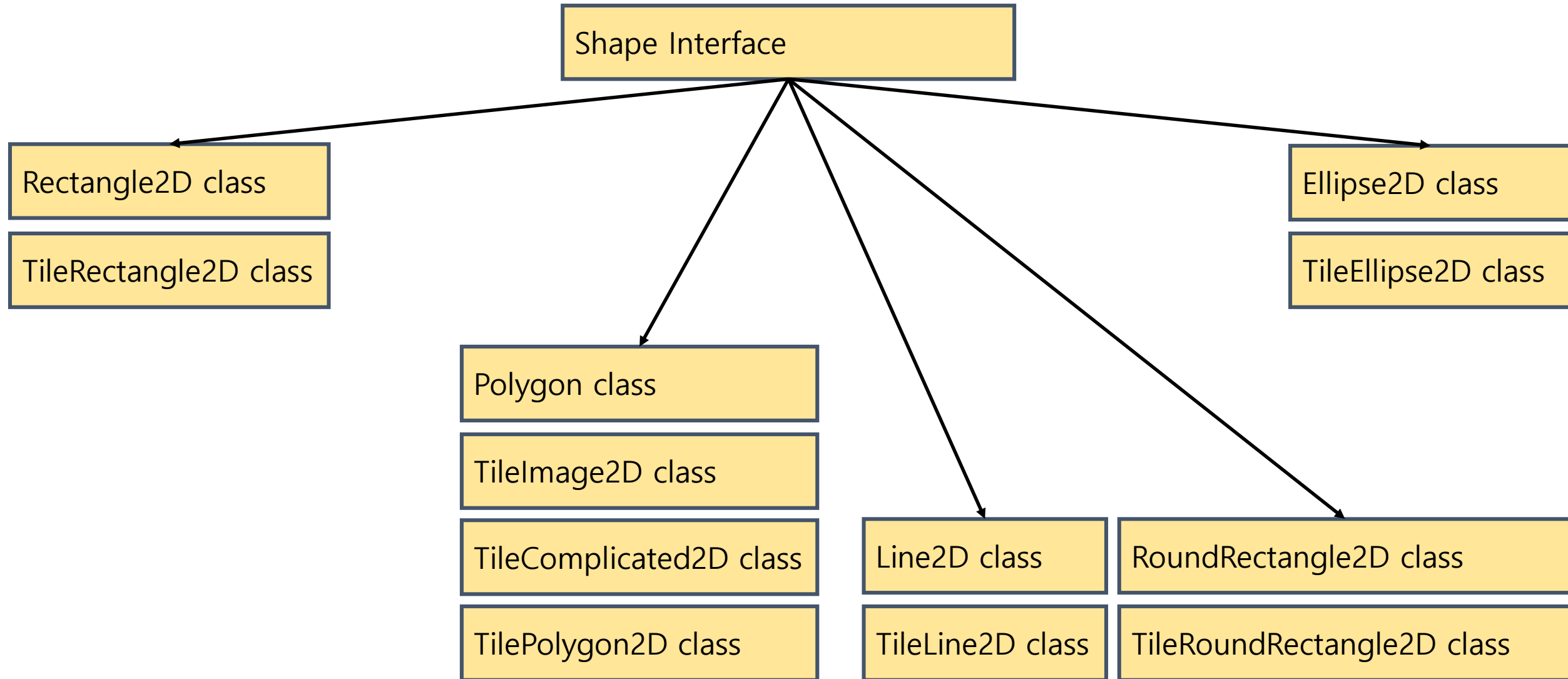
## 2. 기능 생각 및 정리, 제약 조건

- 생각한 기능 5. 그림을 연속해서 시간 차를 두고 보여주게 된다면 애니메이션처럼 보일 것 이다.
- 생각한 기능 6. 애니메이션에 대한 정보는 따로 클래스를 만들어서 실행하게 한다.
- 생각한 기능 7. 애니메이션을 쉽게 만들기 위해 UI를 최소화 하도록 한다.
- 생각한 기능 8. 빠른 프로그램 실행을 위해서 실행 모드와 제작 모드를 나눈다.

# 3. 계층 관계 IDEA

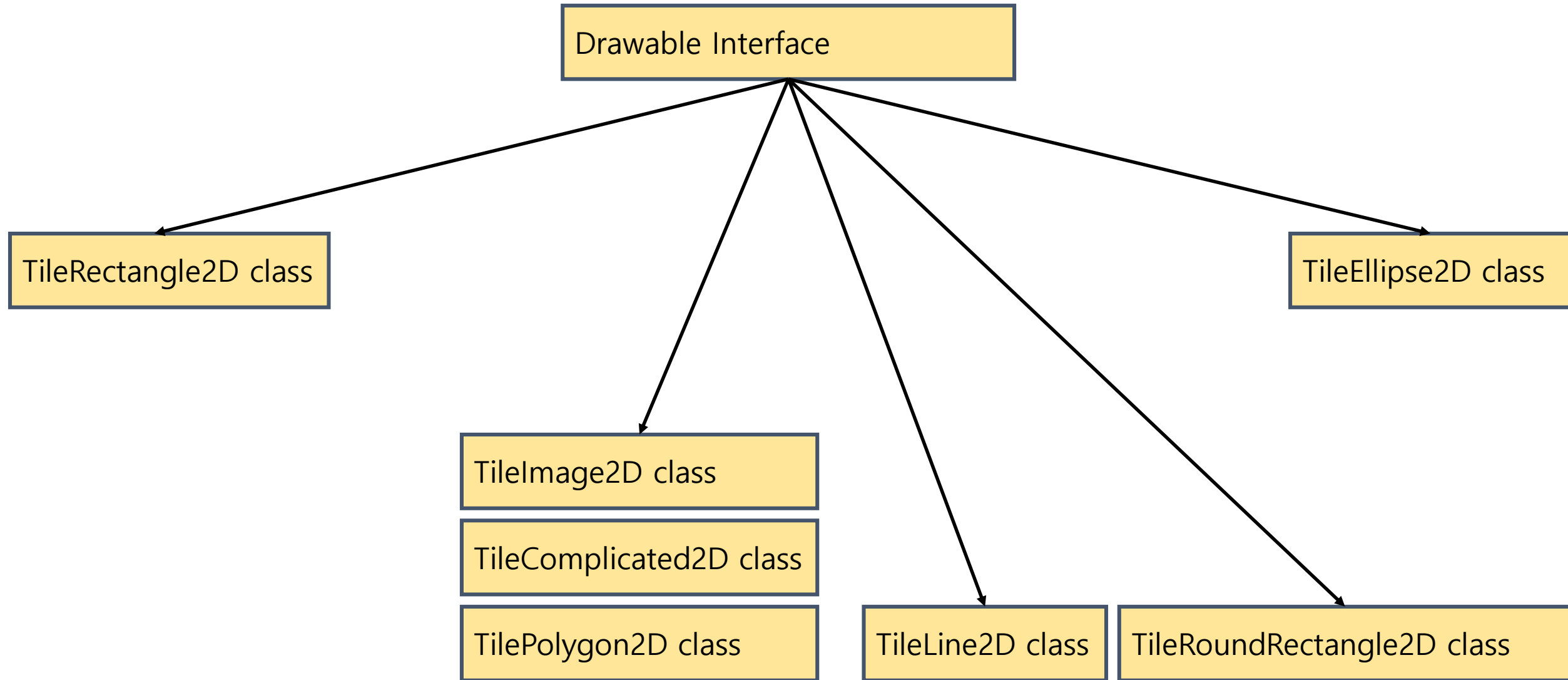


# 3. 계층 관계 IDEA

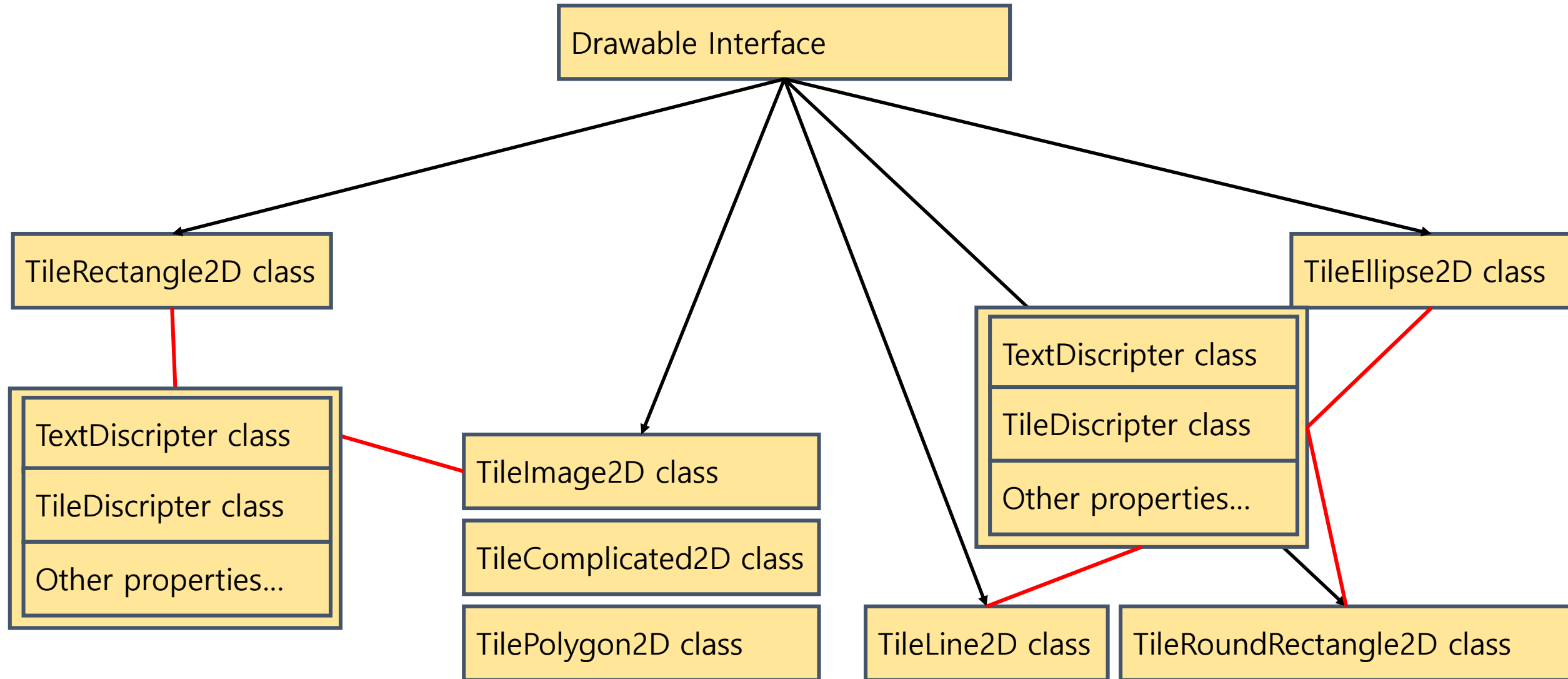




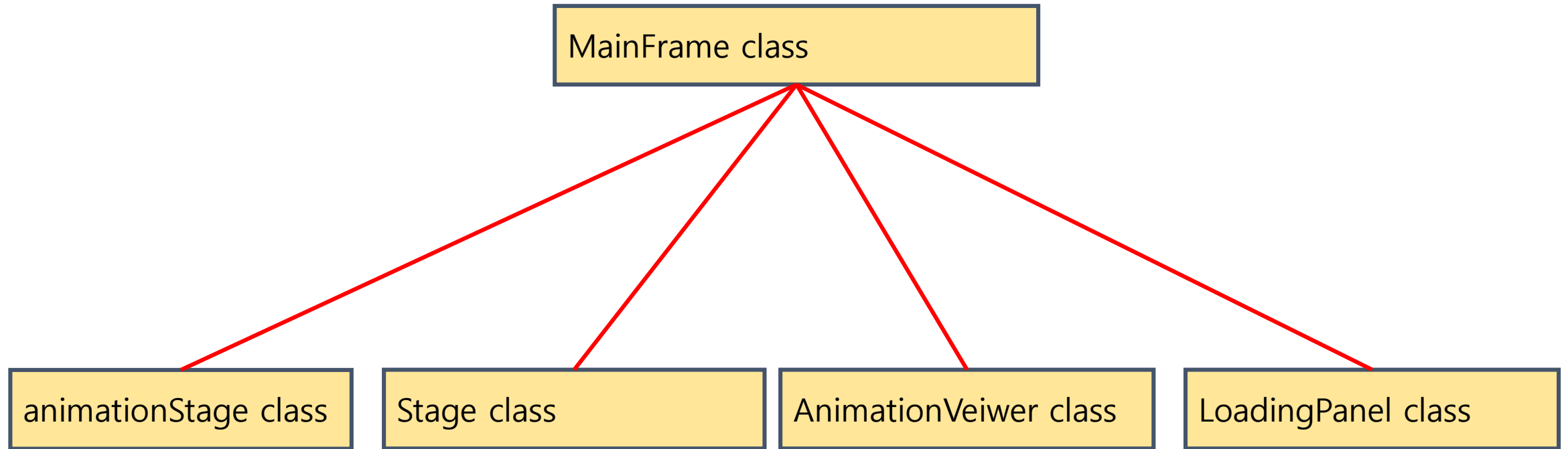
# 3. 계층 관계 IDEA



# 3. 계층 관계 IDEA

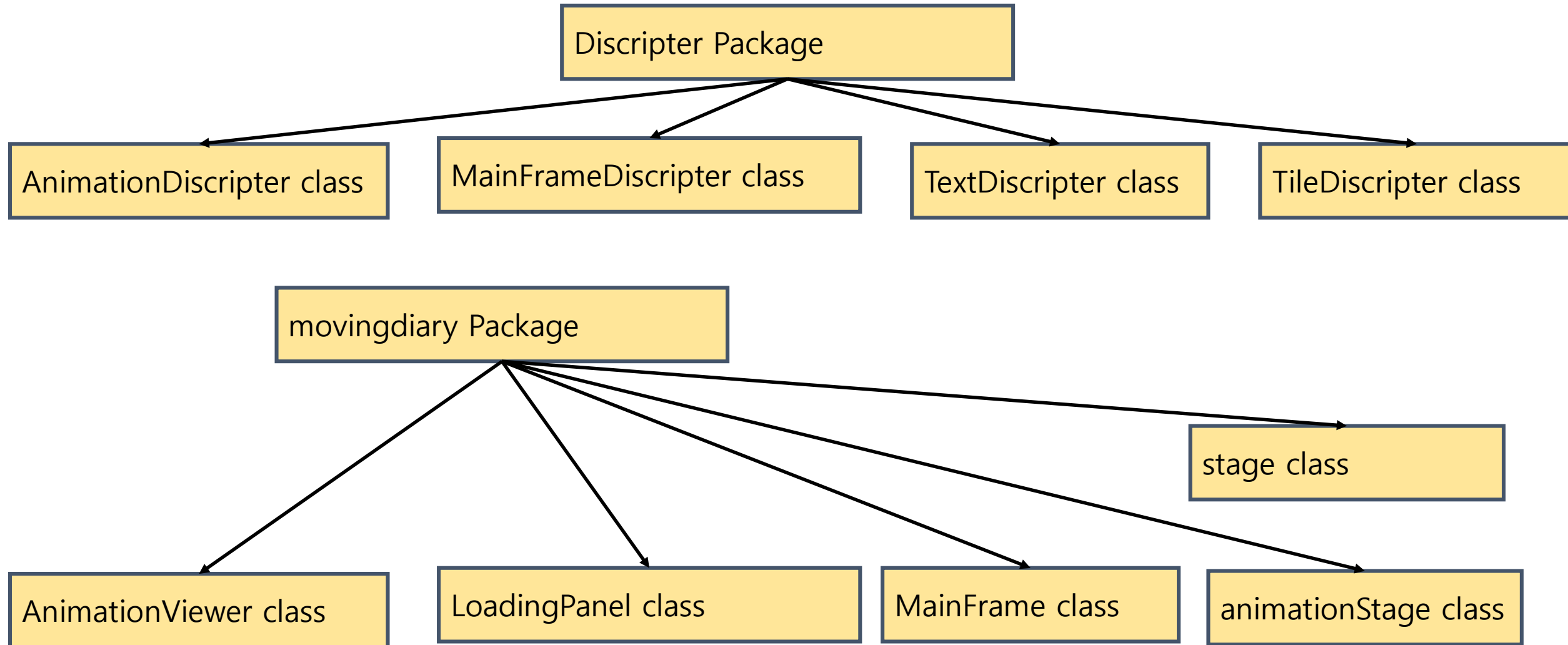


# 3. 계층 관계 IDEA

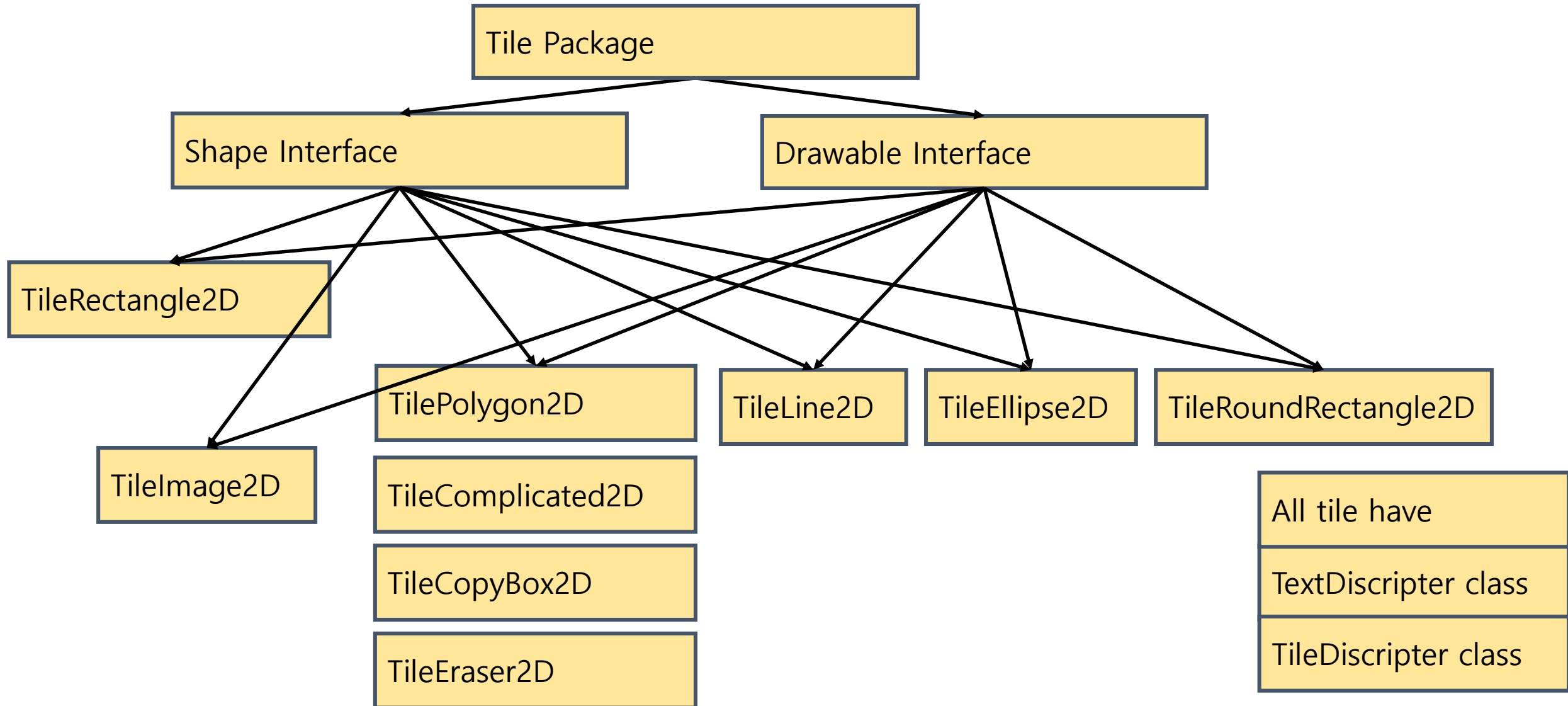


## 4. 데모

# 5. UML < View & Control >



# 5. UML < Model >



# 6. 상세 설계 및 UML

Drawable Interface

-

```
public void draw(Graphics g);  
public boolean contain(int x, int y);  
public boolean isClicked();  
public void setClicked(boolean clicked);  
public void drawBorder(Graphics g);  
public int getShapeType();  
public int getDepth();  
public void setDepth(int depth);
```

# 6. 상세 설계 및 UML

Drawable, Cloneable, Serializable

Draw(g), contain(x, y) ...

```
public class TilePolygon2D extends Polygon  
implements Drawable, Cloneable, Serializable
```

```
private int largeX, largeY, smallX, smallY;  
private final static int ERROR_DISTANCE = 10;  
private boolean complete;  
private TileDiscripiter td;  
private TextDiscripiter ttd;
```

Getter / Setter

```
public class TileEllipse2D extends Ellipse2D  
implements Drawable, Cloneable, Serializable
```

```
private int minX, minY, maxX, maxY;  
private TileDiscripiter td;  
private TextDiscripiter ttd;
```

Getter / Setter

```
public class TileRectangle2D extends Rectangle2D  
implements Drawable, Cloneable, Serializable
```

```
private int leftTopX, leftTopY;  
private int rightBottomX, rightBottomY;  
private TileDiscripiter td;  
private TextDiscripiter ttd;
```

Getter / Setter

TileRoundRectangle2D도 같음



# 6. 상세 설계 및 UML

```
int a = getMiddleX();
int b = getMiddleY();

double cosTemp = Math.cos(Math.toRadians((double)getTd().getRotateDegree()));
double sinTemp = Math.sin(Math.toRadians((double)getTd().getRotateDegree()));

double xRotateValue = (cosTemp-1) * a + sinTemp * b;
double yRotateValue = (cosTemp-1) * b - sinTemp * a;

Graphics2D g2d = (Graphics2D)g;
AffineTransform old = g2d.getTransform();
g2d.rotate(Math.toRadians(getTd().getRotateDegree()));

g.setColor(getTd().getFillingColor());
g.fillRect((int)getX() + (int)xRotateValue, (int)getY() + (int)yRotateValue, (int)getWidth(), (int)getHeight());
g.setColor(getTd().getTextColor());
g.setFont(getTd().getTextFont());
String[] temp = getTd().getTextContext().split("\n");
for(int i = 0; i < temp.length; i++) {
    g.drawString(temp[i], getTd().getX() + (int)xRotateValue, getTd().getY() + (int)yRotateValue + (i * (getTd().getTextFont().getSize() + 2)));
}
g.setColor(getTd().getBorderColor());
g.drawRect((int)getX() + (int)xRotateValue, (int)getY() + (int)yRotateValue, (int)getWidth(), (int)getHeight());

g2d.setTransform(old);
```

# 6. 상세 설계

```
if(getTd().getRotateDegree() == 0) {
    if(getLeftTopX() <= x && getRightBottomX() >= x && getLeftTopY() <= y && getRightBottomY() >= y) {
        result = true;
    }
}
else {
    int x1 = getLeftTopX();
    int x2 = getRightBottomX();
    int y1 = getLeftTopY();
    int y2 = getRightBottomY();

    int w = (int)getWidth();
    int h = (int)getHeight();

    double cosTemp = Math.cos(Math.toRadians((double)getTd().getRotateDegree()));
    double sinTemp = Math.sin(Math.toRadians((double)getTd().getRotateDegree()));

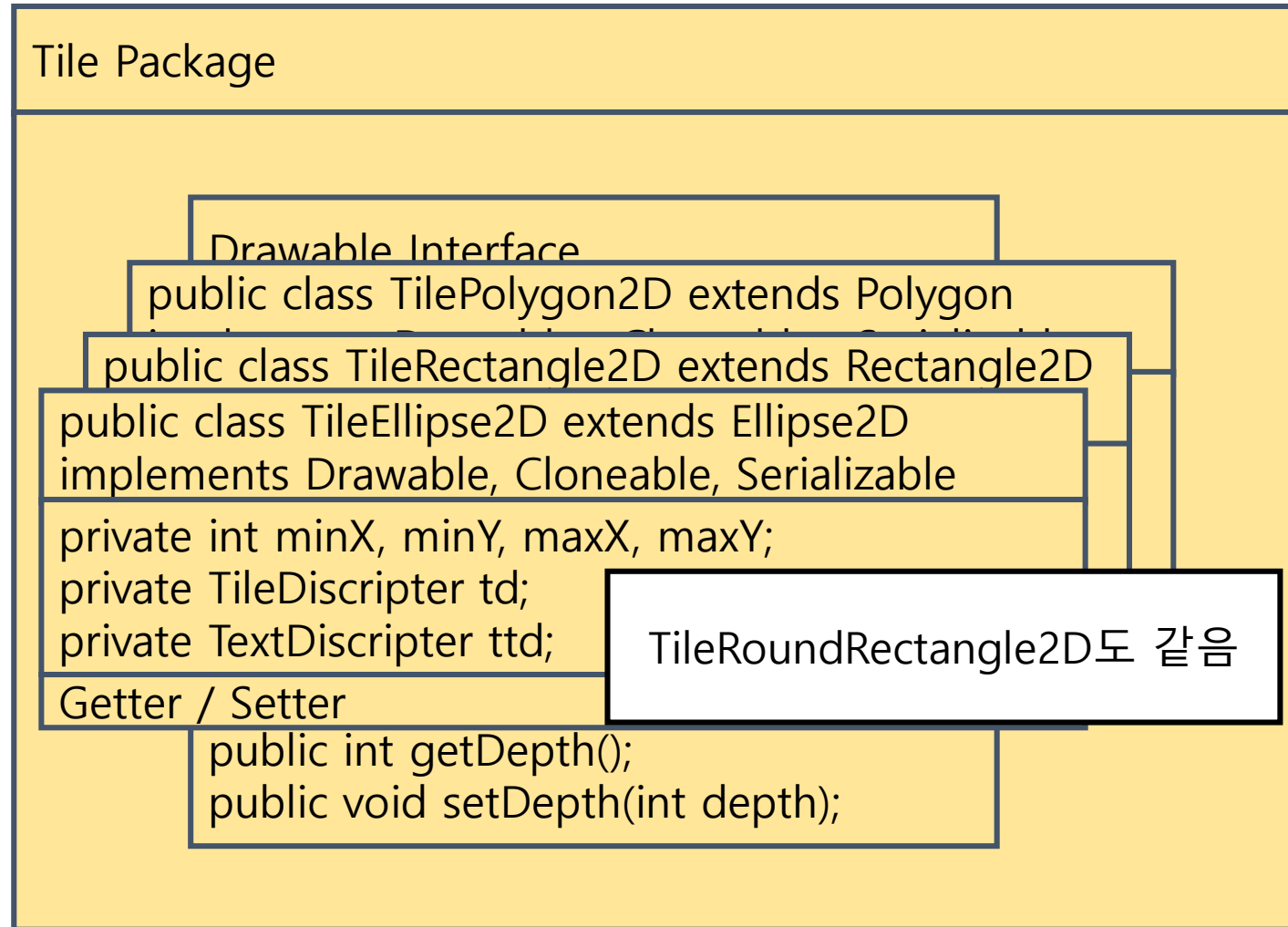
    x1 = (int) ( x1 + w * ( 1 - cosTemp ) / 2 + h * sinTemp / 2 );
    y1 = (int) ( y1 + h * ( 1 - cosTemp ) / 2 - w * sinTemp / 2 );
    x2 = (int) ( x2 + w * ( cosTemp - 1 ) / 2 - h * sinTemp / 2 );
    y2 = (int) ( y2 + h * ( cosTemp - 1 ) / 2 + w * sinTemp / 2 );

    int[] tempX = new int[5];
    int[] tempY = new int[5];

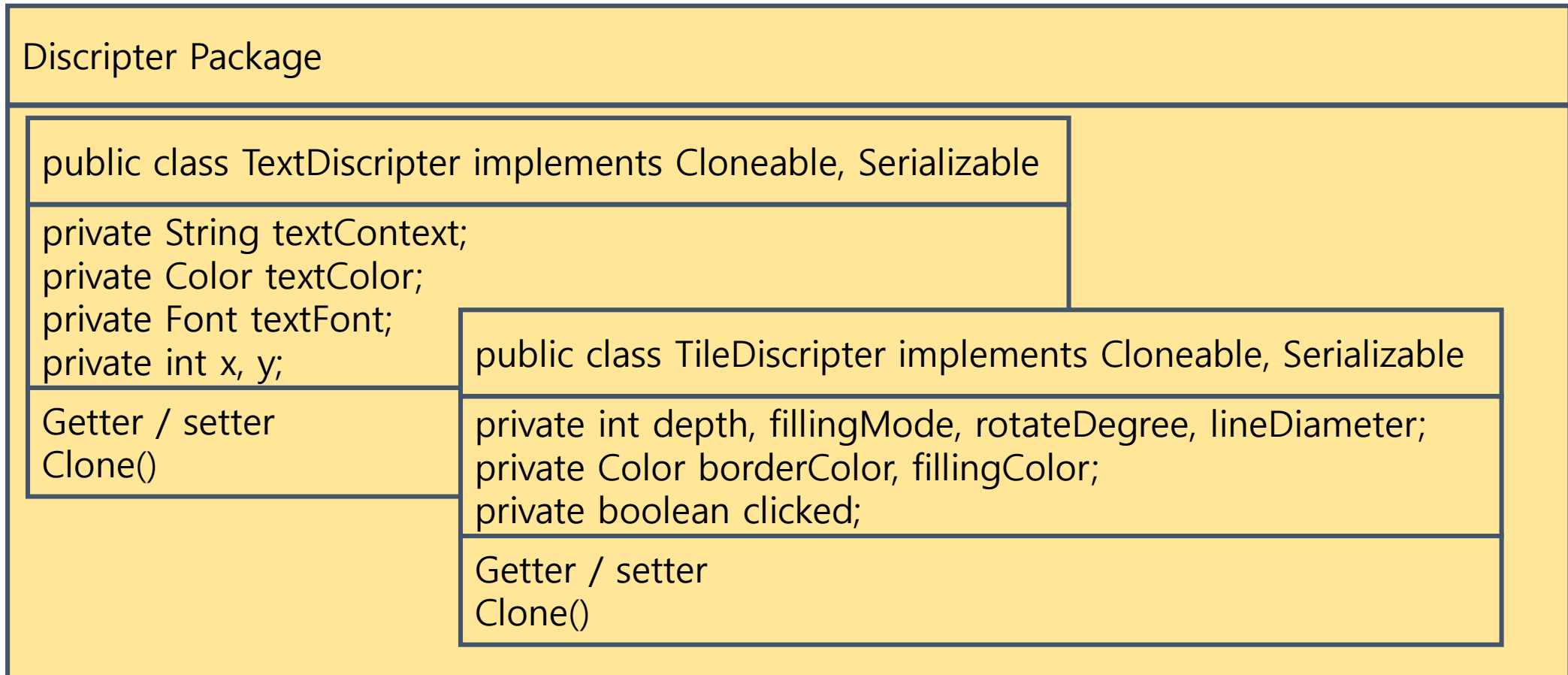
    tempX[0] = x1; tempY[0] = y1;
    tempX[1] = (int) (x1 + w * cosTemp); tempY[1] = (int) (y1 + w * sinTemp);
    tempX[2] = x2; tempY[2] = y2;
    tempX[3] = (int) (x2 - w * cosTemp); tempY[3] = (int) (y2 - w * sinTemp);
    tempX[4] = x1; tempY[4] = y1;

    TilePolygon2D pp = new TilePolygon2D(tempX,tempY,5,true);
    result = pp.contains(x,y);
}
return result;
```

# 6. 상세 설계 및 UML



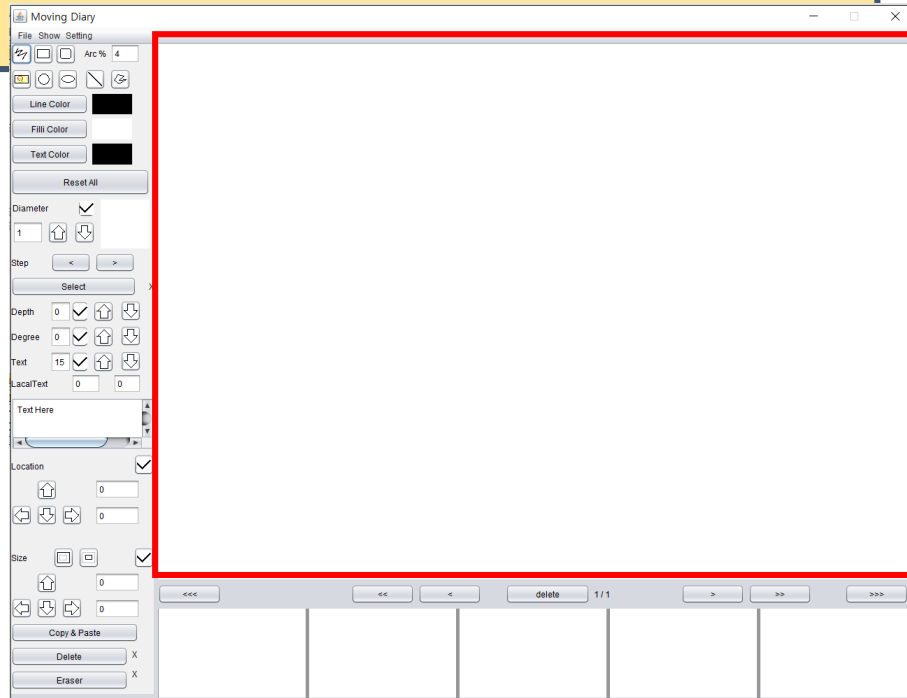
# 6. 상세 설계 및 UML



# 6. 상세 설계 및 UML

```
public class Stage extends JPanel  
implements Cloneable, Serializable
```

```
private ArrayList<Drawable> tileList;  
private int nowDrawingNumber;  
private boolean freeLine, line, rectangle, circle, oval;  
private boolean arc, roundRect, polygon, image;  
private boolean polygonDrawing;  
/**
```



```
ArrayList<Drawable> TileList
```

```
Drawable Interface
```

```
-
```

```
public void draw(Graphics g);  
public boolean contain(int x, int y);  
public boolean isClicked();  
public void setClicked(boolean clicked);  
public void drawBorder(Graphics g);  
public int getShapeType();  
public int getDepth();  
public void setDepth(int depth);
```

# 6. 상세 설계 및 UML

```
@Override
public void paint(Graphics g) {
    super.paint(g);
    for(int i = 0 ; i < getTileList().size(); i++) {
        getTileList().get(i).draw(g);
    }
    paintSelected(g);
}

private void paintSelected(Graphics g) {
    for(int i = 0 ; i < getTileList().size(); i++) {
        if(getTileList().get(i).isClicked()) {
            getTileList().get(i).drawBorder(g);
        }
    }
    if(tempD != null) {
        tempD.draw(g);
    }
}
```

```
public void saveToJPG(String fileName) {
    BufferedImage image = new BufferedImage(1000,800,BufferedImage.TYPE_INT_RGB);
    Graphics g = image.createGraphics();
    g.fillRect(0, 0, 1000, 800);
    for(int i = 0 ; i < getTileList().size(); i++) {
        getTileList().get(i).draw(g);
    }
    try {
        File file = new File(fileName);
        ImageIO.write(image, "jpeg", file);
    } catch (Exception e) {
    }
}
```

# 6. 상세 설계 및 UML

ArrayList<ArrayList<Stage>> animationList

ArrayList<Stage> stageList

```
public class Stage extends JPanel
implements Cloneable, Serializable
{
    Private ArrayList<Drawable> tileList;
    Private int nowDrawingNumber;
    Private boolean freeline, line ...

    Getter / Setter
    saveToJPG( Graphics g )
    Paint( Graphics g )
    PaintSelected(Graphics g )
    Clone()
}
```

ArrayList<Stage> stageList

```
public class Stage extends JPanel
implements Cloneable, Serializable
{
    Private ArrayList<Drawable> tileList;
    Private int nowDrawingNumber;
    Private boolean freeline, line ...

    Getter / Setter
    saveToJPG( Graphics g )
    Paint( Graphics g )
    PaintSelected(Graphics g )
    Clone()
}
```

ArrayList<Drawable> TileList

Drawable Interface

```
-

public void draw(Graphics g);
public boolean contain(int x, int y);
public boolean isClicked();
public void setClicked(boolean clicked);
public void drawBorder(Graphics g);
public int getShapeType();
public int getDepth();
public void setDepth(int depth);
```

# 6. 상세 설계 및 UML

```
public class MainFrame extends JFrame implements  
MouseListener, MouseMotionListener
```

```
public final static int UPDATE_TIME_VAR = 20;  
private boolean selectState, removeState, eraserState,  
polygonState, imageSlected, largerState, smallerState,  
selected;
```

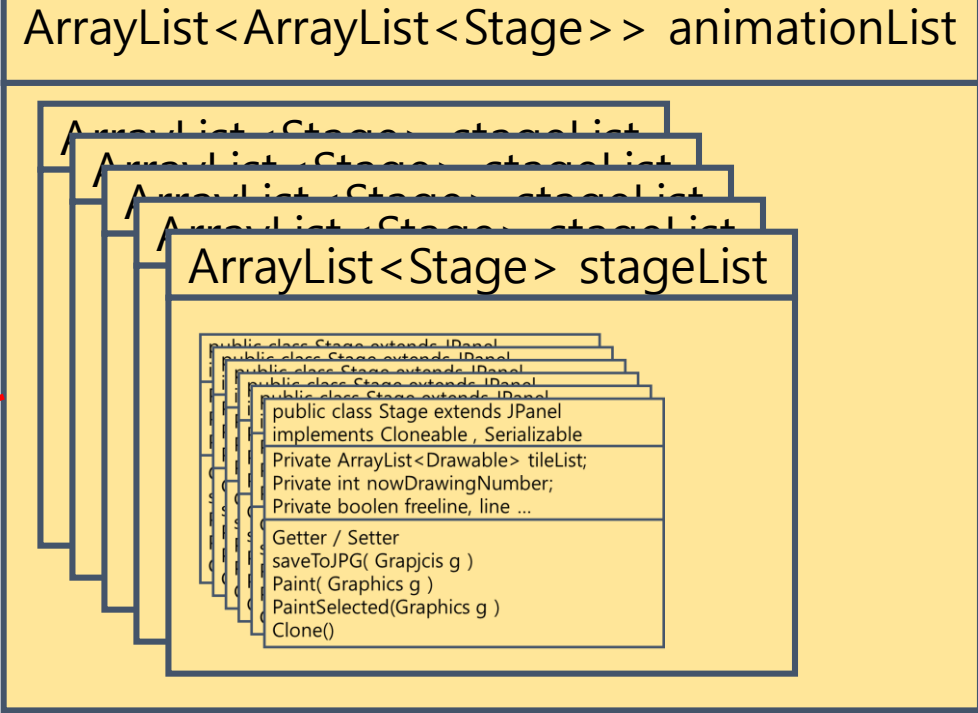
```
private int stageNumber = 0;  
private int maxStageNumber = 1;  
private int animationStageNumber = 0;  
private int animationMaxStageNumber = 1;  
// 그 외 Drag 중 필요한 임시 값들...  
private ArrayList<Stage> stageList;  
private ArrayList<ArrayList<Stage>> animationList;  
private animationStage[] as = new animationStage[5];  
// 그 외 GUI 도구들
```

```
Getter / Setter / ActionListener  
addStage(), addStage( Stage stage ) minusStage()  
saveDiary(String fileExtension) loadDiary(String fileExtension)  
setDownStage(), setPastState(),
```

```
ArrayList<ArrayList<Stage>> animationList
```

```
ArrayList<Stage> stageList
```

```
public class Stage extends JPanel  
implements Cloneable, Serializable  
Private ArrayList<Drawable> tileList;  
Private int nowDrawingNumber;  
Private boolean freeline, line ...  
  
Getter / Setter  
saveToJPG( Graphics g )  
Paint( Graphics g )  
PaintSelected(Graphics g )  
Clone()
```



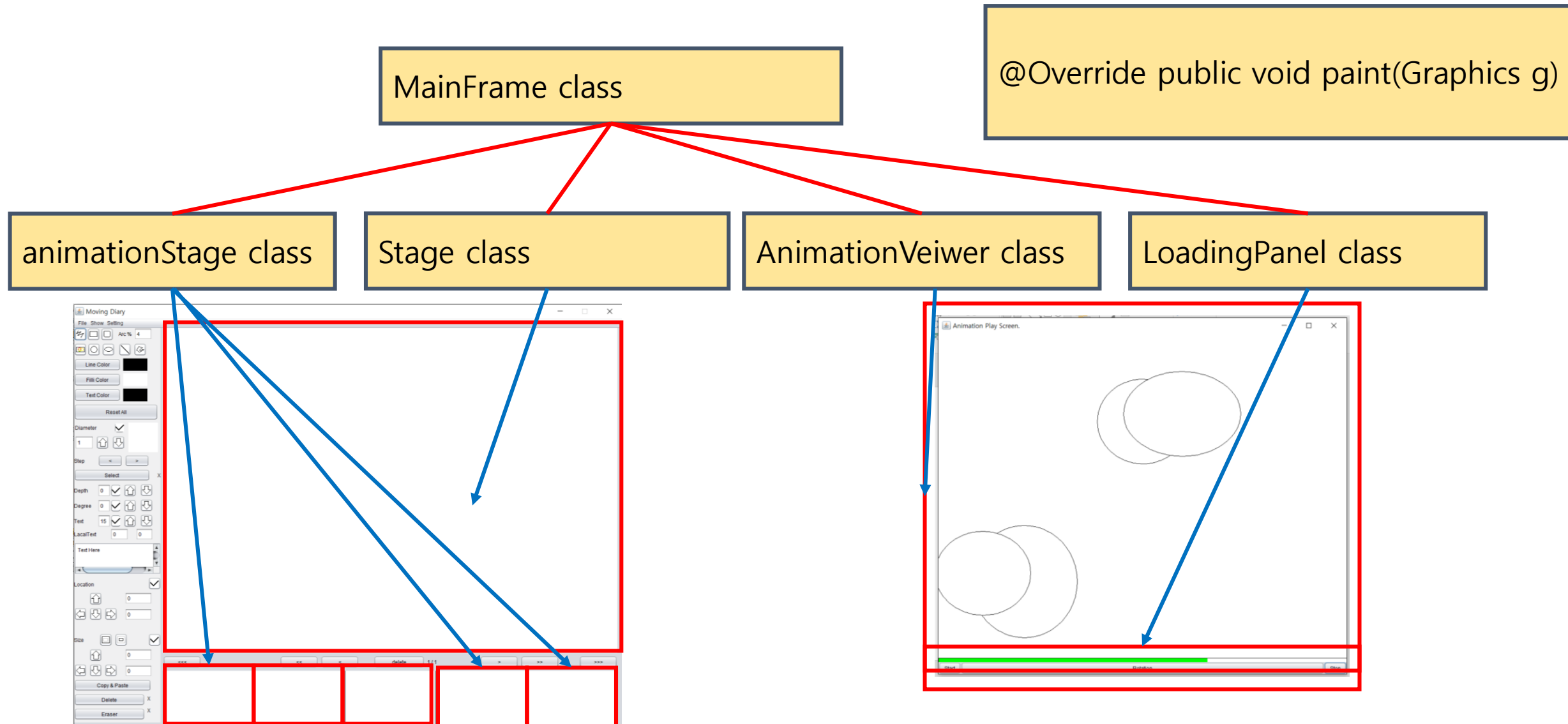


# 6. 상세 설계 및 UML

더 많은 내용이 궁금하신가요...?

```
public class MainFrame extends javax.swing.JFrame implements MouseMotionListener, MouseListener {  
    public final static int UPDATE_TIME_VAR = 20;  
  
    private int firstX, firstY;  
    private int secondX, secondY;  
    private int draggedX = 0;  
    private int draggedY = 0;  
    private int selcetNumber;  
    private int lineDiameter = 1;  
    private boolean selectState, removeState, eraserState, polygonState;  
    private boolean imageSlected;  
    private boolean selected;  
    private Drawable tempShape;  
    private BufferedImage bufferedImage;  
    private ArrayList<Integer> xList = new ArrayList<Integer>();  
    private ArrayList<Integer> yList = new ArrayList<Integer>();  
    private ArrayList<Integer> Xpoint = new ArrayList<Integer>();  
    private ArrayList<Integer> Ypoint = new ArrayList<Integer>();  
    // Drawing State, Not have to serialize  
  
    private int stageNumber = 0;  
    private int maxStageNumber = 1;  
    private int animationStageNumber = 0;  
    private int animationMaxStageNumber = 1;  
    private ArrayList<Stage> stageList;  
    private ArrayList<ArrayList<Stage>> animationList;  
    private animationStage[] as = new animationStage[5];  
    private Stage stage;  
    // Drawable State, Have to Serializing
```

# 6. 상세 설계 및 UML



# 6. 상세 설계 및 UML

```
public class AnimationVeiwer extends JPanel
```

```
private ArrayList<ArrayList<Drawable>>  
specifiedTile;
```

```
private ArrayList<Stage> stageList
```

```
private boolean rotation, stopped;
```

```
private int nowVeiwNumber;
```

```
private int maxVeiwNumber;
```

```
Getter / Setter
```

```
makeSpecifiedTile()
```

```
Paint(Graphics g)
```

```
public ArrayList<ArrayList<Drawable>> makeSpecifiedTile() {  
    ArrayList<ArrayList<Drawable>> specifiedTileList = new ArrayList<ArrayList<Drawable>>();  
  
    for(int i = 0 ; i < maxVeiwNumber ; i++) {  
        specifiedTileList.add(new ArrayList<Drawable>());  
    }  
  
    for(int i = 0 ; i < stageList.size() - 1; i++) {  
        ArrayList<Drawable> oldTileList = stageList.get(i).getTileList();  
        ArrayList<Drawable> newTileList = stageList.get(i + 1).getTileList();  
  
        for(int k = 0; k < oldTileList.size(); k++) {  
            int changedX, changedY, changedR;  
            int changedMX, changedMY;  
            int changedBorderR, changedBorderG, changedBorderB;  
            int changedFillingR, changedFillingG, changedFillingB;  
            switch(oldTileList.get(k).getShapeType()) {  
                case 2 : // TileEllipse2D  
                    // minX , minY ' s change amount can be the indicator of specified stage.  
                    // rotate ' s change amount will be accessible with no specified value  
                    TileEllipse2D oldTe2D = (TileEllipse2D)oldTileList.get(k);  
                    TileEllipse2D newTe2D = (TileEllipse2D)newTileList.get(k);  
  
                    changedX = (int)(newTe2D.getMinX() - oldTe2D.getMinX());  
                    changedY = (int)(newTe2D.getMinY() - oldTe2D.getMinY());  
                    changedR = (int)(newTe2D.getTd().getRotateDegree() - oldTe2D.getTd().getRotateDegree());  
                    changedMX = (int)(newTe2D.getMaxX() - oldTe2D.getMaxX());  
                    changedMY = (int)(newTe2D.getMaxY() - oldTe2D.getMaxY());  
  
                    changedX = (int)((double)changedX / 10);  
                    changedY = (int)((double)changedY / 10);  
                    changedR = (int)((double)changedR / 10);  
                    changedMX = (int)((double)changedMX / 10);  
                    changedMY = (int)((double)changedMY / 10);  
  
                    changedBorderR = newTe2D.getTd().getBorderColor().getRed() - oldTe2D.getTd().getBorderColor().getRed();  
                    changedFillingR = newTe2D.getTd().getFillingColor().getRed() - oldTe2D.getTd().getFillingColor().getRed();  
  
                    changedBorderR = (int)((double)changedBorderR/10);  
                    changedFillingR = (int)((double)changedFillingR/10);  
  
                    changedBorderG = newTe2D.getTd().getBorderColor().getGreen() - oldTe2D.getTd().getBorderColor().getGreen();  
                    changedFillingG = newTe2D.getTd().getFillingColor().getGreen() - oldTe2D.getTd().getFillingColor().getGreen();  
  
                    changedBorderG = (int)((double)changedBorderG/10);  
                    changedFillingG = (int)((double)changedFillingG/10);  
            }  
        }  
        specifiedTileList.get(i).addAll(newTileList);  
    }  
}
```

# 6. 상세 설계 및 UML

```
for(int j = 0; j < 10; j++) {  
    specifiedTileList.get(i * 10 + j).add(  
new TileEllipse2D(  
(int)oldTe2D.getMinX() + changedX * j,  
(int)oldTe2D.getMinY() + changedY * j,  
(int)oldTe2D.getMaxX() + changedMX * j,  
(int)oldTe2D.getMaxY() + changedMY * j,  
(int)oldTe2D.getTd().getRotateDegree() + changedR * j,  
oldTe2D.getTd().getDepth(),  
oldTe2D.getTtd().getTextContext(),  
oldTe2D.getTtd().getX() + changedX * j,  
oldTe2D.getTtd().getY() + changedY * j,  
oldTe2D.getTtd().getTextFont().getFontName(),  
oldTe2D.getTtd().getTextFont().getStyle(),  
oldTe2D.getTtd().getTextFont().getSize(),  
oldTe2D.getTtd().getTextColor() ,  
new Color(oldTe2D.getTd().getBorderColor().getRed() + changedBorderR * j, oldTe2D.getTd().getBorderColor().getGreen() + changedBorderG * j ,  
oldTe2D.getTd().getBorderColor().getBlue() + changedBorderB * j),  
new Color(oldTe2D.getTd().getFillingColor().getRed() + changedFillingR * j, oldTe2D.getTd().getFillingColor().getGreen() + changedFillingG * j ,  
oldTe2D.getTd().getFillingColor().getBlue() + changedFillingB * j )  
)  
);  
  
    }  
break;
```

# 6. 상세 설계 및 UML

```
public class animationStage extends JPanel
```

```
private Stage thisStage;
```

```
addingStage()  
getThisStage(), setThisStage()  
Paint(Graphics g)
```

```
public Stage getThisStage() {  
    return this.thisStage;  
}  
  
public void addingStage() {  
    super.add(getThisStage(), BorderLayout.CENTER);  
}  
  
@Override  
public void paint(Graphics g) {  
    // 1072 , 766 Animation width height  
    // 1072 , 127 -> /5 /1 : 214.4 , 127  
    super.paint(g);  
    g.setColor(Color.WHITE);  
    g.fillRect(0, 0, 214, 127);  
    Graphics2D g2d = (Graphics2D)g;  
  
    g2d.scale(0.195, 0.165);  
  
    for(int i = 0 ; i < getThisStage().getTileList().size(); i++) {  
        getThisStage().getTileList().get(i).draw(g2d);  
    }  
  
    g2d.scale(5.10476, 6.03149);  
}  
}
```

# 6. 상세 설계 및 UML

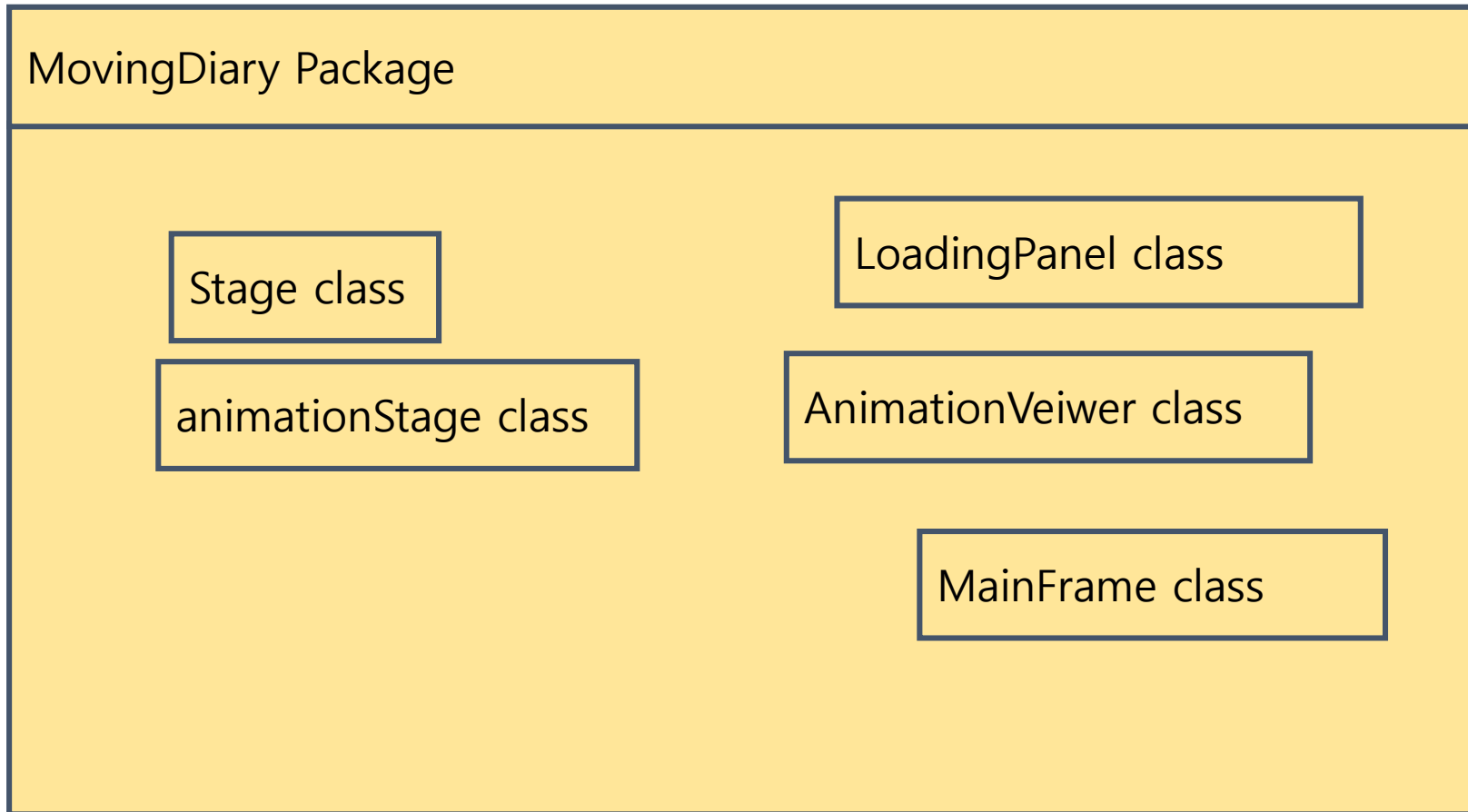
```
public class LoadingPanel extends JPanel
```

```
private int max;  
private int width;  
private int now
```

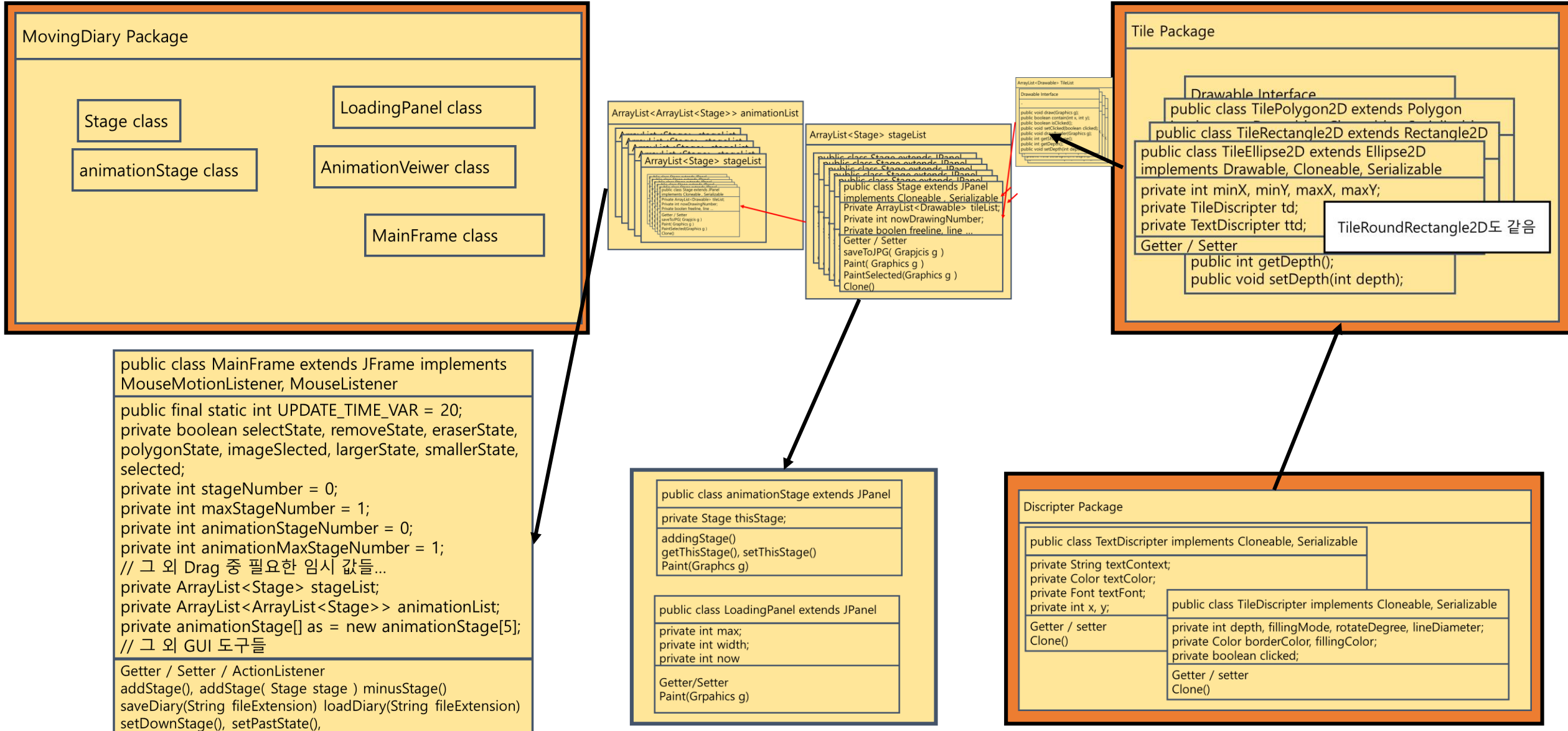
```
Getter/Setter  
Paint(Graphics g)
```

```
public int getNow() {  
    return this.now;  
}  
  
public void setMax(int max) {  
    this.max = max;  
}  
  
public void setNow(int now) {  
    this.now = now;  
}  
  
@Override  
public void paint(Graphics g) {  
    super.paint(g);  
    double result = ((double)getNow() / (double)getMax()) * (double)this.width;  
    g.setColor(Color.WHITE);  
    g.fillRect(0, 0, width, 30);  
    g.setColor(Color.GREEN);  
    g.fillRect(0, 0, (int)result, 30);  
    g.setColor(Color.BLACK);  
    g.drawRect(0, 0, width, 30);  
}
```

# 6. 상세 설계 및 UML



# 6. 상세 설계 및 UML





# 7. 해결 한 어려웠던 점

이미 만들어져 있는 Polygon2D, Rectangle2D 등을 상속 받으면서도 그림을 그릴 수 있게 해주는 클래스를 상속 받고 싶은 설계 문제 해결.

Heap 공간 부족 문제 해결.

애니메이션 출력 시 화면 깜빡임 문제 해결.

BufferedImage 의 Serializable 문제 해결

Implements clonable 에서 clone() 함수의 return 값 문제 해결.

클릭 한 곳에 어떤 타일이 있는가 문제 해결

# 7. 해결 한 어려웠던 점

회전 시 자신의 중심 기준으로 한 것이 아니라 왼쪽 위의 원점을 기준으로 회전하는 점 해결.

ArrayList.size() 까지 for 문을 돌리면서 1 번째 것을 지우는 것에서 오류 해결.

ArrayList 값 복사와 레퍼런스 복사 문제 해결.

Paint 가 재정의 되어야하는 panel 들 재정의로 문제 해결.

저장 후 불러오기에서 이미지가 위치가 변하면 이미지가 깨지는 문제 해결.

-

# 7. 해결한 어려웠던 점



ArrayList.size() 까지 for 문을 돌리면서 1 번째 것을 지우는 것에서 오류 해결.

```
ArrayList al 이 있다고 할 때  
For( int i = some_number; i < al.size(); i++) {  
    al.remove( i );  
}
```

에서 문제는..?

# 8. 해결 못 한 어려웠던 점

이미지 저장 시 BufferedImage 가 직렬화 되지 않아서 예외 발생

네트워크 서버 IP 가 0.0.0.0 : 8000 으로 뜨던 문제

다른 프로젝트와 결합하려 했으나 시간이 부족  
<http://codegiraffe.iptime.org#Savefile>

기능이 구현되지 않은 버튼들이 존재

MainFrame Class 에서 View 와 Control의 혼재가 있었음. Data 접근성 때문에 어려웠음.

-