



# FROGGERS

BY  
„DIE UMELS“

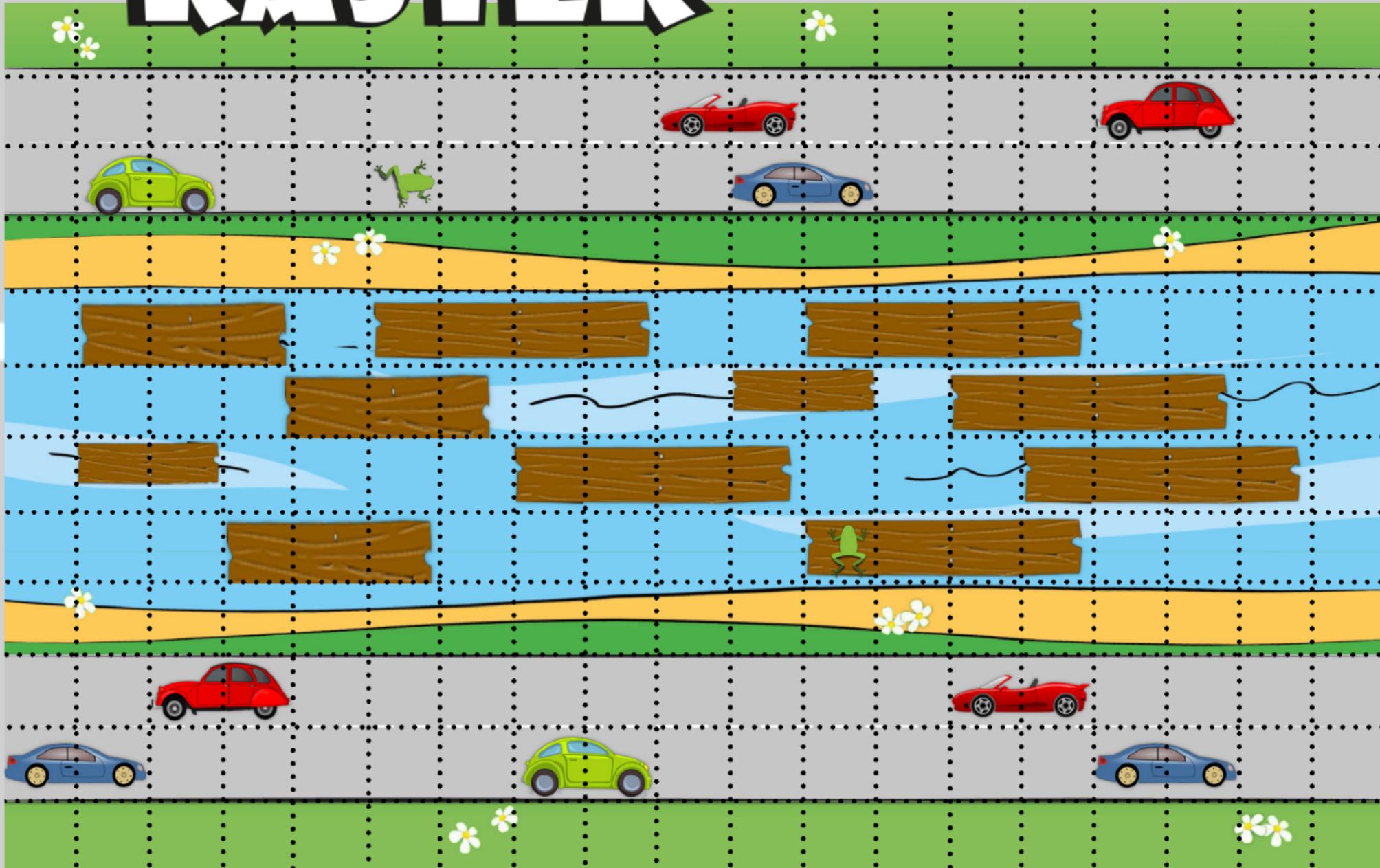
# DIE 3 UMEIS

**MANUEL BOGUS**

**MIKE JASTROW**

**SALIM OUSSAYFI**

# RASTER

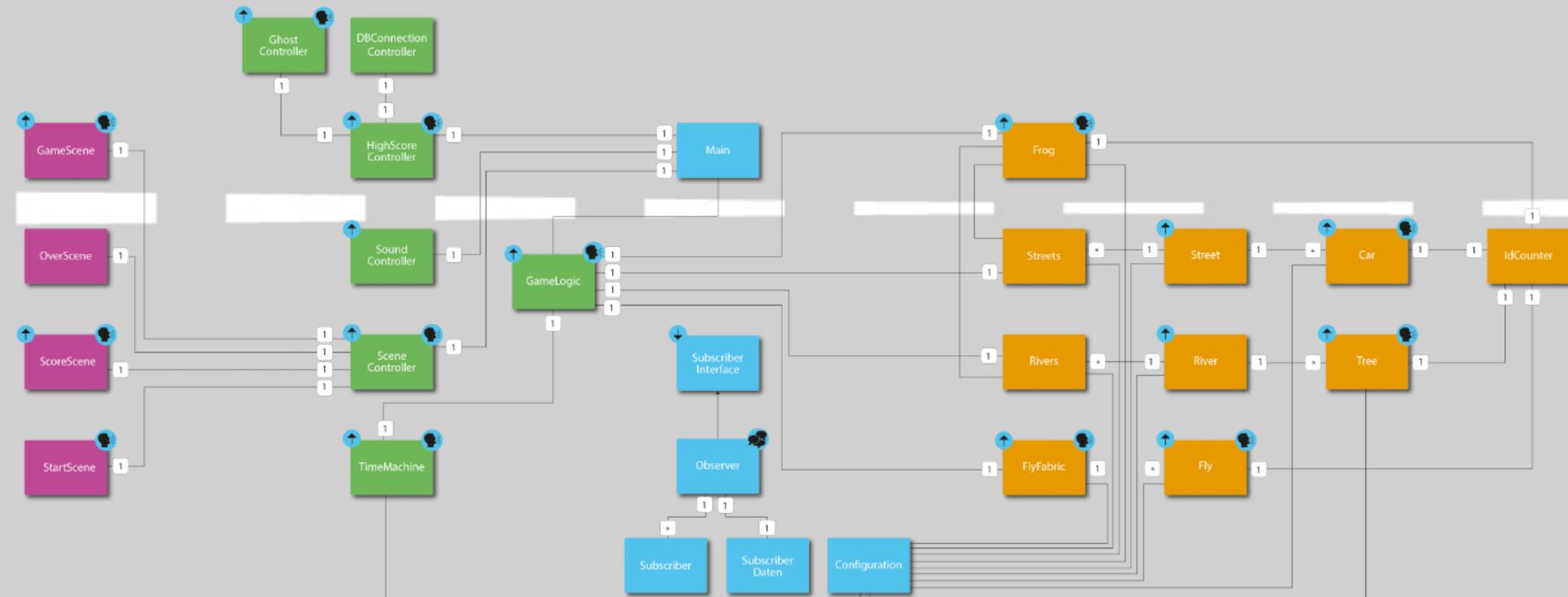


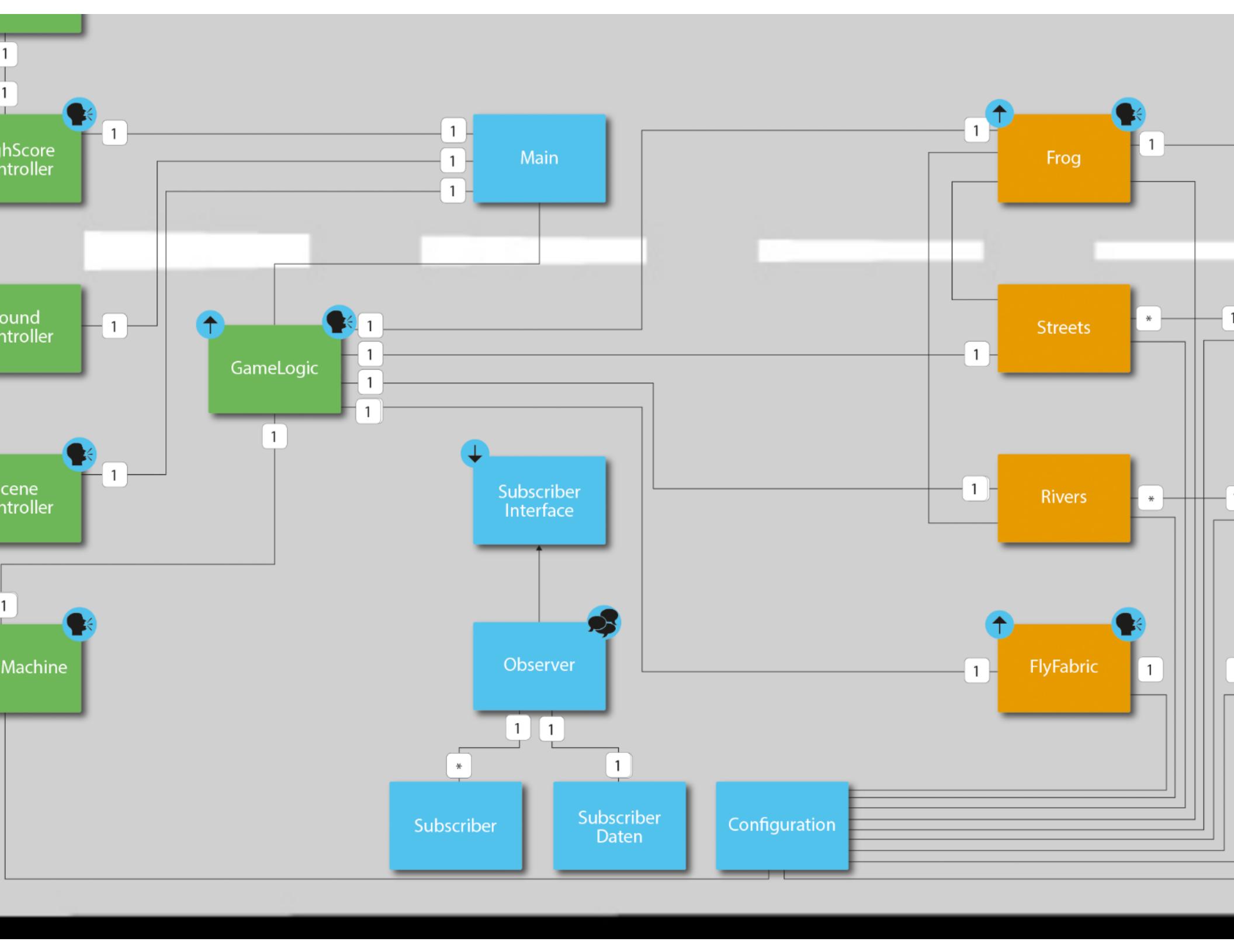
views

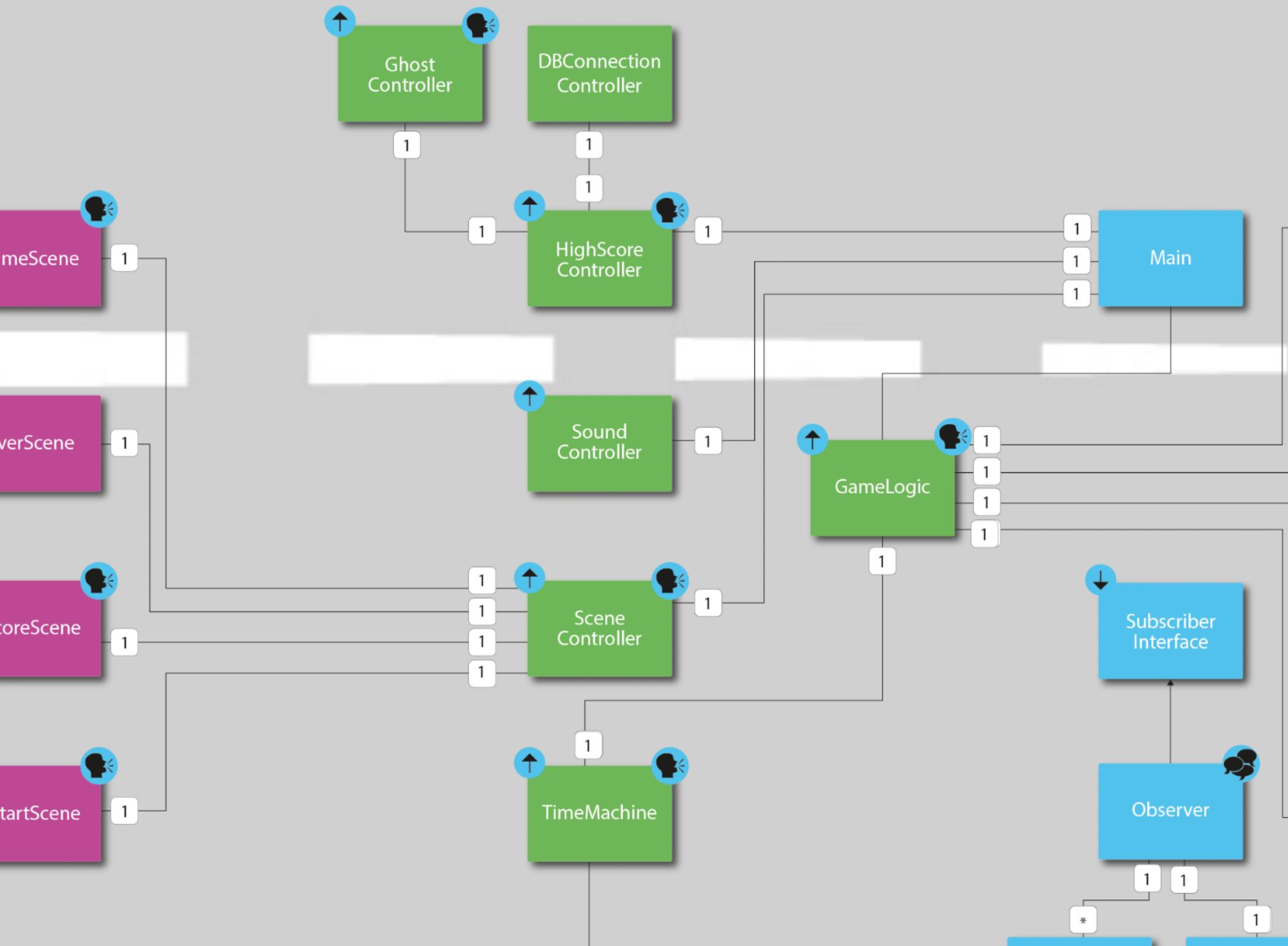
controller

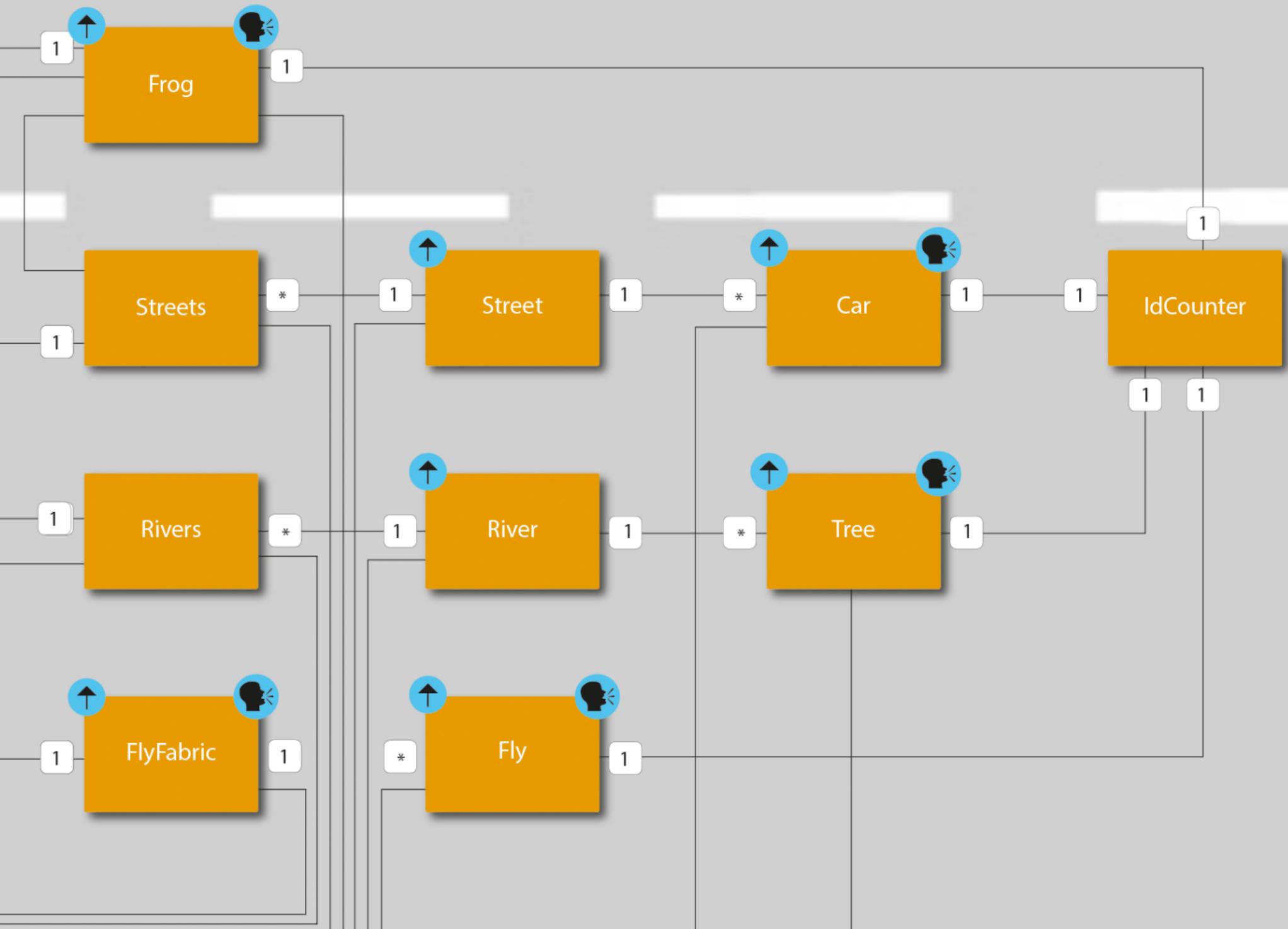
application

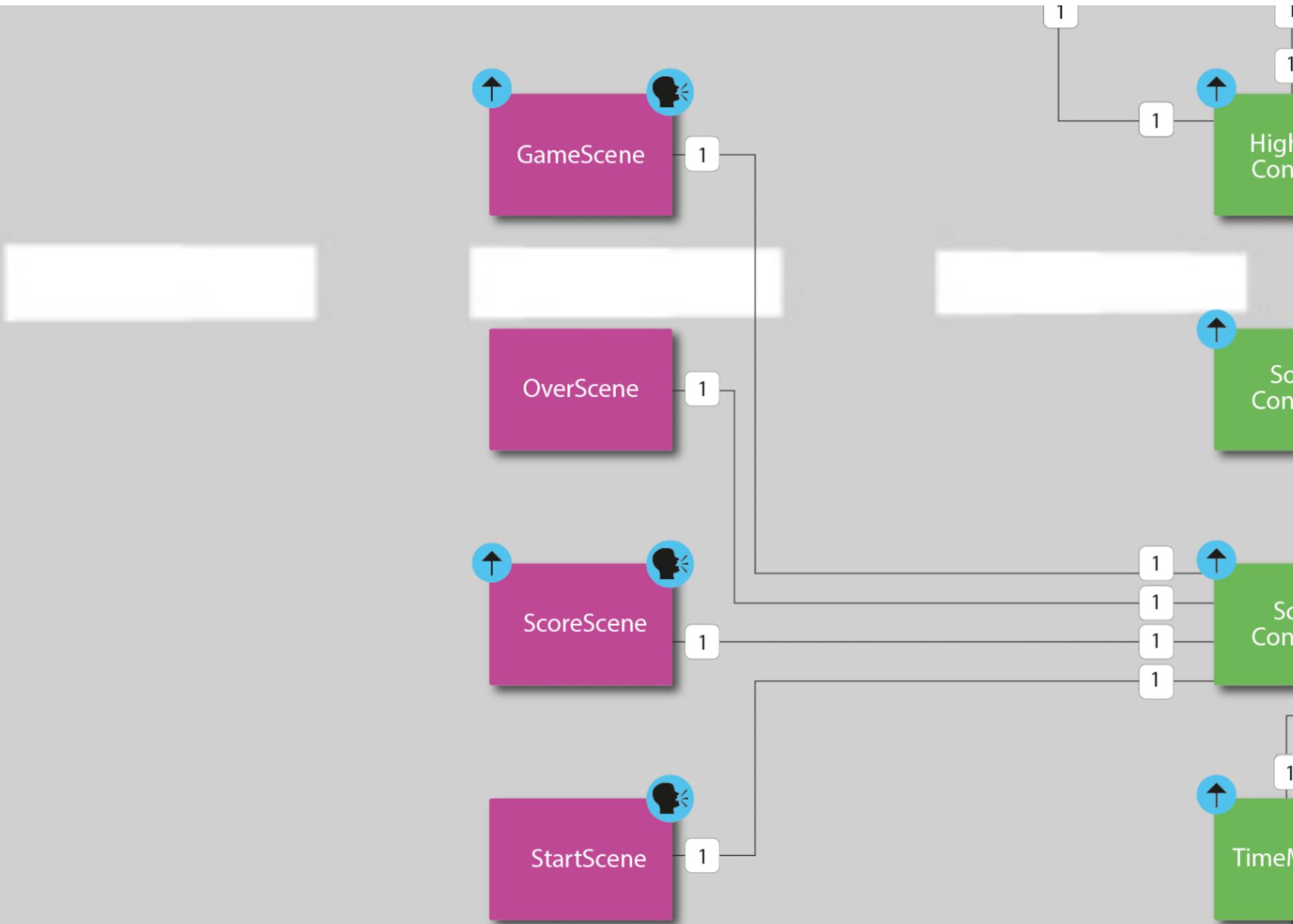
model









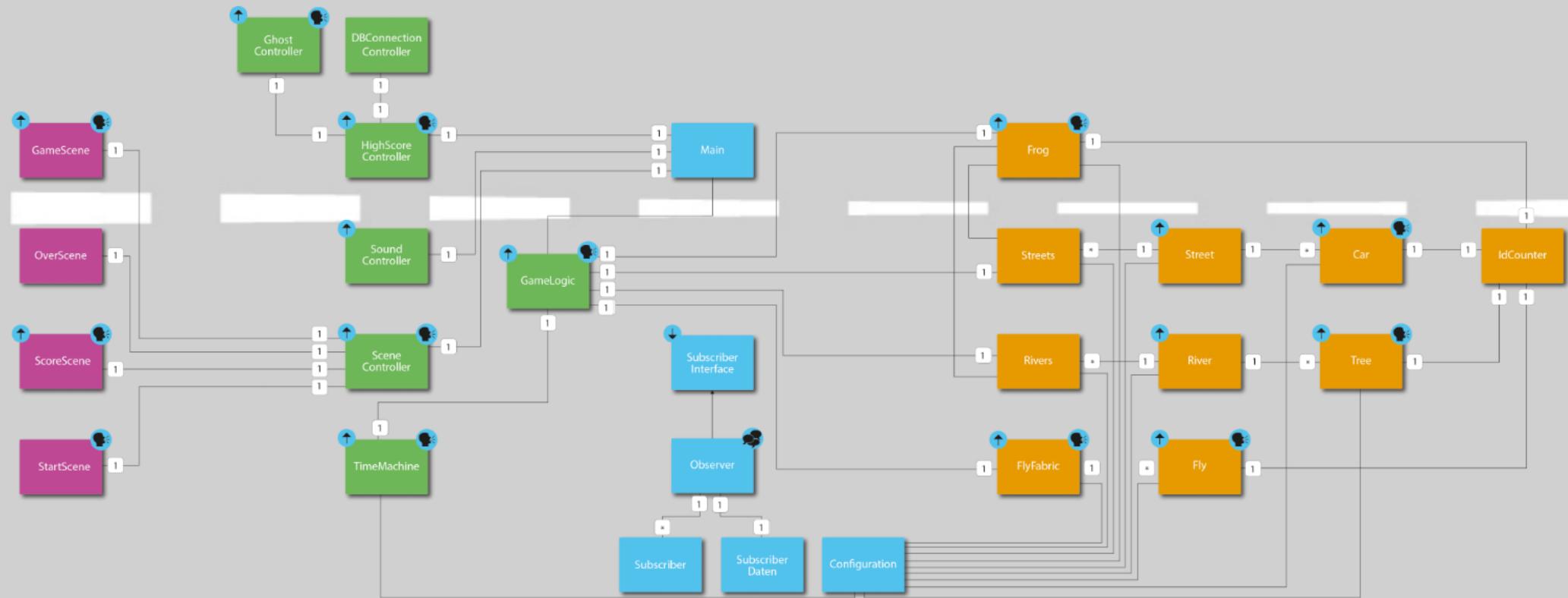


views

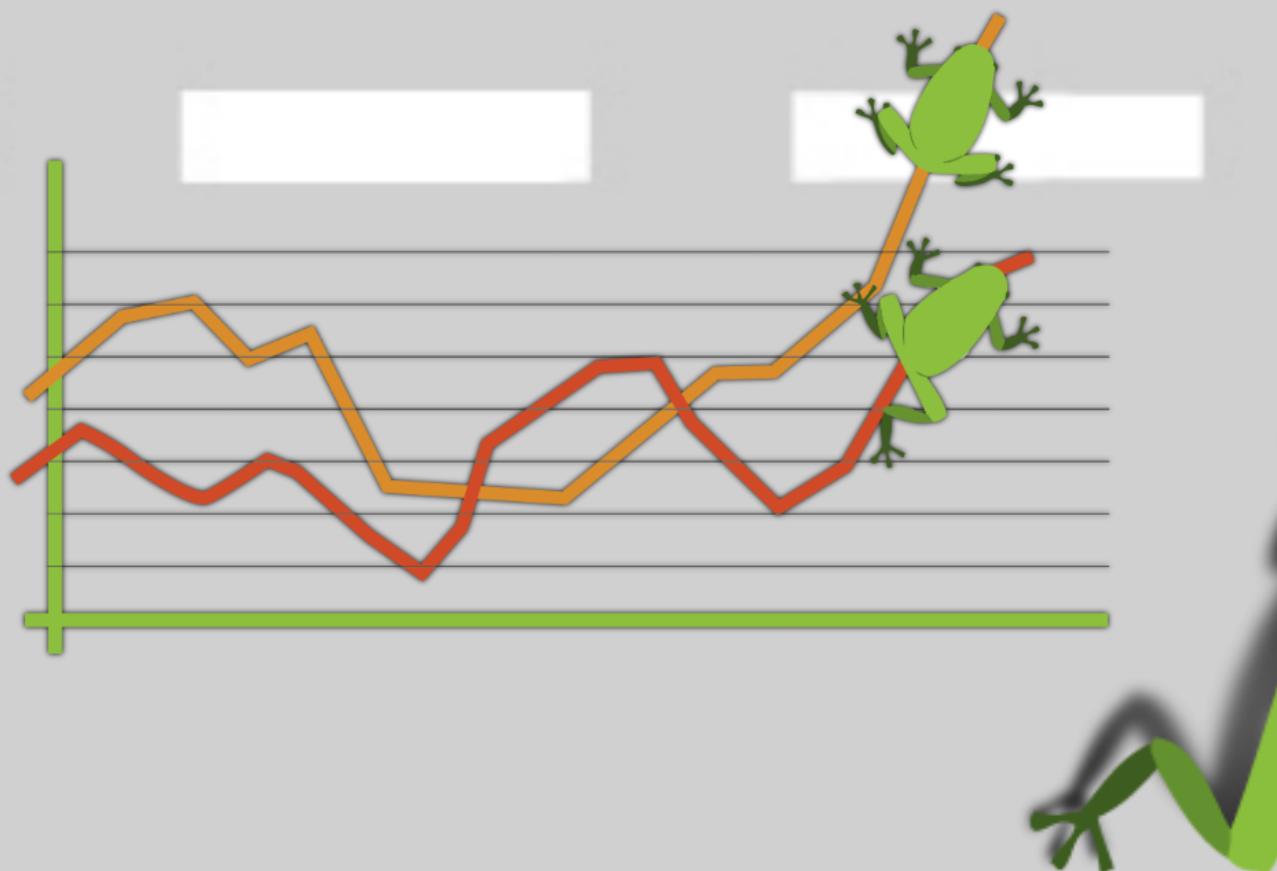
controller

application

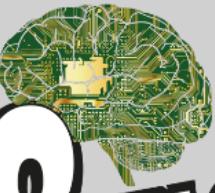
model



# STATISTIKEN



ÜBER **4.000**  
ZEILEN CODE



**156** @  
**EMAILS**



**210** WhatsApp  
NACHRICHTEN



**19** S Skype  
MEETINGS



# GITHUB

Clear current search query, filters, and sorts	
<input type="checkbox"/>	① 3 Open ✓ 26 Closed
<input type="checkbox"/>	Ⓜ Mneü blockieren wenn Spiel läuft #27 by jastrow was closed 6 days ago
<input type="checkbox"/>	Ⓜ SceneController anpassen #26 by chaostype was closed 7 days ago 1 of 1
<input type="checkbox"/>	Ⓜ OverScene #25 by chaostype was closed 6 days ago 5 of 5
<input type="checkbox"/>	Ⓜ Frog mounts Tree #24 by jastrow was closed 7 days ago
<input type="checkbox"/>	Ⓜ THIS ISSUE WAS DELETED AND BLOCKED #23 by slimouGit was closed 9 days ago
<input type="checkbox"/>	Ⓜ W.A.S.D ergänzen #22 by jastrow was closed 9 days ago
<input type="checkbox"/>	Ⓜ Exception: Frog Move nach tot #21 by jastrow was closed 8 days ago
<input type="checkbox"/>	Ⓜ Layers in Views - Froschbild unter Cars und/oder Trees #20 by jastrow was closed 8 days ago
<input type="checkbox"/>	Ⓜ Trigger: stopGame #18 by jastrow was closed 7 days ago 0 of 7
<input type="checkbox"/>	Ⓜ Trigger: resetGame #17 by jastrow was closed 7 days ago 0 of 5
<input type="checkbox"/>	Ⓜ MultiThread tauglichen Counter oder FX-Elemente #16 by jastrow was closed 10 days ago
<input type="checkbox"/>	Ⓜ Observer Subscriber - List = MIST #15 by jastrow was closed 10 days ago
<input type="checkbox"/>	Ⓜ StartScene anpassen #14 by chaostype was closed 7 days ago 4 of 4
<input type="checkbox"/>	Ⓜ notwendige ObserverTrigger #13 by chaostype was closed 7 days ago 4 of 4
<input type="checkbox"/>	Ⓜ Streets Street Car #12 by jastrow was closed 9 days ago
<input type="checkbox"/>	Ⓜ Frosch #11 by jastrow was closed 7 days ago 5 of 5
<input type="checkbox"/>	Ⓜ View & Gamelogic #10 by jastrow was closed 7 days ago
<input type="checkbox"/>	Ⓜ ServerController #9 by jastrow was closed 6 days ago
<input type="checkbox"/>	Ⓜ Server API #8 by jastrow was closed 6 days ago
<input type="checkbox"/>	Ⓜ Rivers River Trees #7 by jastrow was closed 9 days ago
<input type="checkbox"/>	Ⓜ HighScoreScene anpassen

32 ISSUES  
247 COMMITS





**PROBLEMZONEN**

# MULTITHREADING QUEUE

```
@Override
public void run() {
    while(this.running) {
        try {
            Thread.sleep( Configuration.timeSteps );
        } catch (InterruptedException e) {
            e.printStackTrace();
        }

        SubscriberDaten daten = new SubscriberDaten();
        daten.name = "time";
        daten.time = this.timeLog;

        this.timeLog += Configuration.timeSteps;

        if(this.timeLog.compareTo((this.timeEnd - 100)) >= 0) {
            this.running = false;
            Platform.runLater(new Runnable() {
                public void run() {
                    Observer.trigger("timeKilledFrog", daten);
                }
            });
        } else {
            Platform.runLater(new Runnable() {
                public void run() {
                    Observer.trigger("time", daten);
                }
            });
        }
    }
}
```

**TIMEMACHINE**

# FROG ON TREE

```
public void calling(String trigger, SubscriberDaten data) {
    if(trigger == "car" || trigger == "tree") {
        this.collisionCheck();
    }
}
```

FROG

```
public void collisionCheck() {
    this.frogOnTreeId = this.rivers.collisionCheck(this.positionX, this.positionXend, this.positionY);
    if(this.frogOnTreeId == 0) {
        this.killed = true;
        this.triggerObserver("Killed");
        Observer.trigger("stopGame", new SubscriberDaten());
    }
}
```

FROG

```
public Integer collisionCheck(Integer positionX, Integer positionXend, Integer positionY) {
    for(River river: this.riverlines) {
        if(river.getPositionY().compareTo(positionY) == 0) {
            return river.collisionCheck(positionX, positionXend);
        }
    }
    return -1;
}
```

RIVERS

```
public Integer collisionCheck(Integer positionX, Integer positionXend) {
    Integer treeStart;
    Integer treeEnd;
    for(Tree tree: this.trees) {
        treeStart = tree.getPositionX();
        treeEnd = tree.getPositionXend();
        if(
            (positionX.compareTo(treeStart) > 0 && positionX.compareTo(treeEnd) < 0)
            ||
            (positionXend.compareTo(treeStart) > 0 && positionXend.compareTo(treeEnd) < 0)
        ) {
            return tree.getId();
        }
    }
    return 0;
}
```

RIVER

```
public void calling(String trigger, SubscriberDaten data) {
    if(trigger == "tree") {
        if(data.id.equals(this.frogOnTreeId)) {
            if(data.leftToRight) {
                this.moveOnTree(data);
            } else {
                this.moveOnTree(data);
            }
        }
    }
}
```

FROG

```
private void moveOnTree(SubscriberDaten data) {
    Integer newX;
    if(data.leftToRight) {
        newX = this.positionX + data.lastMovementDistanceX;
    } else {
        newX = this.positionX - data.lastMovementDistanceX;
    }
    this.checkGameZoneMove(newX, this.positionY);
}
```

FROG



# CANVAS

```
//Sammelliste fuer GUI Elemente
private Queue<ImageView> frogs = new ConcurrentLinkedQueue<ImageView>();
private Queue<ImageView> pictureCont = new ConcurrentLinkedQueue<ImageView>();

private void createNewCarObject(SubscriberDaten data) {
    //Hilfsvariablen deklarieren
    ImageView help = this.getGUIObject(data);

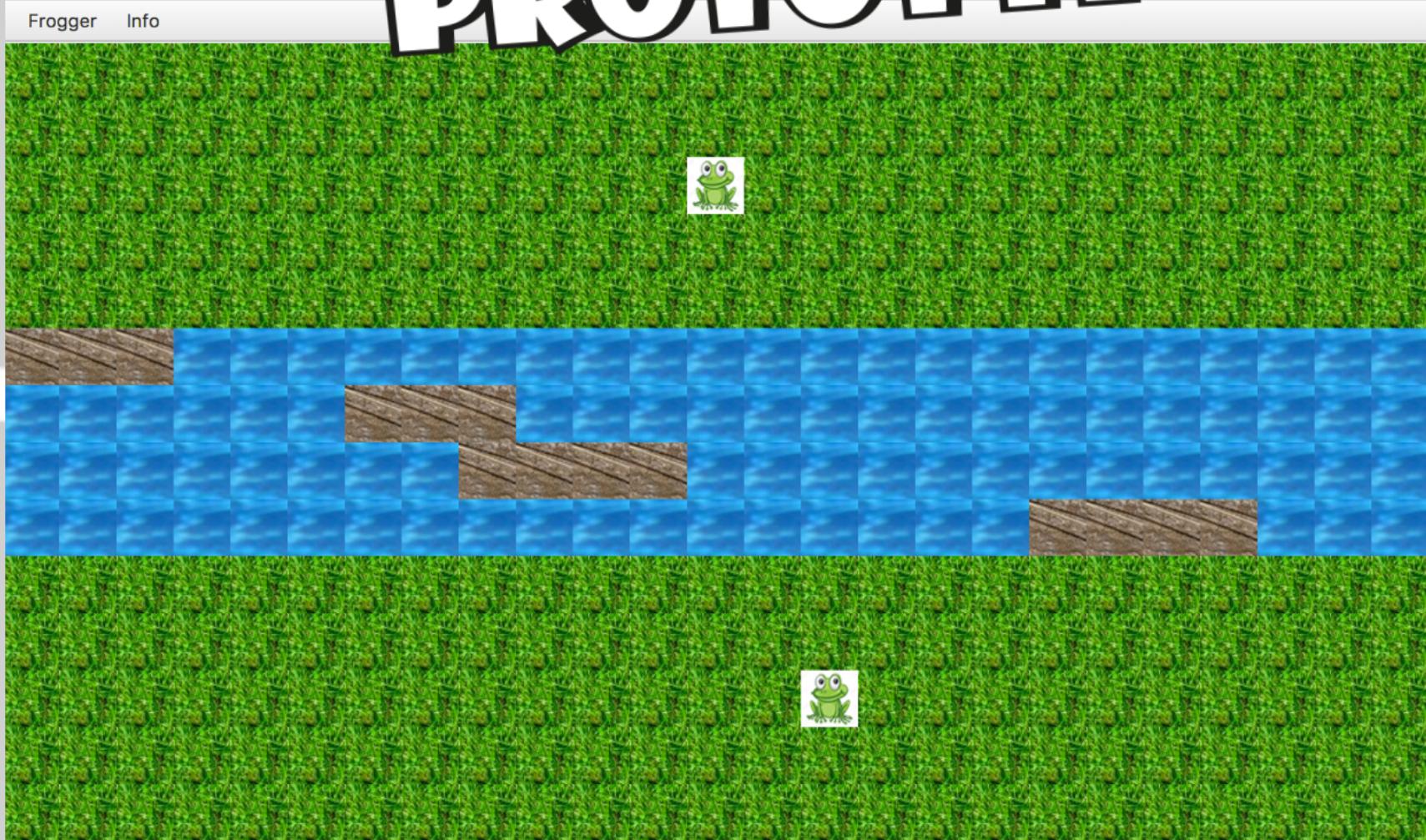
    if (data.leftToRight) {
        help.setImage(this.carLeftToRight[rand.nextInt(2)]);
    } else {
        help.setImage(this.carRightToLeft[rand.nextInt(2)]);
    }
    help.setFitHeight(50);
    help.setId(data.id.toString());
    this.pictureCont.add(this.setPosition(help, data));
    this.updateElements();
}

private void updateElements() {
    //Szene leeren
    this.graphicsContext.clearRect(0, 0, 950, 600);
    //Bauume/Autos in GUI setzen
    for(ImageView help: this.pictureCont){
        this.graphicsContext.drawImage(help.getImage(), help.getX(), help.getY());
    }
    //Frogs in GUI setzen
    for(ImageView help: this.frogs){
        this.graphicsContext.drawImage(help.getImage(), help.getX(), help.getY());
    }
}
```

**GAMESCENE**



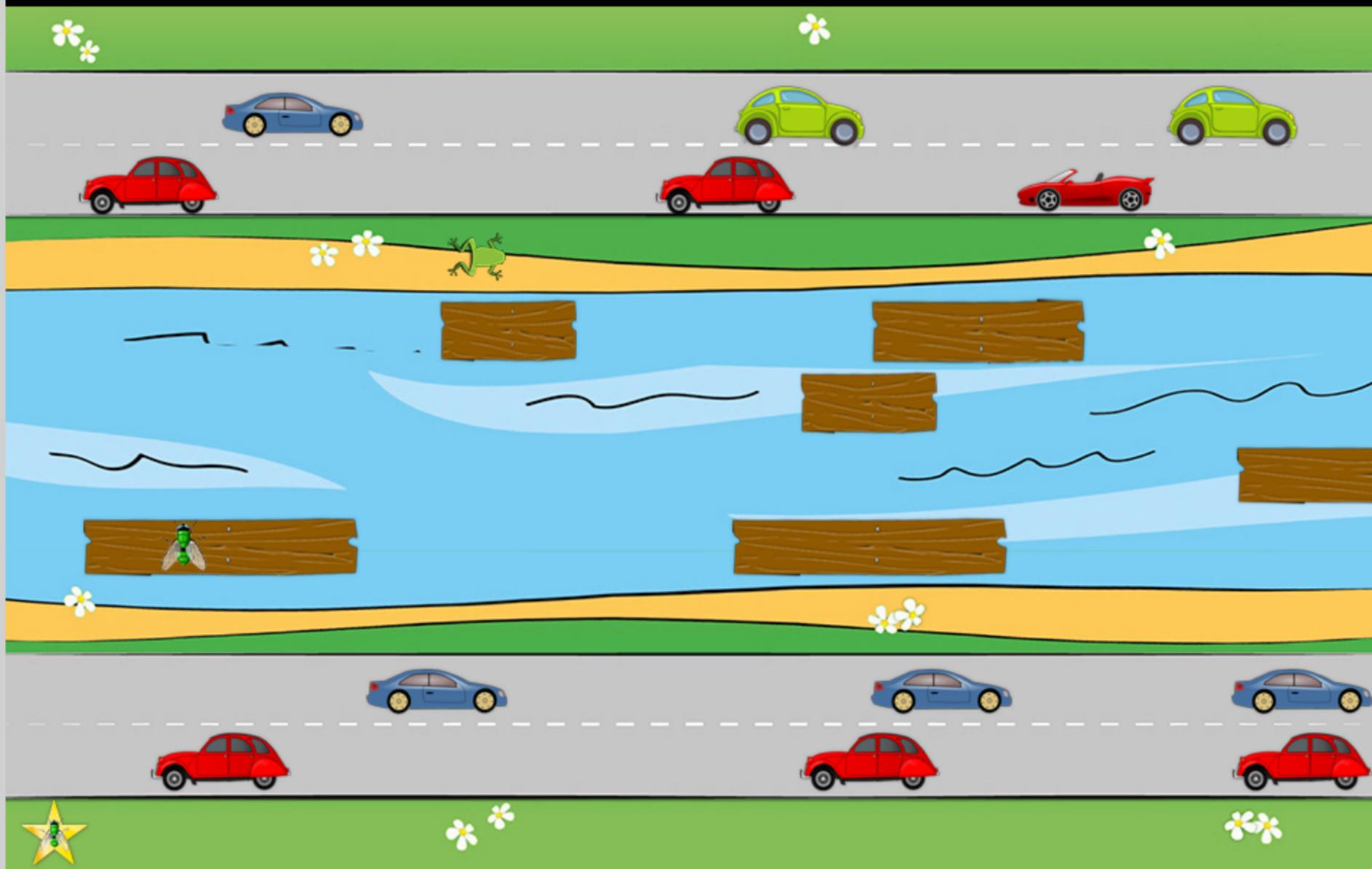
# PROTOTYP



# FROGGERS

Frogger Info

Punkte: 03920 - Restliche Spielzeit: 34 Sek.





VIelen Dank!

