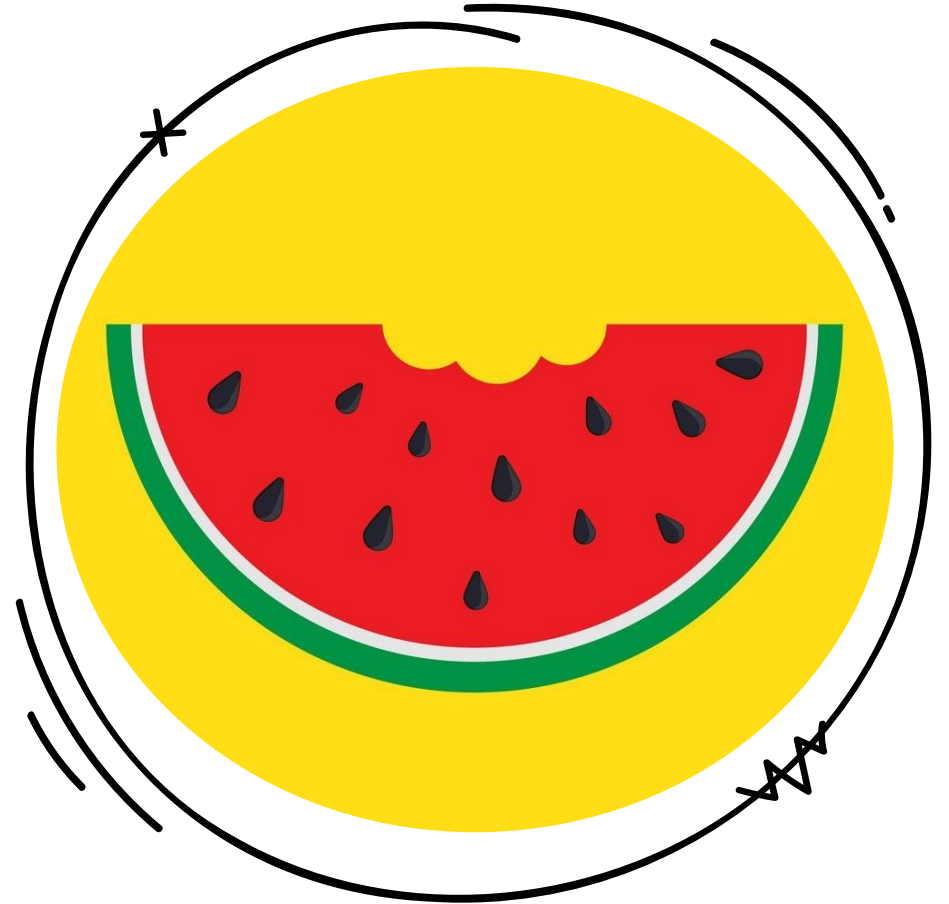


WATERMELON INC.



E. Cogotti | F. Pesciatini | G. Petrelli | M. Gómez



Augmented Reality Engine Application

Flexible development of location-based mobile
augmented reality applications with AREA

E. Cogotti | G. Petrelli

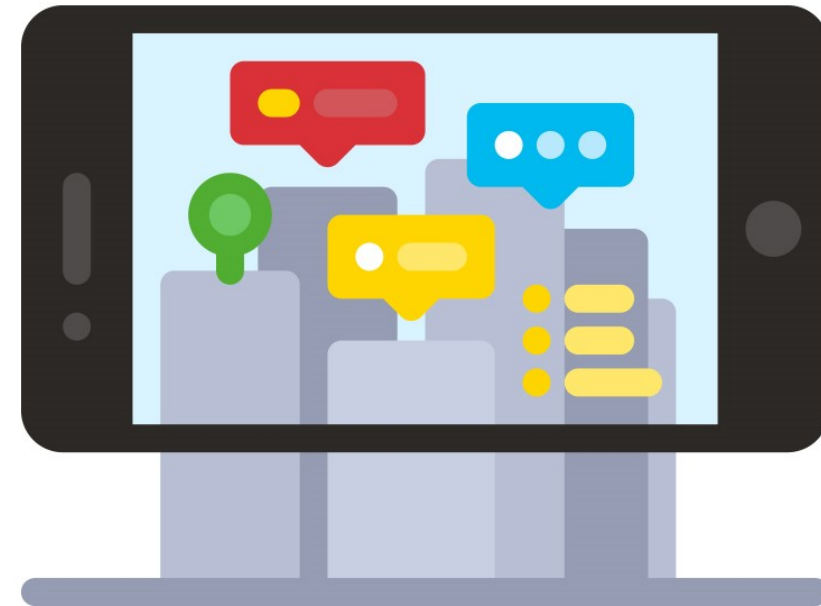
Introduction

Topic : mobile augmented reality

Issues:

- Dynamic OS environment
- Costly and time-consuming development

Goal: flexible framework



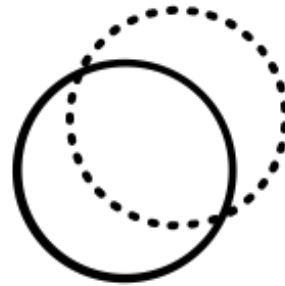
What's AREA and its Goals

AREA is a kernel to implement location-based AR applications

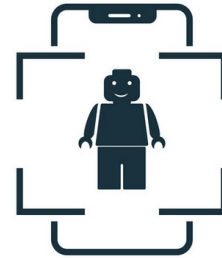
Cross-platform



Abstraction



Games development on top of AREA



Algorithm Pillars

- Objects detected in the camera view
- Pol, tracks, area, 3D objects
- 3D world relate object and user
- Physical camera adjusted w.r.t virtual 3D camera



Standard AREA setting and sensors

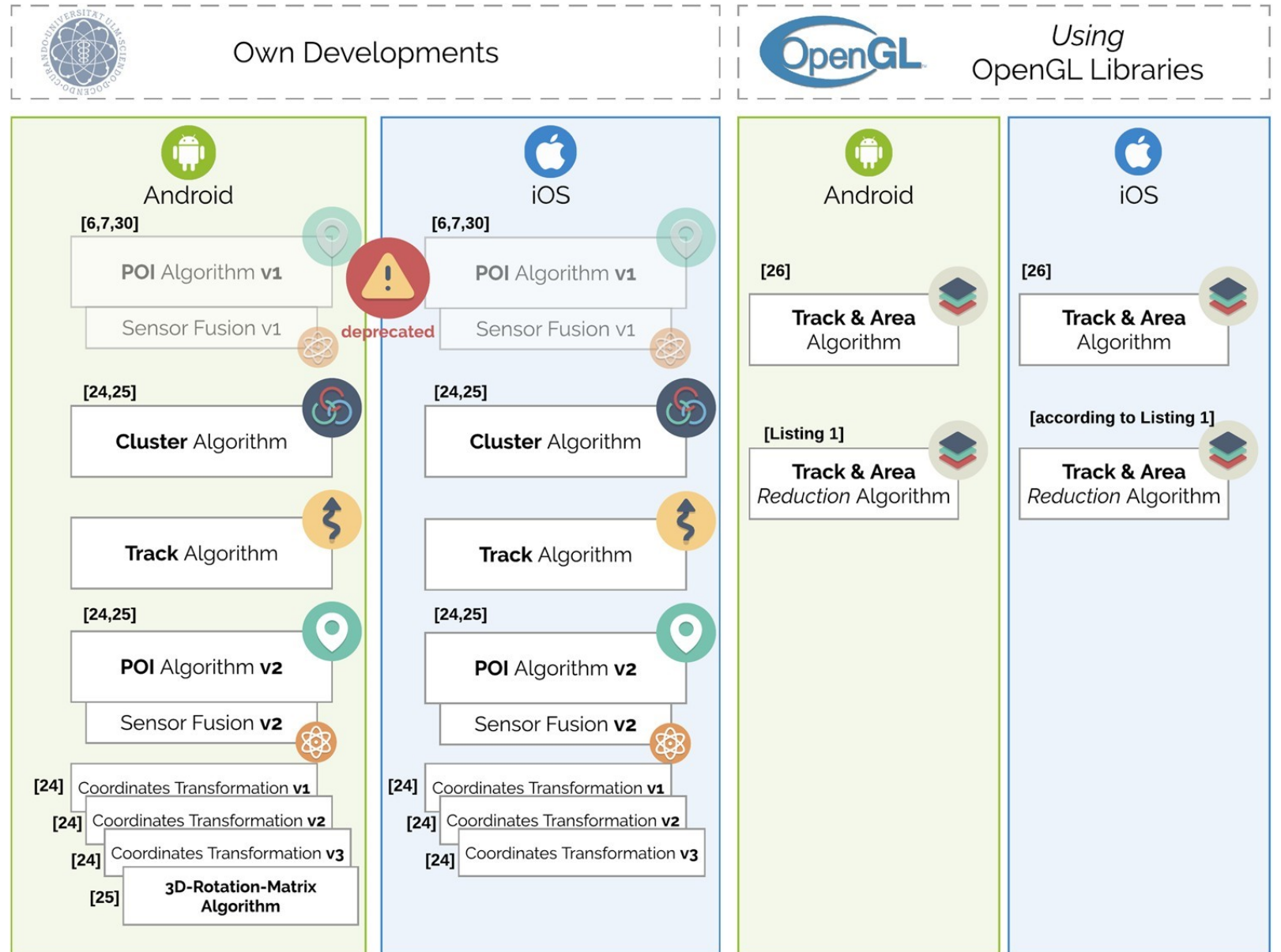
Sensors:

- Camera
- GPS
- Compass
- Gyroscope

```
/// Creates the standard settings of Area
init() {
    radius = 1000
    minRadius = 200.0
    maxRadius = 2000.0
    maxPOIVisible = 100
    cameraFieldOfView = 60.0
    poiClustering = true
    useGoogle = true
    compassBetterOnlyPortrait = true
    radiusPicker = AREARadiusPicker()
    areaIsModal = true
    horizontalClusterWidth = 18.0
    verticalClusterHeight = 8.0

    googleAPIKey = "...
    radius = initializeRadius()
}
```

Algorithm Framework





Optimization of track algorithm

- A virtual track of 1 km needs almost 80 KiB.
- Adaptive levels of detail are needed.
- Track algorithm for performance boost needs specific data structure:
 1. *Checkpoints list*
 2. *degreesY list*
 3. *degreesXZ list*
 4. *Pairs list*

Track optimization in practice

Table 2 Track optimization reduction levels



Level	Reduction schema
0	No reduction
1	Tracks within 50 m of a user's position (i.e., using 50 track points)
2	Tracks within 100 m of a user's position (i.e., using 40 track points)
3	Tracks within 200 m of a user's position (i.e., using 30 track points)
4	Tracks within 300 m of a user's position (i.e., using 20 track points)
5	Tracks within 400 m of a user's position (i.e., using 10 track points)
6	Tracks beyond 400 m of a user's position (i.e., using 5 track points)



ARGame introduction

- Game Goal: find the avatars spread around the city.
- Credits, gained finding avatars, can be used to be redeemed in the participating stores of the city.
- Users learn more about the city.

A screenshot of the ARGame rewards screen. The screen has an orange header bar with the text 'Drillisch LTE' and '12:20'. Below the header, there is a list of rewards. Each reward entry includes a red teddy bear icon, a credit value, the reward name, and a date. The rewards are: 1 Pack Jelly babies (March 5), 3D Ra-Pha 01 (March 11), and 3D Ra-Pha 05 (March 16).

<		-40	1 Pack Jelly babies > March 5
		+10	3D Ra-Pha 01 > March 11
		+150	3D Ra-Pha 05 > March 16

ARGame development and integration

Area20MainActivity initialization

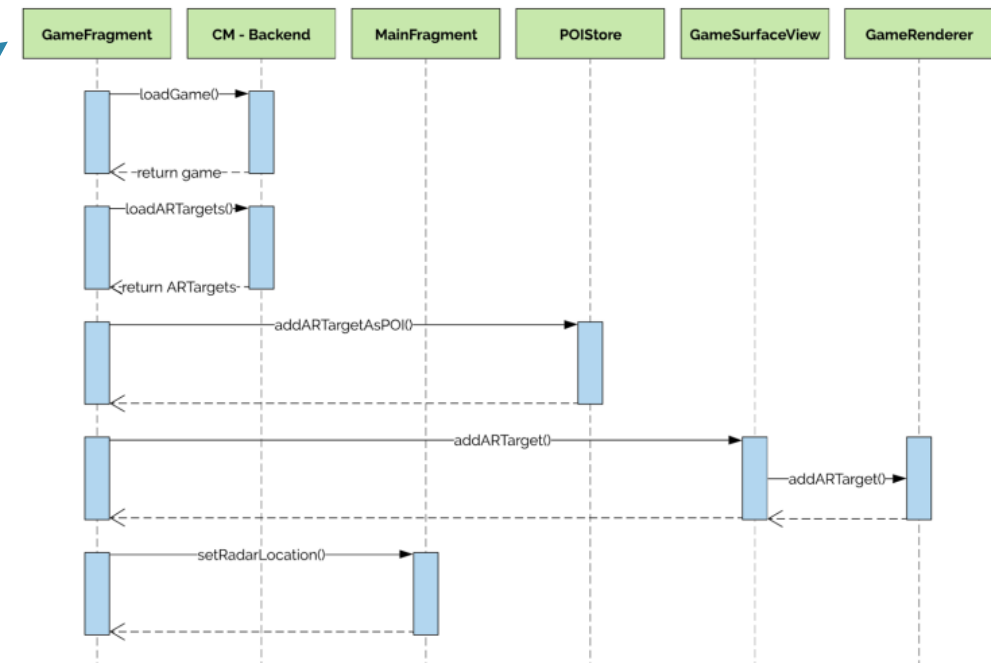
Listing 6 AREA onCreate Method (Android)

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    Toast.makeText(this, "Load Data...", Toast.LENGTH_SHORT).show();

    MapAndArConfig config = getIntent().getExtras().getParcelable("config");
    if (config == null) {
        throw new RuntimeException("AREAMainActivity must be given a proper
            MapAndArConfig object");
    }
    this.config = config;
    callARFragment(getFragment(config));
    Prefs.INSTANCE.openDatabase();
}
```

Call sequence for Area20GameFragment



Class diagram with ARGame extensions on android

