Meeting Agenda

Group: Super duper Omega gruppen
Date: 2020-09-25
Chair: Oskar
Participants: Participants: Samuel, Oskar, Erik, Behroz, Sebastian, Pelle

Objectives (5 min)

Question 1

How should emitters be implemented? Inheritance? Builder?

Question 2

Look at the events. Should we remove angle, and maybe even position? ModelUpdate doesn’t have a position.

Question 3

Look at the Tower class diagram and get feedback.

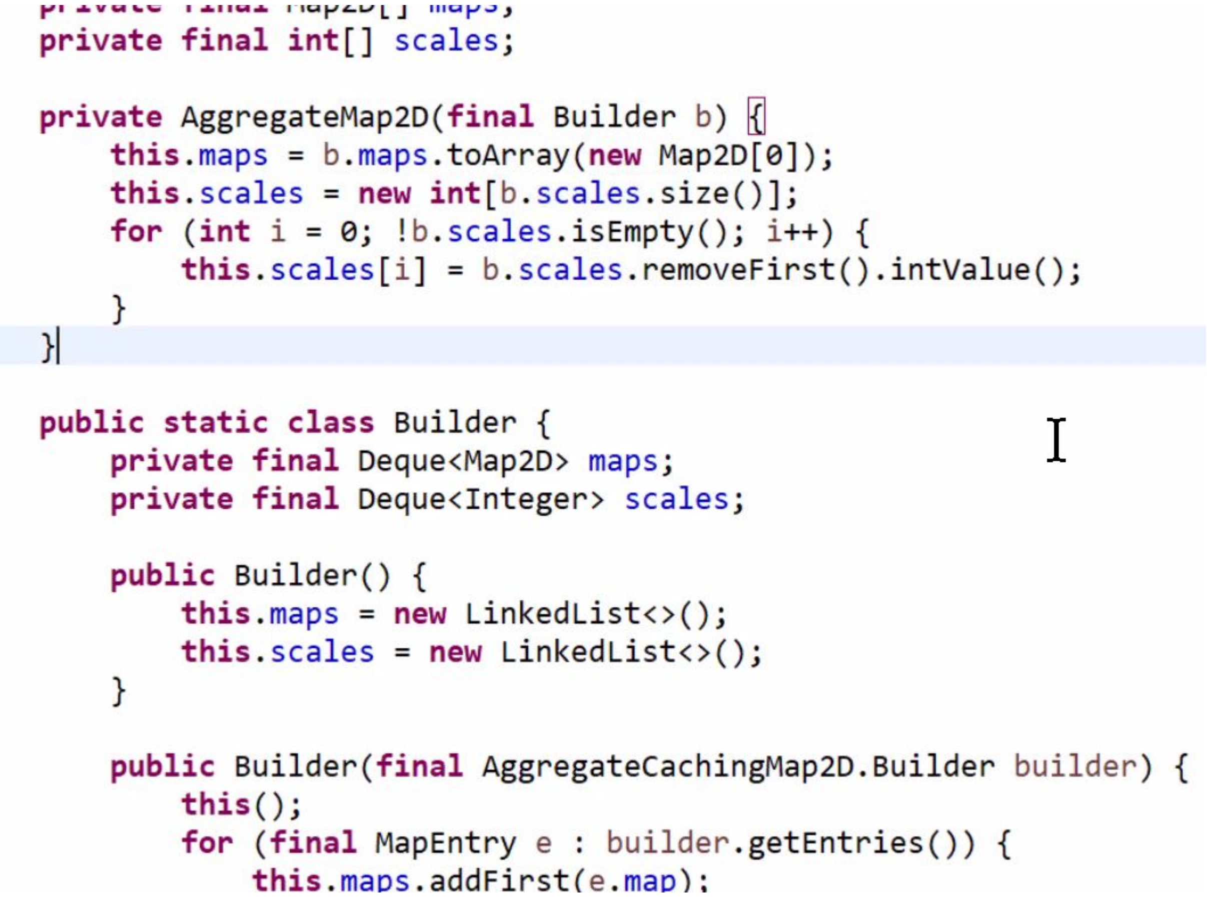
Question 4

Should we remove the Int vector class and just use double?

Discussion items (135 min)

Question 1

Pelle thinks that a builder pattern might be the best design. The builder sets both the information about how the emitter looks as well as the position of the emitter. Then a factory can take in a position and set the correct data as well as the position using the builder.



```
    private final Map2D[] maps;
    private final int[] scales;

    private AggregateMap2D(final Builder b) {
        this.maps = b.maps.toArray(new Map2D[0]);
        this.scales = new int[b.scales.size()];
        for (int i = 0; !b.scales.isEmpty(); i++) {
            this.scales[i] = b.scales.removeFirst().intValue();
        }
    }

    public static class Builder {
        private final Deque<Map2D> maps;
        private final Deque<Integer> scales;

        public Builder() {
            this.maps = new LinkedList<>();
            this.scales = new LinkedList<>();
        }

        public Builder(final AggregateCachingMap2D.Builder builder) {
            this();
            for (final MapEntry e : builder.getEntries()) {
                this.maps.addFirst(e.map);
```

yoink yeeted from Pelle.exe

Question 2

Doesn’t have a great idea on how to fix it. We might be able to use a visitor pattern but that is maybe a bit overkill for this problem. Using instanceof is equally bad. Therefore we decided to keep it since it’s not a huge problem.

Question 3

Could have a common superclass using template method. He thinks that is a good idea.

Should write with aggregation. Using the non filled diamond shape.

We might move EnemyFinder’s range to the tower.

Question 4

Use just double. Have getTruncatedX() to remove the decimal part of it. And use math.floor on it before returning.

Outcomes and assignments (5 min)

We will implement builder pattern and a factory for emitter.

We will implement the new Tower structure but use a template method pattern for fireing method to remove the small code duplication.

We will remove the Vector class and rename VectorD to Vector.

We noticed an issue with fps when we have multiple towers which is because of the rotation of the towers. We will implement a system to save the images.

Wrap up

Next meeting: 2020-09-28 15:00