

Storing Map Information

In some competitions, it may be necessary to store additional information that varies on a per-map basis. An example of this is predefined layers in spleef. The game needs to know where the layers exist in order to handle logic such as only allowing block break for the layers, or allowing the layers to decay. ArenaSpleef makes use of [this functionality](#), and will be used as a reference point at various points in this tutorial.

Storing Information in Your Map

As mentioned earlier in this set of documentation, no game logic should exist in your **CompetitionMap** class. Like **Arena**, it should only contain information and configuration. In the example of spleef, [SpleefMap](#) contains all the stored layers, but the actual logic to decay the layers, or handle block breaking is handled in the relevant [SpleefCompetition](#) or [SpleefArena](#).

In our example, we will store a set of infection points which will infect players when they walk into them:

```
public class MyCompetitionMap extends LiveCompetitionMap {
    public static final MapFactory FACTORY = MapFactory.create(MyCompetitionMap.class,
MyCompetitionMap::new);

    @ArenaOption(name = "infection-points", description = "The points where players can be infected.")
    private List<Bounds> infectionPoints;

    public MyCompetitionMap() {
    }

    public MyCompetitionMap(String name, Arena arena, MapType type, String world, @Nullable Bounds bounds,
@Nullable Spawns spawns) {
        super(name, arena, type, world, bounds, spawns);
    }

    public List<Bounds> getInfectionPoints() {
        return this.infectionPoints == null ? List.of() : List.copyOf(this.infectionPoints);
    }
}
```

```

public void addInfectionPoint(Bounds bounds) {
    this.infectionPoints.add(bounds);
}

public void removeInfectionPoint(Bounds bounds) {
    this.infectionPoints.remove(bounds);
}

@Override
public LiveCompetition<?> createCompetition(Arena arena) {
    if (!(arena instanceof MyArena myArena)) {
        throw new IllegalArgumentException("Arena must be an instance of MyArena!");
    }

    return new MyCompetition(myArena, arena.getType(), this);
}
}

```

With the code above, BattleArena is now capable of loading a list of **Bounds** from the maps YML file. **Bounds** is a class provided by BattleArena, which simply let you store a minimum and maximum position. It will be used in our case, as we will be able to store a list of bounds throughout various points in the map to infect players.

Now that we have the necessary code in order for this to work, we will want to create a way for a user to add bounds to the map. For this, we will be creating a custom command executor.

Revision #1

Created 14 December 2024 14:53:56 by Redned

Updated 14 December 2024 15:03:42 by Redned