Group Name:	Group 5: Mac & Cheese
Date: 30.05.2025	

Agenda #5: Weekly Meeting Date

1. **Division of Labor for This Week:**

Assignment 1	Brainstorm	Draft	Elaboration	Revision	Final
A. JUnit tests for tower placement-related methods and tower classes	Team	Ege Yiğit Erbil	Ege Yiğit Erbil	Ege Yiğit Erbil	Team
B. JUnit tests for map editing-related methods	Team	Valentin Py	Valentin Py	Valentin Py	Team
C. JUnit tests for gold bag collection-related methods	Team	Selin Ünal	Selin Ünal	Selin Ünal	Team
D. JUnit tests for enemy wave generation-related methods	Team	Ece Göncü	Ece Göncü	Ece Göncü	Team
E. JUnit tests for tower upgrade-related methods	Team	Ece Arslan	Ece Arslan	Ece Arslan	Team

Plan for Next Week:
We are planning to work on polishing the implementation code and improving the UI components and visuals, as we will be preparing the project for final review and demonstration.

2. **Division of Labor for Next Week:**

Assignment 2	Brainstorm	Draft	Elaboration	Revision	Final
A. Finalizing Map Editting	Team	Valentin Py	TBD	TBD	Team
B. Finalizing Tower-Related Implementations	Team	Ege Yiğit Erbil Ece Arslan	TBD	TBD	Team

C. Finalizing Enemy-Related Implementations	Team	Ece Göncü Selin Ünal	TBD	TBD	Team
D. Finalizing UI	Team	Team	TBD	TBD	Team

3. Assessment of Success:

We accomplished what we had planned for the previous week. This week, we planned to write proper specifications and complete the test implementations for selected methods and class. Specifically, each team member individually identified a non-trivial method and wrote appropriate Javadoc-style specifications (including requirements, modifications, and effects), in the source files and implemented a minimum of three JUnit tests. We wrote an overview, abstract function, representation invariant, repOk method, and a minimum of four JUnit tests to check the class. As indicated, each member created a separate git branch for their specifications and tests and merged their contributions into the main branch. A few topics missing/needing clarification might be whether some helper methods should be tested separately, and the parts left to update are implementing more robust corner-case tests for the chosen methods and class.

4. Open Issues & Questions:

- ➢ Can we use parameterized tests in JUnit for repetitive test cases with minor variations, or should each test be written explicitly?
- ➢ If a method is already covered indirectly via testing of another class, do we still need to write explicit unit tests for it?
- ➢ For the repOk() method in the chosen class, should we assert failure with exceptions or just return false when invariants are violated?

etc…