# Sprint Plan #3

**Game:** Fishy
**Group:** 11

| User Story | Task | Task Assigned To | Estimated Effort per Task (in hours) | Priority (A-E) |
|---|---|---|---|---|
| **Exercise 1: 20-Time, Reloaded** | | | | |
| | 1.1: Making requirements for new extension/improvement | Millen | 0.5 | A |
| | 1.2: Sending requirements to TA | Millen | 0.1 | A |
| | 1.3: Powerup Factory | Millen | 2.5 | A |
| | 1.4: Powerup 1: Adding shield powerup from turtle. | Martijn | 3 | B |
| | 1.5: Powerup 2: 1 up | Millen | 3 | B |
| | 1.6: Powerdown 1: Poison fish which inverts the controls | Martijn | 3 | B |
| | 1.7: Powerdown 2: Adding powerdown when fish speed gets a lot higher | Millen | 3 | B |
| | | | | |
| **Exercise 2: Design Patterns** | | | | |
| | **Pattern 1: State pattern** | | | |
| | 2.1: Write a natural language description of why and how the pattern is implemented in your code. | Ricardo | 0.5 | A |
| | 2.2: Make a class diagram of how the pattern is structured statically in your code | Ricardo | 0.5 | B |
| | 2.3: Make a sequence diagram of how the pattern works dynamically in your code | Ricardo | 0.5 | B |
| | | | | |
| | **Pattern 2: Singleton pattern** | | | |
| | 2.4: Write a natural language description of why and how the pattern is implemented in your code. | Leon | 0.5 | A |
| | 2.5: Make a class diagram of how the pattern is structured statically in your code | Leon | 0.5 | B |
| | 2.6: Make a sequence diagram of how the pattern | Leon | 0.5 | B |

| | | | | |
|---|---|---|---|---|
| | works dynamically in your code | | | |
| | | | | |
| **Exercise 3: Software Engineering Economics** | | | | |
| | 3.1: Explain how good and bad practice are recognized | Danique | 0.5 | B |
| | 3.2: Explain why Visual Basic being good in the good practice group is a not so interesting finding of the study | Danique | 0.5 | B |
| | 3.3: Enumerate 3 other factors that could have been studied in the paper and why you think they would belong to good/bad practice | Danique | 0.5 | B |
| | 3.4: Describe in detail 3 bad practice factors and why they belong to the bad practice group | Danique | 0.5 | B |
| | | | | |
| **Own assignments:** | | | | |
| | 4.1: Refactoring code | Leon | 5 | C |
| | 4.2: Testing | | | |
| | 4.2.a: Writing tests for new features | Martijn, Millen | 3 | D |
| | 4.2.b: Writing tests for models | Danique | 3 | D |
| | 4.2.c: Writing tests for opponents | Ricardo | 3 | D |
| | 4.2.d: Writing tests for position | Leon | 3 | D |
| | | | | |
| | | | | |