P2 Übungen, Iterative Development

Abbildung 1: A ludo board

ACME Inc. is doing consulting now. The industry has learnt that ACME is the place to go to for ASCII graphics implementation of board games. Today, an implementation of Doris & Frank's boardgame Ursuppe is called for.

Since it's a large project, the clients insist that you phase out development. The first stage will only implement the board and its rendering. The second stage will add the basic game logic, without supporting any of the provided genes, or interactivity. The third and final stage will add genes to the game.

Rules

The game rules.

Simplifications:

- You may assume exactly 3 players.
- You need only implement 4 genes.

First stage

For stage 1, ACME *only* committed to their clients to initialize the Ursuppe board for 3 players. Verify that the board is initialized according to the game rules, and that it is displayed correctly (i.e., the correct number of

amebas are on the field: 2 per player, you can place the amebas at random in this stage, all fields have all 6 foodstuff cubes on them, 2 of each color.)

- Implement the `Board` and `Square` classes.
- Write Unit tests and make sure that all tests are green.
- The board should be able to render itself as ASCII graphics.

When you're done, tag your solution as `ursuppe01` with the `-a` option on (`git tag -a ludo01`). Be sure to push with the `--tags` option so we can see your tag. See man git-push.

Second stage

After the successful stage 1, your contract was extended. (Luckily, you're an experienced board game developer by now.) For stage 2, ACME promised the client to deliver a simplified variant of the game. The game shall be played automatically (as opposed to interactively) by three players. The players may not buy genes, and amebas never move, but always drift (the terms are explained in the game rules.).

- Complete the `Player` class.
- Write up a `Game` class that takes the game thru all 6 phases of a round. In phase 1, amebas always drift.
- Write JUnit tests that indicate that your game works properly according to the game rules. Achieve test coverage above 50 %.
- Write a `shouldRunRandomGame` smoke test method that makes three players play (automatically and not interactively). The output shows whose turn it is and the board after every step.

When you're done, tag your solution as `ursuppe02`. Be sure to push the tag using the `--tags` option as in stage 1.

Third stage

In this stage:

- Implement moving of amebas, and implement any 4 basic genes (no advanced genes).

When you're done, tag your solution as `ludo03`. Be sure to push the tag using the `--tags` option as in stage 1.

Deadline

The first stage is due next week on Wednesday, March 30, as usual. The other 2 stages, however, are both due the same day, on May 4, 2011.