ETH Game Programming Lab 2010

Odysseus' Quest

Race to Ithaca

Remo Meyer, Thomas Plüss, Andreas Burch

Prototype Chapter

22.03.2010

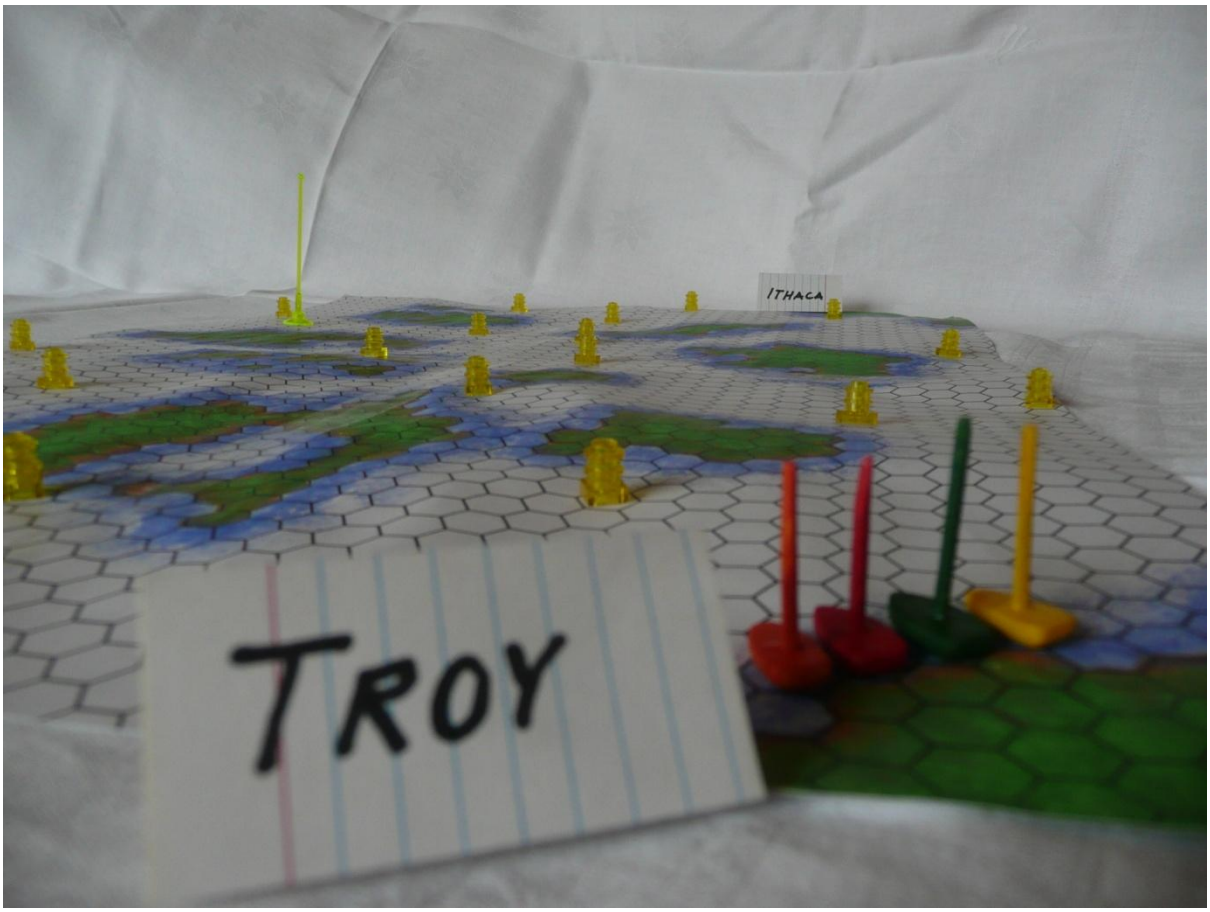
TABLE OF CONTENTS

1 GAME PROTOTYPE

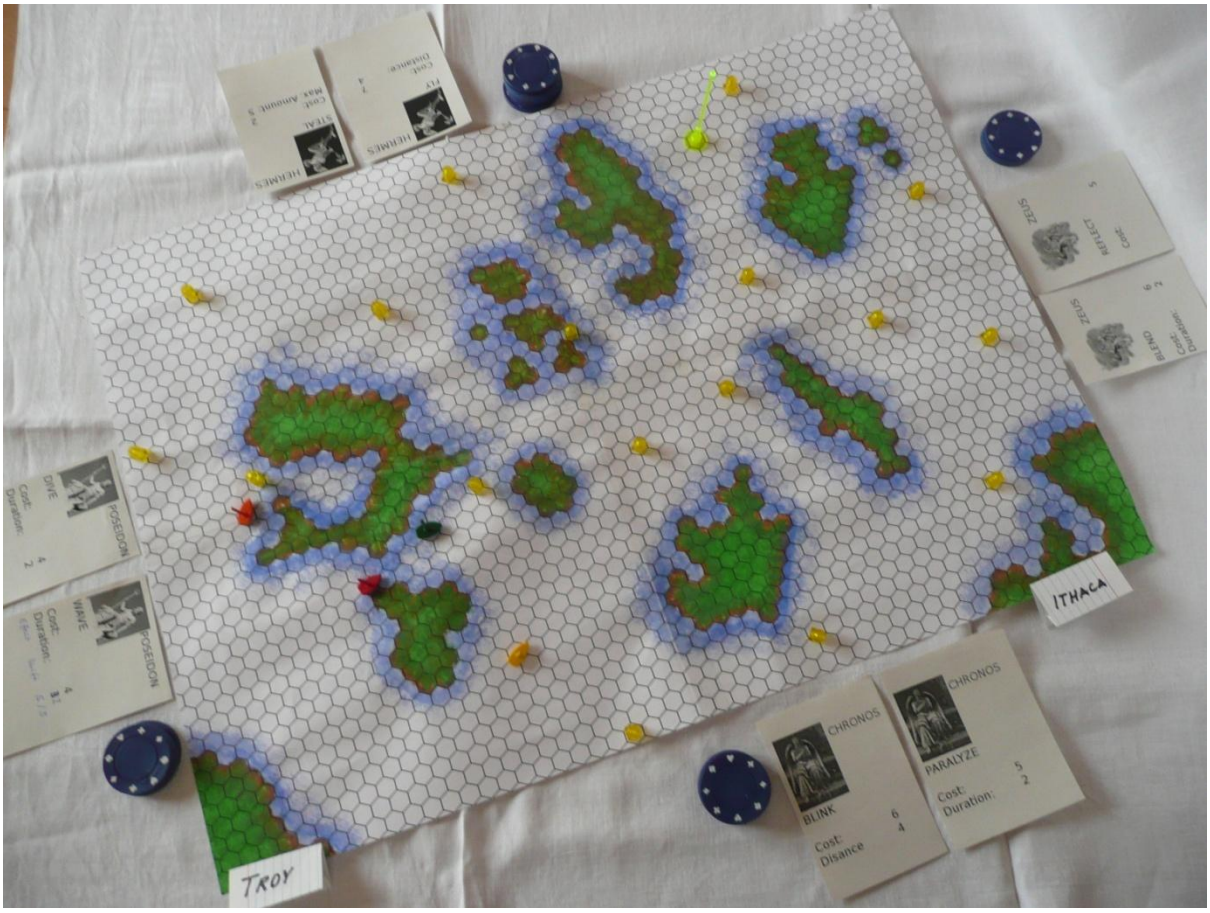
1.1 SETUP PHASE

Our prototype is a turn based board game. It contains the core elements of our game idea. Like the Xbox game to be built, it can be played by 2-4 players. The board consists of hexagonal fields that represent either water or islands. Ambrosia has been (pseudo-)randomly distributed to the water fields.

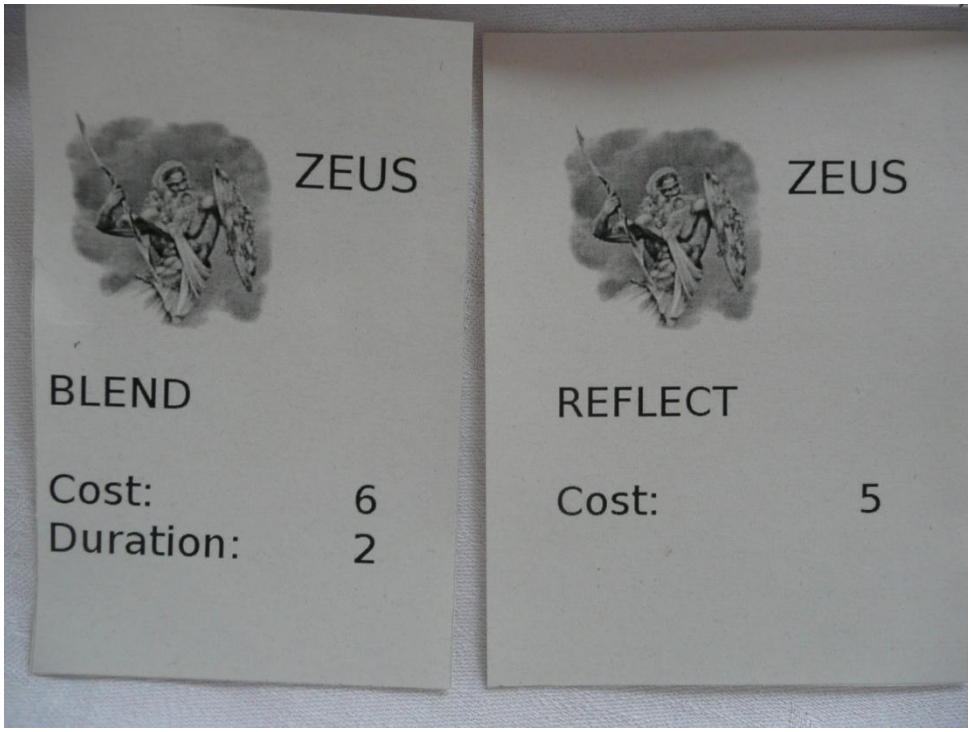
The following picture shows the board in the initial state. The players all start at Troy. The little yellow blocks indicate ambrosia and the yellow high pole (in the back to the left) depicts the first checkpoint all players have to head for.



The following image shows an overview picture of the game some time after the beginning. The poker chips represent the collected amount of ambrosia.



In the setup phase, the players can choose a god and are given the god cards, on which the abilities (with cost and effect) are described (see next picture for an example).



1.2 TURN PHASE

During one turn, a player has 4 moving actions (move straight or turning). If a player has to turn twice in a row on a single hexagon (e.g. because he is stuck in a dead end), he has to make an "empty" action in between. Furthermore the player can make use of his god's abilities by sacrificing his collected ambrosia. For certain attacks (like "wave" or "paralyze"), he can choose a target for the auto-aimed bullet.
Instead of using a specialized ability of the Olympic gods, the player can also boost himself forward by paying ambrosia and therefore gaining two additional actions/moves.

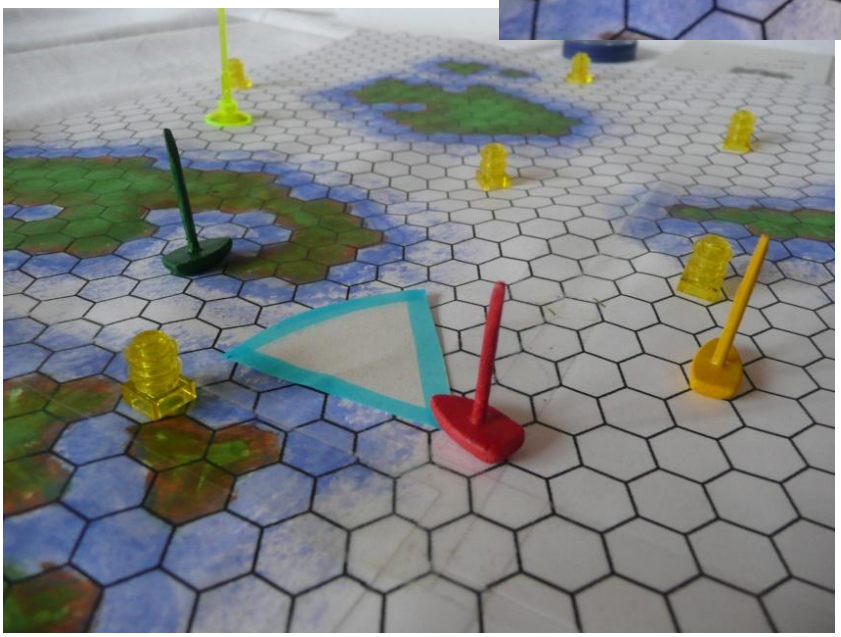
1.3 ABILITIES

We used the following list of gods and abilities when playtesting the prototype.

- Zeus
  - Blend
    - Cost: 6
    - Effect: All other players have to draw blind for the next two turns
  - Reflect
    - Cost: 5
    - Effect: An attack ('Wave', 'Paralyze', 'Steal') of an opponent is reflected to the opponent
- Hermes
  - Steal
    - Cost: 2
    - Effect: Steals up to 5 ambrosia from targeted opponent
  - Fly
    - Cost: 7
    - Effect: The player is able to fly over obstacles for one turn
- Chronos
  - Paralyze
    - Cost: 5
    - Effect: The targeted opponent cannot navigate for the next two turns, i.e. moves straight in the direction he was moving before
  - Blink
    - Cost: 6
    - Effect: The player teleports 4 fields forward in the direction he was moving before
- Poseidon
  - Wave
    - Cost: 4
    - Effect: The player sends out a wave in its moving direction. The wave moves over two turns (4 fields in first turn and 8 fields in second turn) shifting away opponents hit by the wave front in the wave direction by 5 fields in the first turn and 3 fields in the second term
  - Dive
    - Cost: 4
    - Effect: The player transforms to a water form and is linked to the sea what grants him invulnerability for two turns.

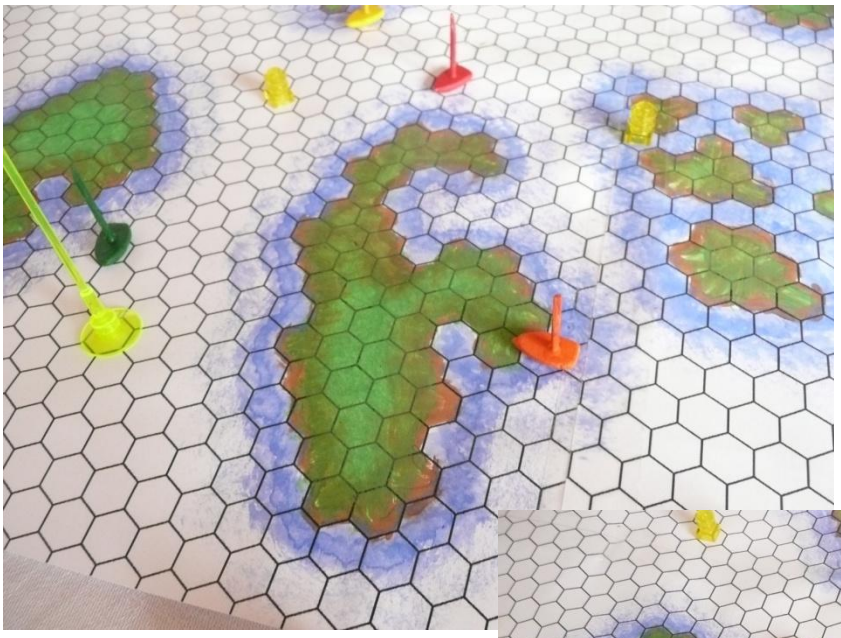
1.3.1 EXAMPLE “WAVE”

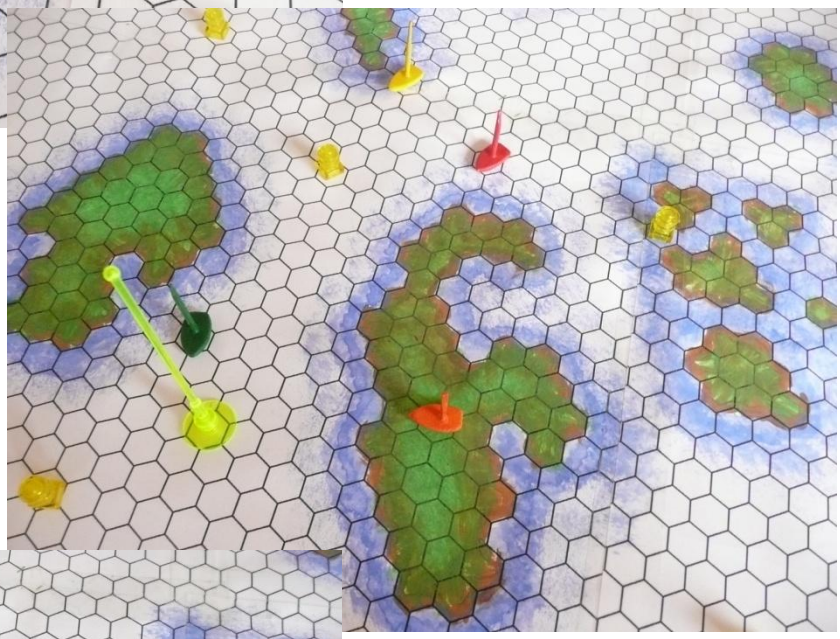
The following sequence of three images shows the usage of the ability “wave”. The effect is limited to some sector in the direction of movement of the attacker. The attacked player gets flushed away.



1.3.2 EXAMPLE “FLY”

“Fly” is another ability, one could make use of. In the following sequence of three images you can see how the orange player can sail through/over the island, while the others have to go around it.





1.4 RULES

The following list of rules should help making the game play clear:

- The players do their moves in a fix order
- In a player's turn, he can use the four actions for moving:
    - Moving straight costs one action.
    - Turning by 60 degrees costs one action.
    - Turning by 120 degrees costs three actions.
    - Only “water”-hexagons may be sailed on (except for 'fly', the player can fly over obstacles like islands and opponents).
    - Crashing into an opponent costs two actions (one for moving, one for crashing). The opponent gets shifted by one field in the direction the crashing player moves.
- Each player has to pass all the checkpoints in the given order
- 'Reflect' can be used spontaneously, meaning it doesn’t have to be the turn of the player using 'reflect'.

1.5 WINNING CONDITION

- The player which has passed all checkpoints and reaches Ithaca first wins the game

2 WHAT WE HAVE LEARNED FROM THE PROTOTYPE

- The checkpoints have shown to be the main source of player interaction, and thus fun. So to keep the game entertaining, it is necessary to have many (>10) checkpoints. The distribution of the checkpoints should also make sure that there is interaction even if one player is ahead and directs to another checkpoint than his opponents.
- Ambrosia has to be distributed quite cleverly; otherwise the leader would be privileged too much.
- Abilities with an "area effect" (against all opponents in some distance around the player) didn't work that well. That's why we changed some abilities, such that kind of a missile is fired and auto-aimed to the closest opponent.
- The costs and effects of the abilities that we used in the prototype turned out the be quite balanced.
- The game would also work well with a distant camera for all players instead of the split screen. We are not sure yet, which option will be the better one. Depending on several factors (game play, fun, performance, ...), we will decide on this matter in the next weeks.

3 PROGRESS IN DEVELOPMENT SCHEDULE

The software engineering part (including general setup of the project, defining which thread is responsible for which part and the general hierarchy of the classes) can be considered as finished. In this part we're even a bit further than we hoped to be, as it has proven to be easier to include the physics library, and the basic menu structure straight ahead than to do this afterwards (it was actually planned as low target and will then be finished). So we're pleased to already have a collision system and a basic menu structure (which will be filled with more functionality in the next weeks).

A basic boat model is in the game and ready to cruise.

The camera is set up and follows the player through the scene.

The steering part has been proven to be more crucial and difficult than we actually thought, and we will test some different approaches and decide not later than in the end of this week how to go on.

We've implemented an image to height map converter, to be able to create the islands in a efficient way. So we are now half way through the process of creating the scene, the islands are there and will be flooded with some basic water implementation (not yet with waves) in the next days.

The basic checkpoint logic and collision checking with the boats is implemented, there are some minor tweaks that has to be done, but the core is implemented and because this is scheduled for next week, we've enough time left.

Overall it works quite well and we're a bit ahead of the schedule. We'll use this time to work out how to implement steering best, and go on with the rest of the schedule as planned.