ETH Game Programming Lab 2010

Odysseus’ Quest

Race to Ithaca

Remo Meyer, Thomas Plüss, Andreas Burch

Interim Report Chapter

19.04.2010

TABLE OF CONTENTS

1 FIRST 5 WEEKS OF IMPLEMENTATION - A DIARY

1.1 WEEK 1

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 have to be done, but the core is implemented and because this is scheduled for next week, we've enough time left.

1.2 WEEK 2

The tuning of the rigid body collisions (sailing boats, islands) proofed to be quite challenging, but we approach slowly to a stable solution. Some problems have still to be solved (e.g. slightly displaced physics objects).

For the steering we tried an approach of, accelerating with forces and damping with damping forces, which looks quite good. However there's a lot of balancing (parameter tuning) left.

The spilt screen (4 player mode) isn’t implemented yet.

There's a problem with the update of the camera, respectively the synchronization of the threads/cores, which introduced some annoying flickering.

1.3 WEEK 3

The location of ambrosia is read from a bitmap (similar to the height map). The players can collect ambrosia by sailing into the model.

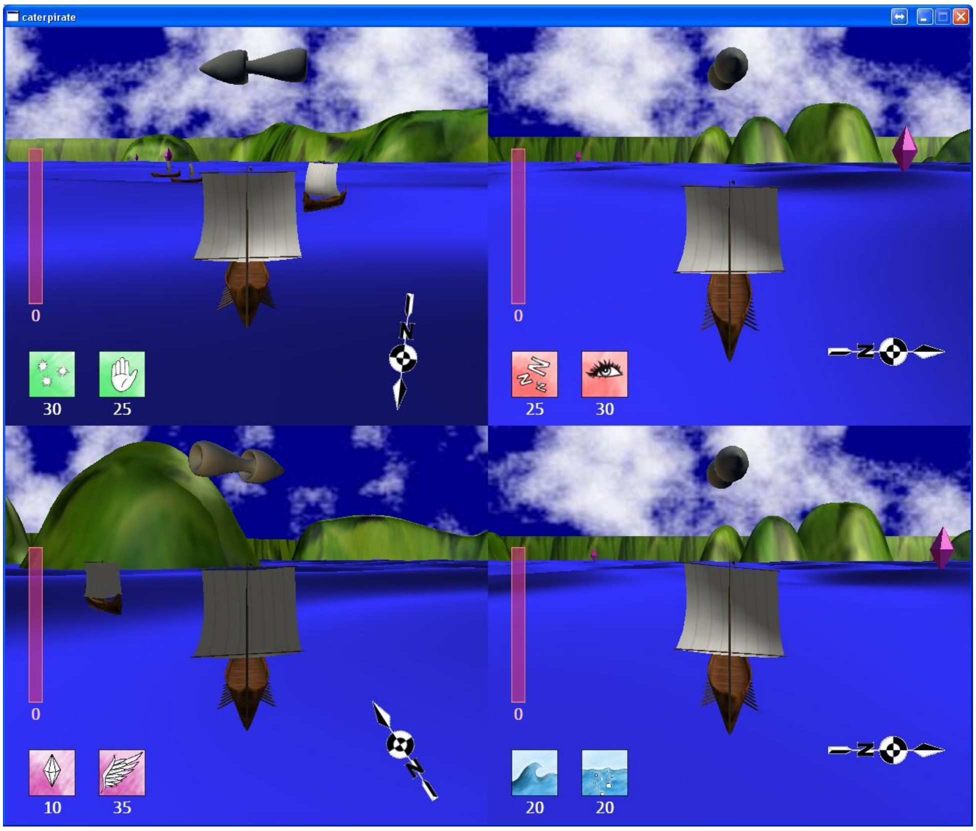
There’s a first version of the HUD: An arrow above the player directs him to the next check point. Furthermore the ambrosia amount is shown on a bar.

The abilities are missing up to now.

We did a first approach towards water simulation. Because of performance reasons we skipped our initial plan of make a shallow water simulation on the CPU and start to make all the computations on the GPU.

It took quite a while to find out how a texture lookup in the vertex shader can be done (it's actually quite easy, just use shader model 3.0 or higher). We now can move the vertices of surface up and down, but as there's no shading yet, there can’t be seen a lot.

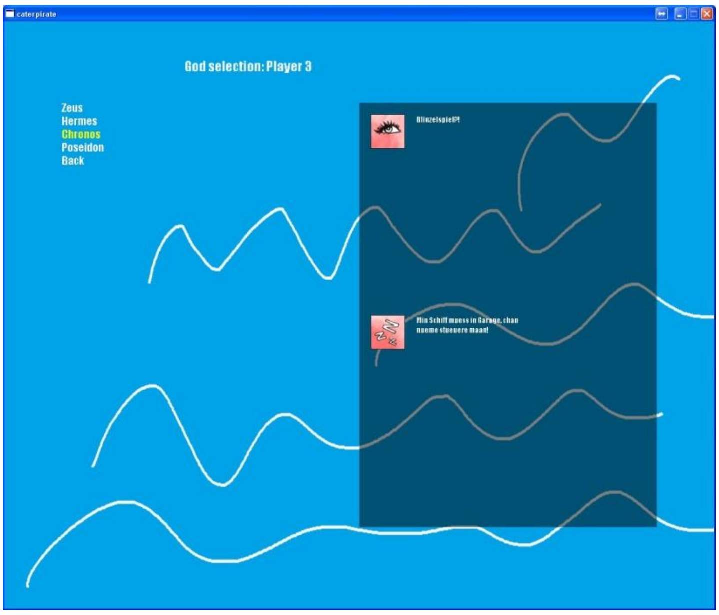
Split screen is implemented. Up to 4 Players are supported now.



1.4 WEEK 4

A more detailed, textured model of the ship is in the game (with a sail and rudders)

The menu has now all needed functionality (ship and god selection for all players)

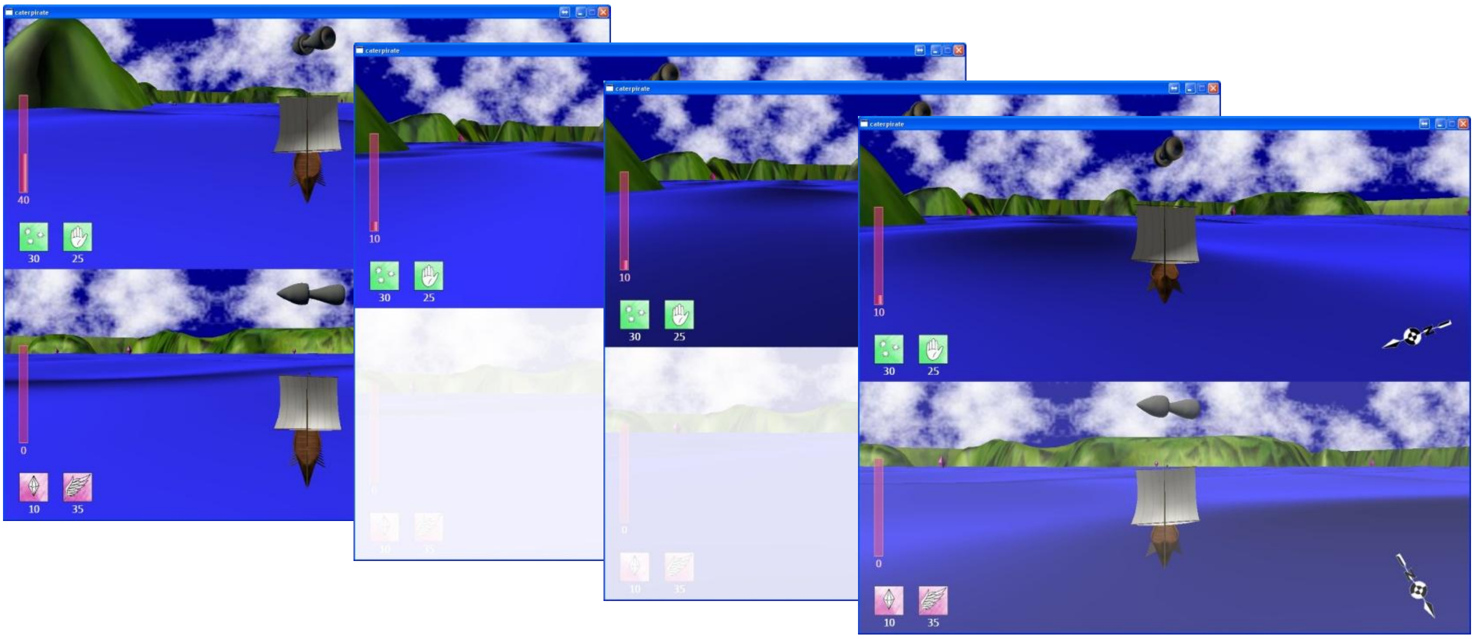


God selection: Player 3

Blinzelspiel?!

Min Schiff muess in Garage, chan nueme stueuere maan!

Ability “blend”: For some seconds the opponents screen becomes white and then slowly fades back to normal.

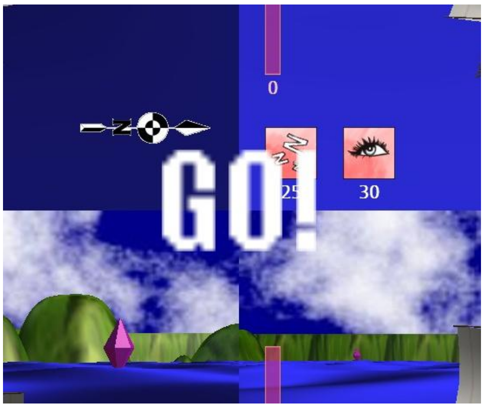


The shading of the Water is implemented. To avoid too much geometry in the scene, we only draw the visible part of the water for each player. We also tried to implement shallow water on the GPU. It could be done, but there are several problems, including that the lookup tend to smear the waves out, and thus energy has to be reintroduced which would lead to ugly artifacts.

Animated wing model for the fly ability.

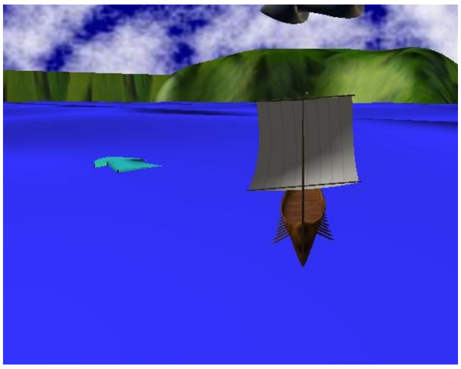
1.5 WEEK 5

There’s a countdown from 3 to 0 before the race starts.



Ability “blink”: Short range teleport for the player’s ship. It’s also possible to go through a hill. If the ship would end up inside the hill, the teleport is reverted (but the ambrosia keeps withdrawn).
Ability “paralyze”: Your opponents can’t change the steering for some seconds.
Ability “steal”: You’ll get some ambrosia from your opponents.
Ability “dive”: For some seconds your opponents can’t see you (there’s also no collision) and you are immune to attacks (blend, paralyze, and steal)
Ability “veto”: If there’s an ability used on you (e.g. paralyze) you can stop this effect and in turn reflect it to the player that started the attack.

There’s a global wind direction that shows up in the direction the water moves. Moreover the players are faster if they sail in this direction (this effect is too weak at the moment).



Power up items in the scene can be used to change the global wind direction.

We came up with a better and more stable water “simulation” solution. We take the current wind direction as direction of how to advect the wave. The “Wave” is read out of different textures which are weighted by the Wind direction and a term which determines how near the point is to an island.

The ships are now floating on top of the waves.

A skybox is drawn to give the scene a background.

We extended the HUD and now show the abilities you can use and their cost. Also the global wind direction in respect to your orientation is shown.

2 SUMMARY AND OUTLOOK

2.1 MAJOR CHANGES (IN CONTRAST TO THE FINAL GAME PROPOSAL)

We introduced a steering model which is more sophisticated than the one we planned in the beginning. There's now a base velocity (rowing) and one which is based on a global wind direction. The player will be able to strike and hoise the sail when it's appropriate (this is not yet implemented). The wind influences also the movement of the waves; this makes the scene less static and gives the player a feeling of the wind. To help the player there's also an arrow which indicates whether the player is moving with or against the wind. The global wind is for all players the same and can be changed by the wind-direction-power-up (green arrow).

2.2 OVERALL PROGRESS (LAYERS)

We're pretty happy with the result we're having at the moment. As to be expected, the schedule could not always be satisfied. But overall we achieved to be at the stage we aimed to for this interim report.

Some things are already implemented although they are actually scheduled for the next weeks. Like the coupling of boat and water which just was needed to give a better feeling of being on the ocean, or some animations which we have already in Maya and just wait to be ported into our game.

Other things proved to be quite challenging and time consuming, like the steering of the boats (which we had to redo and improve several times, and is still not finished).

However in the sum we progress as we hoped to and this milestone give us the needed information to see where still more work is needed and what already works quite well.

2.3 CHANGES TO SCHEDULE

We leave the schedule as it is, because it generally proved to be quite accurate and a good leading tool.

However, there will be some changes to the weights. E.g. we'll focus more on the environment (and with it the map), because the game feels now a bit slow (see 2.5) and we think this is the right way to enforce that the player interact more often and thus make the game even more exciting and fun.

2.4 KNOW BUGS

Sometimes the physics overreact.

The collisions are sometimes displaced.

2.5 RESULTS FROM FIRST PLAYTESTING

The player has at the moment to little decisions and actions to make.

The perception of the velocity is not really given.

Interaction has to be enforced more.