Interim Report

Development schedule - status

Functional minimum

- singleplayer
- multiple identical sheep
- sheep can jump over fence *(fence to be modeled)*
- sheep walk away from player (no freewill movement)
- simple world, flat plane, simple fence *(fence to be modeled)*
- static camera
- no character animation

Low target

- competitive multiplayer
- refined sheep movement
- split screen

Desirable target

- character animation *(in progress)*
- different levels
- improved graphics *(we have a nice sheep, but missing most of the other models)*
- achievements/points for players *(simple sheep counting)*
- special sheep
- power-ups
- sound effects / theme song

Sheep movement

After the first simple “walk away from player” implementation, we added some flocking behaviour. Our work is based on Reynolds’ 1987 Paper “Flocks, Herds, and Schools: A Distributed Behavioral Model”. Since the results from his approach resemble more the flocking of birds than sheep, we added some modification. The main problem was that birds don’t stop. Nevertheless, the results looked promising and seemed to add value to the game.

At first, we simply added a friction force, but the results didn’t look natural. We discovered that modifying the center of measuring the separation force made the flock behave amazingly similar to actual sheep. We moved the center of the sheep that’s currently evaluated slightly backwards, which resulted in the sheep, in addition to repulsing others, pushing itself back.

We have a very good working sheep flocking implementation in Processing. The different frame rates and scale however made it hard to port the algorithm to the XBox game and we are still struggling with finding the right parameters. We also want to introduce sheep states (graze, wander) like proposed in Buckley 2007, “Modelling Sheep Flock Interaction with Sheepdogs”.

Animation

We are currently rigging our models. Animation will be one of the next steps to implement. Apart from that, we came up with a nice little algorithm to decide where the sheep should be heading:

Turning the sheep to whatever direction the velocity happens to point to resulted in some really nervous “jumping”, especially when the sheep were almost standing still. To work around that issue, we implemented a virtual tail, modelled as a strut attached at the sheeps center with a bearing that allows free rotation around the up-axis. The tail will be dragged by the sheep. The sheep will always head away from its tail.

For now, the sheep are static. When they enter the hot zone in front of the not yet existing fence, they disappear and reappear somewhere on the playing field.

Modeling, level design

Modeling turned out to be more challenging than we originally assumed. It is therefore the part we are behind our schedule the most. To make the design of new levels easy, we intend to generate them out of bitmaps we can create in a graphics editor. But since we allocated most of our resources to modeling for now, we haven’t started implementing different levels yet.

Our current games features a simple rectangular field with two hot zones. It turned out to resemble a soccer field. Shooting goals with many balls that look like sheep and behave like flocks turned out to be great fun.

Camera, HUD

Our game currently has two camera modes: legacy with one camera showing the whole playing field and following where the camera shows only part of the field but follows the player. If there are two players present, the following cam uses a split screen.

The legacy mode is still very useful to test and debug the sheep behavior, but will be impractical on larger maps. The following cam shows more details from the models and adds an element of discovery. We however have yet to find the right camera distance and angles, especially for the split screen mode.

For a shorter round trip time while developing, we didn't implement a start menu yet. The game starts right away, and also never ends. A score is shown at the top left corner of the screen which for now just counts the sheep having entered the players hot zone. Displaying the score led to some rendering issues. But we hope to fix them for our presentation.

Collision detection

We used the simple physical model provided by the Reynold flocking implementation to do a simple collision detection between sheep. Basically, we defined an additional repulsing force for every sheep with a very small radius but a large magnitude. This does not avoid all the collisions, but since sheep are fluffy fellows, some of them being squished together doesn't seem unnatural.

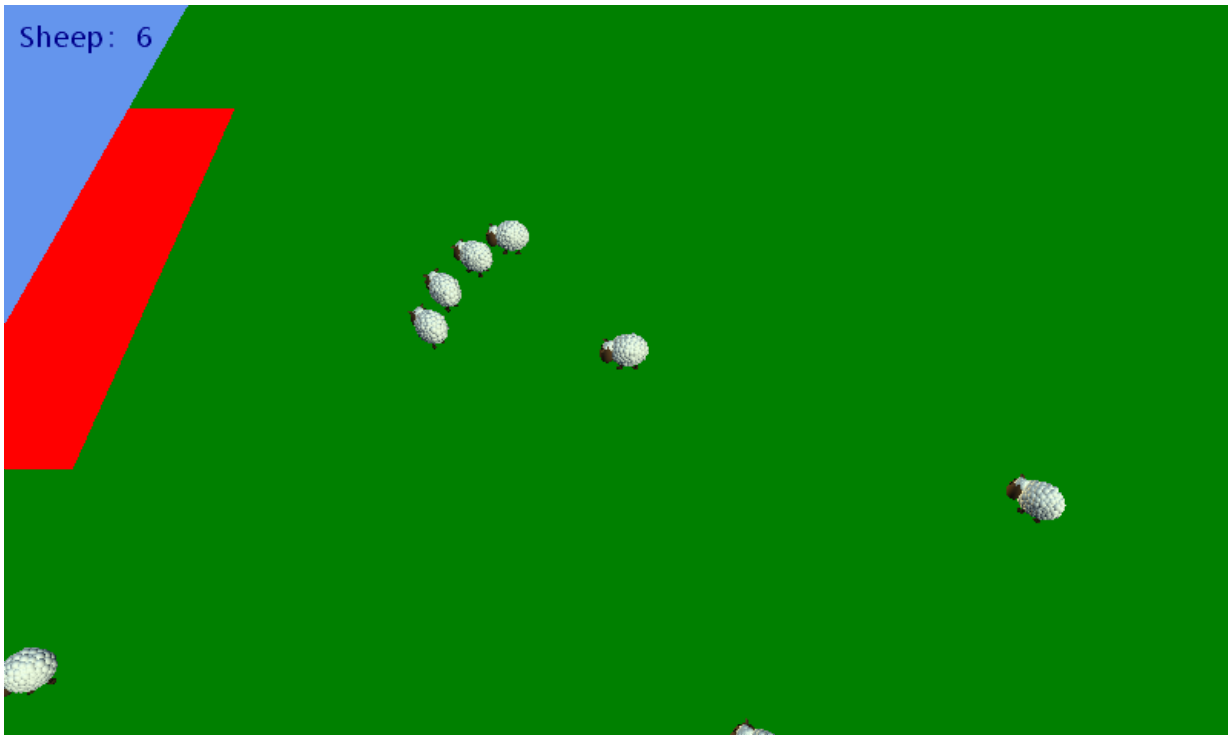
For the walls and other objects we designed a hard limit based on position and a repulsive force orthogonal to it (so the sheep stop walking towards it). The implementation is currently stalled since we don't have our final level framework yet.

Overall progress

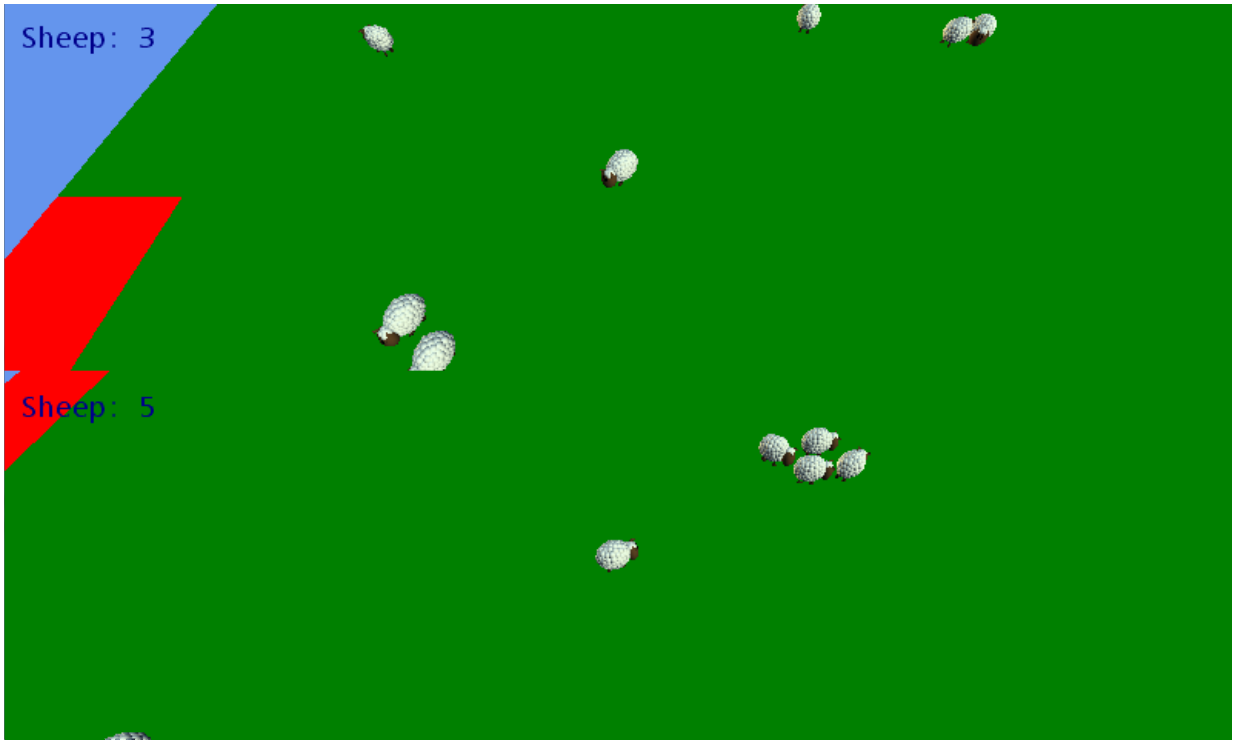
We had to push around a couple of tasks to free resources for the modeling but are still doing OK and work approximately according to our schedule. We will probably start with our first play testing this week. Our next steps are implementing power ups, tuning parameters and adding sound.

We are confident that the models and the level engine will be finished within two weeks. We reserved some time for unforeseen things and don't need it for anything else.

Pictures



Singleplayer with simple HUD and sheep model for the player



Multiplayer with split screen


Flocking test in processing, implementing modified Reynolds algorithm