

# Interim Report

*Game Programming Lab 2016*

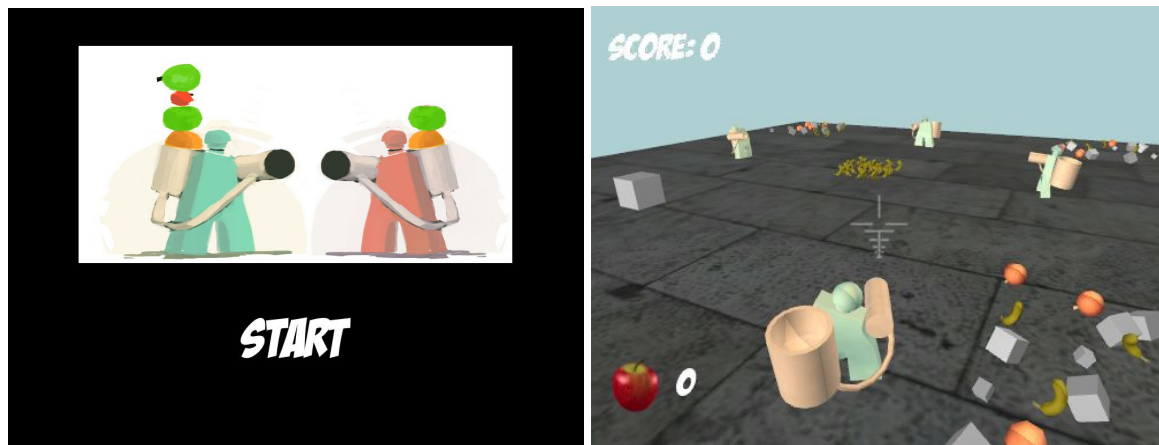
*Jessica Falk, Sandro Lombardi, Sandro Ropelato, Don Schmocker*

*25th April 2016*

## Progress

### Overview

So far, we have accomplished the completion of the “low target”. Major points implemented from this target are for example split screen and multiplayer support, basic detection of player to player collision, simple sound effects and game states as well as illustrations and implementation of three fruits. At the moment, we are working towards the completion of the desirable target. This includes e.g. background music, level design and the implementation of all five fruits.



Start screen and viewport of one player during the game.

### Functional minimum

For a functional minimum, we were expecting to create static player models, the implementation of the “apple” item, a basic map with spawn points and simple shooting and picking up behaviour.

We have implemented relatively fast our first test map consisting of a simple ground model and a static player model. Some basic player walking animations were made and later coupled with the implementation of controls to give the appearance of a walking avatar. Furthermore, we were able to implement a freely movable first person camera really fast. This allowed us to spend some time creating a third person camera and compare the feeling and handling of both against each other. Later we incrementally added spawning of objects, shooting of apples and a really simple hit detection which basically concluded the completion of the functional minimum.

## Low target

For the low target we can divide our goals into two parts. On the design side, we wanted to animate our player models, create icons for the fruits, simple obstacles and models for two new fruits. On the programming part, our goals were to integrate multiplayer support, basic collision detection, implement the logic for two more fruits, add an inventory and implement the pause and end of the game.

We were able to achieve all of these goals and finished the low target.

We started by implementing the multiplayer support to make the game more playable. This was done by adding a split screen to the game that supports up to four player. We then proceeded to create two more fruits: melon and banana

This provided more variety to the game and the need for an inventory. The next step was to implement the dropping of the food, whenever the player got hit. Adding a timer, a score and an endscreen was all that was left to do to make it an actual game.

To be able to create an appealing level design, it is important that the collision detection works correctly. For this, we started implementing our own. Right now, it is still very basic, but already provides the needed detection to ensure that players can neither run into each other nor into obstacles. We intend to further improve the physics to make the game more realistic.

## Desirable target

About half of the goals defined as desirable target have been reached so far. Basic physical behaviour has been implemented for both the characters and the food items. Further points completed are the orange, extended player animations, sound effects and the property that players lose percentually more food dependent on the amount of food they carry.

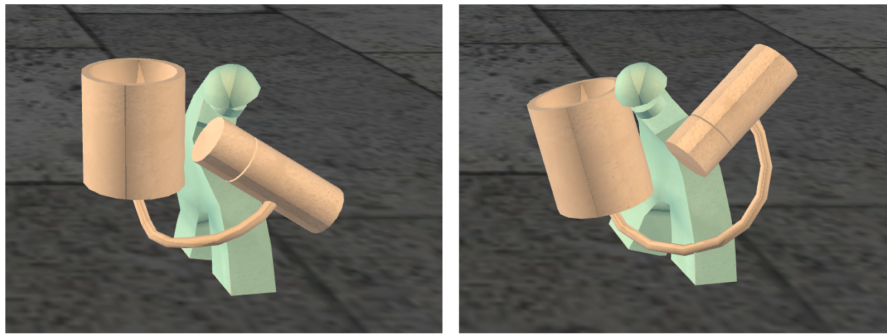
For the completion of this target, we still have to implement the peanuts, obstacles to use in level design, a level selection menu, background music, and the slowdown effect of players carrying too much fruits. We expect the level selection menu to take the most amount of time and the weight property the least amount of time.

## Problems and Challenges

Some tasks proved to be more complicated than expected. We have, however, managed to implement them as planned and did not have to exclude any features so far.

## Animated player models

While it was relatively easy to import and play skeleton-based animations on player models, we realized that visually representing the vertical angle in which a player aims can not be achieved by simply playing one animation. We therefore pursued an approach in which we create every animation twice, once with the player aiming to the bottom and once with the player aiming to the sky. By continuously blending between the corresponding two animations we managed to have the player model pose with the exact angle in which a shot would be fired.



Animation of player standing - looking down and up according to aiming angle..

## Design Revisions

Actually, we did not have any design revisions (at least none we are aware of). It seems that the progress of our project was proceeding flawless without any design-related issues arising. Most of the time, we were able to implement and test two design choices relatively fast to confirm or oppose the idea we had in mind.