Alpha Release Report

Game Programming Lab 2016
Jessica Falk, Sandro Lombardi, Sandro Ropelato, Don Schmocker
9th May 2016

Progress

Overview

We managed to complete all layers for the alpha release. We finished up the desirable layer by completing the level selection menu, adding a slow down effect for players carrying too many fruits, adding background music, implementing the berries and creating additional models for our map design.
The berries were previously peanuts, but we decided that we should be more consistent and only use fruits in our game.

The last target that we had to complete was the high target. We had already implemented the scattered fruit drop and had added first visual effects. We finished this layer up by adding a visual effect to the orange and the berries, creating a settings menu to change the window resolution and designing a new map. We also wanted to implement food rotting, but realized that this would not really add to the gameplay as described later in this report. We also added a few new features that will also be explained in the Design Revisions part.

Problems and Challenges

Collision detection and response

Our technical achievement was the implementation of our own small physics engine. Therefore we handled collision detection and collision response ourselves without the help of an external library. There were multiple challenges related to this achievement which are listed below:

- **Support for different collision shapes**: We wanted to include differently shaped objects. Therefore we had to include different shapes of collision bounding primitives. We ended up with spheres, axis-aligned boxes, oriented bounding boxes and rays. We were able to approximate player and fruit models with bounding spheres and obstacles with bounding boxes. For special cases like camera to wall collision and the berries we had to use ray casting.
- **Collision response**: Objects in our game behave differently while colliding with each other. Fruits bounce off walls while players and cameras slide along them. Corner cases led to more challenges in this area (e.g. what happens if a fruit hits the corner of a bounding box?).
- **Fast and small objects**: Rather late in the development stage, we discovered that fast fruits (e.g. banana) and small objects (e.g. thin walls) lead to a failure in collision detection (objects pass through each other without collision). The reason was that we checked for collisions in every update call. Sometimes, a collision occurred in between two subsequent frames and we were not be able to detect it.
- **Creation and initialisation of bounding primitives**: We had to automatically read the vertices of each loaded mesh in order to determine the dimensions of the bounding box and the bounding sphere. This was a lot harder than initially thought.

Screen resolutions on mac

The retina display of the MacBook caused a lot of trouble with the resolution settings of our game screen. Selecting certain resolutions led to strange behaviour like a partial display of the game scene when the full screen mode was activated.

Design Revisions

Based on the feedback of the last demonstration, we have made some smaller changes in our design.

Fruit Basket

In class, a lot of questions were asked regarding the fruit basket each player would carry on their back. Originally we did not consider the basket to have an impact on the game experience, neither physically nor visually. However, we decided to overthink this design decision and at least visually involve the carried fruits in the basket. Now fruits are shown in the fruit basket where the number of displayed fruits is proportional to the number of fruits in the inventory.

Avocado

There was a suggestion in class to use some kind of magnet to attract nearby fruits. We liked the idea and therefore implemented the avocado. It will spawn 1 to 2 times per game. If picked up, the nearby fruits will automatically fly towards the player and go into his inventory. However, you have a high risk of losing the avocado and then it can be picked up by other players giving them the advantage.

Orange

To further enhance game experience we made the orange repel fruits while active. This way, shooters will have to be more careful when shooting someone under the influence of the orange.

Banana

In the current state, the banana was overpowered. To account for this, we let the banana go rogue with a certain probability with each shot. If a banana goes rogue, it will not return to the owner but rather bounce around the area and harm players standing in their way (even the owner itself). After some collisions the banana will have slowed down and finally become pickable again.

Drop Cascading

In another attempt to enhance game experience and the fun factor, we let dropped food be able to hit nearby players. This way, a chain reaction can occur where nearby player get hit by the dropped fruits of their opponents.

Fruits have weight

To penalize those with huge amounts of fruits, we slow the player down by penalizing them with a maximum of 10% decrease in speed.

Discarding Fruit Rotting

Originally, we planned to let fruits rot and disappear after some time. We decided to discard this idea since the round for one game takes about 2 minutes which makes food disappearance more or less obsolete. Furthermore, we have a global fruit limit which is another reason for this design revision.

Bouncing of fruits off the ground

To add to the visual appeal and a great game experience we let certain fruits bounce off the ground.