

PTVC Submission 2

ANT: Ants never tremble

Gameplay:

3D Geometry (6 Points)

We have implemented a player character (ant) , two static objects (glassBox, table) and four softbodies (blueberry, strawberry, orange, watermelon). Objects were self made in blender and loaded into our game.

Playable (3 Points)

Currently we have a player which can move around and jump/fly, it can also interact/ collide with objects in our space. The player can move fruits around.

Min. 60 FPS and Framerate Independence (3 Points)

Everything that can be frame independent should be correctly implemented.

Win/Lose Condition (3 Points)

Merging the biggest fruits wins you the game. Moving fruit requires energy, if your energy runs out you die and loose the game.

Intuitive Controls (2 Points)

The player moves with WASD and jumps/flies with SPACE, the view can be adjusted by pressing the left mouse button and moving the mouse around or pressing V to capture the cursor, with the wheel of the mouse the player can also zoom in/out. It is also possible to switch to a floating camera by pressing C. When pressing F more fruits start to spawn. ESC closes the game. Some other (not gameplay relevant) controls include: F1: Toggle Wireframe, F2: Toggle back-face culling.

Intuitive Camera (2 Points)

We have two camera models, one that can roam around freely and one that is bound to the player.

Illumination Model (2 Points)

We currently have a light source that highlights the area around the box/table. Each object has an assigned material and is able to correspond to the light. We are using a point light and spotlight.

Textures (2 Points)

Textures can be loaded and used, currently only the table is using a texture.

Moving Objects (2 Points)

The player is a moving object, but the highlight of our game are our softbody fruits which move, jiggle and bounce around the game.

Adjustable Parameters (1 Point)

We have config files that offer customisation for the window size, title and camera. You can also resize the window by dragging it.

Collision Detection (Basic Physics) (6 Points)

We implemented physics using PhysX, our player can not move through static objects, it can also interact with softbodies, making them move around. Our softbodies can also not go through static objects. There are two types of collisions detections Player \longleftrightarrow Fruit and Fruit \longleftrightarrow Fruit. When fruit touch each other they merge to a bigger fruit if they are from the same type which will fall from the sky. If the fruit touches the player it reduces the ant's energy, possibly leading to its death.

Advanced Physics (4 Points)

We implemented softbodies (shown as fruits in the game) using PhysX with this as our reference:

<https://nvidia-omniverse.github.io/PhysX/physx/5.6.0/docs/DeformableVolume.html> .

Heads-Up Display (4 Points)

Is fully implemented allows for Text or Images to be rendered to the screen.

<https://learnopengl.com/In-Practice/Text-Rendering>

Camera Object Tracking (2 Points)

Our player camera follows our player, wherever they go.

Effects:

Shadow Map with PCF (16 Points)

While we do have a spotlight, it does not yet throw shadows on our objects.

Procedural Texture (8 Points)

We were originally planning to make a cloud-like texture using particles, but on the feedback for submission 0 we were told that this might not be the best idea, therefore we implemented the fog using procedural texture, which can be easily seen when flying about the box into the fog.

Hierarchical Animation (4 Points)

We added a small animation for walking, the legs move and the ant sensors move as well a little.

Physically Based Shading (6 Points)

Is successfully implemented. All of our objects have a PBS material.

Additional libraries:

We are currently using the following libraries:

- Assimp - For loading mesh data
- GLFW
- GLM
- PhysX - For handling physics
- spdlog - For better/more useful logging messages
- stb_image - For loading images (used for the textures)
- freetype - Font Bitmaps

Additional notes:

The spawning of the fruit is not optimal which leads to us pre-loading our fruits, this leads to a short wait time at the beginning of the game.