

Comp 477 - Team 1 Project Documentation

Daniel Vellucci	27416288
Ayush Kharade	40042388
Samer Yazbeck	40049573
Lyonel Zamora	27385986
Christian Plourde	26572499

1 Project Summary:

Title: Soft body simulations in 3D Pac man (Arcade Game)

Creating an arcade game, specifically a 3D recreation of pacman using several animation techniques such as soft body simulation, particle systems, path finding & motion capture.

2 Animation Techniques Used

2.1 Soft body cloth Simulation:

The traditional Ghosts in the pac man game are created in this project using a cloth simulation. A linked structure of nodes which independently interact with other objects to simulate cloth. This cloth is instantiated as a plan, on top of a sphere. Using gravity & collisions on all nodes of this linked structure, the cloth covers the sphere, which is used to depict a ghost for our pacman project.

The cloth object is parented to the sphere, so it follows the sphere around the map. All 4 ghosts are created in this similar way, with a sphere that moves around, having a cloth simulation on top of it.

2.2 Collision Detection:

Collision detection is used in many places to implement game play features such as:

- Colliding with walls, ghosts, coins, walls & cherries
- Resolutions according to different collisions.

Collision detection techniques:

- The game map is created using a tile system, where each (X,Y) coordinate represents a tile.
- The map is a grid which makes it easier to use grid-based collision detection system.
- It is a voxel-based world, but there are no objects on the Z axis (Up axis), so essentially, its like a 2D grid.
- Positions of pacman & ghosts are rounded to an integer and compared to calculate collisions.
- Collision resolutions are simple cases:
 - o Decrement player life on collision with a ghost.
 - o Decrement coin count on collision with a coin.
 - o Enable cherry ability on collision with a cherry.
 - o Stop moving when about to collide with a wall. In this case, the next tile in the direction in which player is moving is considered.
 - o If there is a wall in front of the player, you stop moving.
 - o [if we are doing pickable hammer]
 - o Hammer ability that can be picked up to destroy walls.

2.3 Particle Systems:

Particle systems are used to create visual effects for the game.

2.3.1 Breakable Walls

- Brick Walls (or red walls) block player's path in the map.
- These can be broken by running into them which would create a breaking visual by creating smaller brick particles that get created, which expire after a while.
- The smaller brick particle scatter in different direction.

2.3.2 Trail visual

- Particle system is used to create a trail visual that is left behind the player and the ghosts.
- This is similar effect to a comet or a jet trail.
- Helps create illusion of speed or suggesting that player and the ghosts are moving quickly.
- Creates smaller particles that expire after a while.

2.4 Motion Capture from Kinect:

Motion capture can be used as an alternative control model to control the player and play the game. Implemented using the Windows Kinect SDK V2.0.

Motion capture allows to control the player with your fists.

Mimic up, down, left and right motions with your fist while in front of the Kinect to control the player movement.

Player will only change direction if there is a possible route to go to, otherwise would not affect.

Player can still be control with the arrow keys if Kinect is not being used.

3. Data Structures Used

3.1 FreeGlut

Using this main library for creating OpenGL windows and rendering shapes such as spheres for pac man and ghost.

- used to create walls using cubes.
- Used to create spheres for player.
- Used to instantiate particles in the particle systems.
- [if we do hammer ability]Used to create a hammer.

3.2 GLM

Used for computing math and matrix operations.

3.3 Kinect SDK

Using Kinect.h in the project to use the Kinect device and track specifically the hand joint for alternative player controls.

3.4 GLEW

info

3.5 GLFW

Used for taking in Keyboard input while executing the main loop. Keyboard controls are used for controlling the player as well as changing settings of the project such as:

- Change camera style (Top down or follow player)
- Rotate camera for an alternative view of the map.
- Player controls (Arrow keys to go Up / Down / Left / Right)

4. Use Cases / Test Cases

aa

5. References

5.1 Pacman Build, which was used as a base and built upon.

“Link”

5.2 Cloth Simulation

Cloth simulation sample taken from “link”
Was used as a base and integrated into the project.

5.3 Kinect SDK Version 2.0

Project uses Kinect.h that comes from this SDK.

It allows you to take in data from the Kinect, track specific joints, in this case, the arm joint was tracked to get player input.

This is only available for windows