

Overall we are very pleased with how our project turned out. It took more time and our jobs blended together more than we anticipated, but we implemented nearly everything we set out to do.

We have the player's paddle in place and moving on the x and y axis. The ball is affected by gravity, and collides with the walls and paddle. We were able to implement our scoring mechanic like we wanted and have the back wall reset the ball and rally. Our simulation loop is in place and is updating the transforms of the moving parts.

We would have liked to find better textures, but we were much more concerned with making certain our game's mechanics worked correctly. We had a lot of trouble dealing with bullet and collisions and that ended up costing us a good chunk of our time.

Final Report Comments are in Red

- **Overall Design**

Basically a racquetball-esque game

- Scoring (Implemented)
 - Keep track of rally (Implemented)
 - Current
 - Global high score
 - Resets upon a fault (ball goes behind) (Implemented)
- Look
 - Room (Was created and collides with ball.)
 - Racquetball court textures (Unable to implement)
 - Cubic room shape
 - Ball (Was created. Physics work on the ball.)
 - Spherical
 - Vibrant color, possibly red (Unable to implement)
 - Rubbery texture (for a bouncy look) (Unable to implement)
 - Obvious shadow (Implemented)
 - Paddle
 - Similar to ping pong paddle (Ended up going rectangular)
 - Relatively wide (Implemented)
- GUI
 - Score (Rally count) (Implemented)
- Sound
 - Collision noises (Implemented)
 - Ball
 - Paddle
 - Wall
 - Swing noise (Implemented)
 - Maybe some music (Unable to Implement)
- Additional Aspects
 - Paddle(Implemented)

- Camera behind the paddle (Implemented)
 - 3rd person style
 - Moves side to side with the paddle
- Swing upon a mouse click (Implemented)
 - Possibly a left click foreswing and a right click backswing
- Can move in 2 dimensions (Implemented)
 - Side to side
 - Up and down
 - Locked on the other axis
- **Software architecture and plan**
 - Top view
 - Ogre (Works well.)
 - Renders scene and changes to scene
 - OIS
 - Bullet (Implemented)
 - Computes physics
 - CEGUI (Implemented)
 - Create widgets for our GUI
 - SDL (Implemented)
 - Sound library
 - Total time: 1 hour per library to set up
 - Loops
 - Game (Implemented)
 - Player input
 - Sounds
 - Scoring logic
 - Simulation (Simulation loop works properly. It is run at 60 fps.)
 - Collision physics (Implemented and refined.)
 - Rendering (Scene renders and updates properly.)
 - Update scene
 - Total time: 5-6 hours per loop to set up
 - Object structures
 - Ball object files (In place.)
 - Paddle/Player object files (In place.)
 - Game object files (In place.)
 - Total time: 6 hours
 - Game state
 - Game object files (Implemented)
 - Scoring
 - High scores
 - Total time: 2 hours
- **Division of Labor**
 - Kamron

Kamron Ledet - kal2763
Caleb Barnwell - cmb5827
Mitchell Kaysen - mk34534

- CEGUI (Implemented)
- Camera (Implemented. Caleb helped with this.)
- Sound (Implemented)
- Caleb
 - Bullet (Implemented. Mitchell helped with this.)
 - Collision (Implemented. Mitchell helped with this.)
 - Scoring (Implemented. Kamron helped with this.)
- Mitchell
 - Set up object files (Implemented)
 - Paddle swinging (Implemented. Caleb helped with this.)
 - Level Setup (Implemented. Caleb helped with this.)
- Planned Meetings:
 - Tues, Thurs, Fri, Sundays
- Interfaces
 - Github