Technical report

Camera and character

The project involves 3rd person camera, which gives an ability rotate around the character when motionless.
When moving, the character rotates to face the same direction as the camera does.

Collisions

Camera uses sphere collisions, and character - capsule collisions to realistically interact with the environment.

Sphere-plane collisions(for camera)

$Given$:

$$\hat{O}^{(3)} = O - sphere's\ center(normalized),$$

$$r^{(0)} = r - sphere's\ radius,$$

$$\hat{P}^{(1)} = P\ \ - collision\ plane(normalized)$$

$Find$:

$$O_1^{(3)} = O_1 - sphere's\ center\ after\ collision$$

$Solution$:
1) $Getting\ distance\ from\ sphere's\ origin\ to\ the\ plane$

$$d = |O \vee P| = H^{-1}(H(O) \wedge H(P))$$

2) $d < r => Sphere(O, r)\ \cap\ P$:

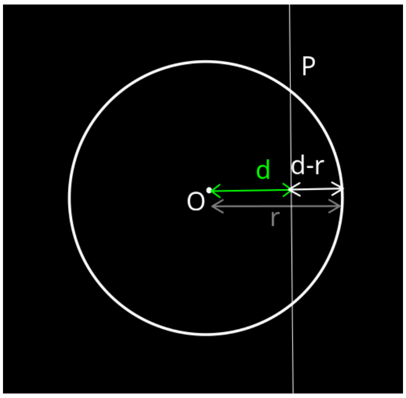
2.1) $Retrieving\ the\ translation\ motor$:

$$T = 1 - \frac{1}{2}t(-e_0),$$

$$t = (d - r)\hat{P}^{(1)}$$

$\mathbf{NOTE}: The\ e_0\ coefficient\ of\ P\ is\ irrelevant,\ because\ it\ turns\ into\ void\ regardless.$

$\mathbf{NOTE}: See\ rationale\ behind\ translation\ by\ (d - r)\ in\ the\ Figure\ 1.$



$$Figure\ 1.\ Sphere - plane\ collision\ resolution$$

$2.2)\ Translating\ the\ sphere's\ origin$:

$$O_1^{(3)} = < T\ O\ T >_3 \ \ (gep) - solution$$

Capsule-plane collisions(for character)

$Given$:

$$h^0 - capsule's\ height$$

$$r^0 - capsule's\ radius$$

$$\hat{O}^3 = O = O_x\ e_{021} + O_y\ e_{013} + O_z\ e_{032} + O_w\ e_{123} - capsule's\ origin$$

$Find$:

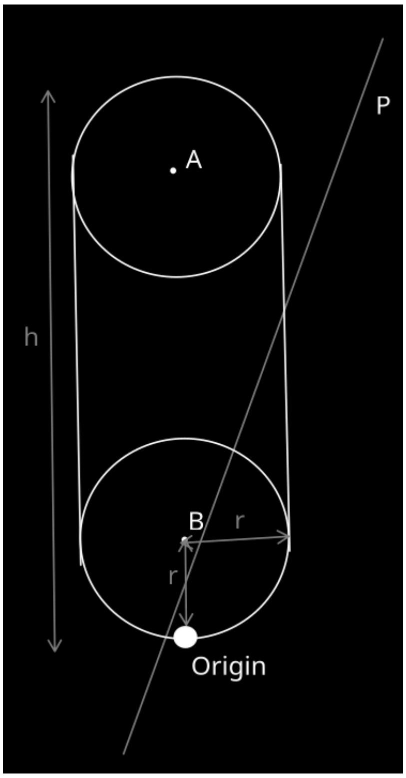
$$O_1^{(3)} = O_1 - capsules\ center\ after\ collision$$

$Solution$:

$1)\ Finding\ A^3\ and\ B^3.\ See\ Figure\ 2\ for\ reference$

$A^3 = A = O_x\ e_{032} + \textbf{(h-r)}\ O_y\ e_{012} + O_z\ e_{013} + O_w\ e_{123}$

$B^3 = B = O_x\ e_{032} + \mathbf{r}\ O_y\ e_{012} + O_z\ e_{013} + O_w\ e_{123}$



$$Figure\ 2.\ Capsule - plane\ collision$$

$2)\ Retrieving\ the\ closest\ sphere$

$$d_A = |A \vee P|,\ d_B = |B \vee P|$$

$$|d_A| < |d_B| => A\ is\ closer\ and\ vice\ versa$$

$3)\ Performing\ collision\ handling\ on\ the\ closest\ sphere.$
See [Sphere-plane collisions(for camera)].
$\mathbf{NOTE}: Instead\ of\ applying\ translation\ to\ sphere's\ origin,\ it\ should\ get\ applied\ to\ the\ capsule's$

Lazer shooting

Lazer reflection

Formulas used

- **Transformations** (for controls and collision resolution)

$$R * T * X * (R * T)^{-1},$$

$$R\&T - motors,$$

$$X \subset R^n, n \in [1, 3]$$

- **Distance acquisitions** (for collision detection)

$$d_1 = |A_n \vee B_m|,$$

$$n, m \in [1, 3]$$