CMSC 425: Solutions to the First Midterm Exam

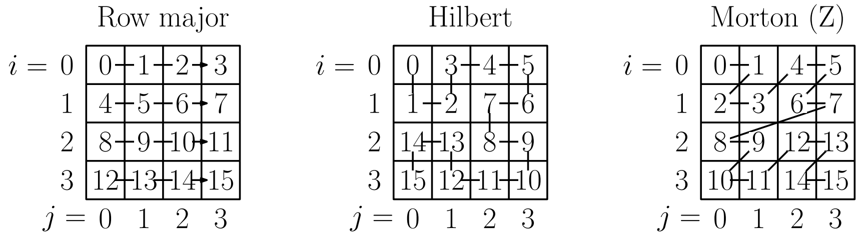
Solution 1:

1.1 Declaring a Unity game object (or more accurately its RigidBody component) to Kinematic means that the object is entirely under the control of the attached scripts. Unity physics (forces and collisions) do not affect the object's motion.

1.2 $a - 3(b - a)$ is equivalent to the affine combination $4a - 3b$. Is this *not* a convex combination, since the coefficients 4 and -3 do not lie in the interval $[0, 1]$.

1.3 Using Space.Self means that the transformation is applied relative to the object's local coordinate system (as given by its transform), and using Space.World means that the transformation is applied relative to the world's coordinate system.

1.4 See Fig. 1.



Row major

$i = 0$	0	1	2	3
1	4	5	6	7
2	8	9	10	11
3	12	13	14	15
$j =$	0	1	2	3

Hilbert

$i = 0$	0	3	4	5
1	1	2	7	6
2	14	13	8	9
3	15	12	11	10
$j =$	0	1	2	3

Morton (Z)

$i = 0$	0	1	4	5
1	2	3	6	7
2	8	9	12	13
3	10	11	14	15
$j =$	0	1	2	3

Figure 1: Answer to Problem 1.4: Ordering cells of a grid.

1.5 Without blending, cracks and gaps may develop in the mesh (e.g., between the upper-arm skin and lower-arm skin when the elbow is bent).

Solution 2:

2.1 Given a point p, it lies within the collider if it satisfies two conditions. First, it lies within distance r of the center point c, and second, it does not lie on the same side of c as indicated by the vector $\vec{u}$. Define $\vec{v} = p - c$ to be the vector from c to p. If $\|\vec{v}\| > r$ then p is too far away, and so does not lie within the collider. Otherwise, we test whether the angle between $\vec{v}$ and $\vec{u}$ is acute. If so, p lies the same side of c and indicated by $\vec{u}$. To test this, we check whether $\vec{v} \cdot \vec{u} > 0$. If both conditions fail, then p lies within the collider.

2.2 True: If condition-1 holds, then in order for the two complete spheres to intersect, the southern hemisphere of collider-1 must intersect the northern hemisphere of collider-2. The second condition enforces this.

2.3 True: If $c_{1y} < c_{2y}$ then every point of collider-1 lies below any point of collider-2, so the colliders cannot intersect. If $\mathrm{dist}(c_1, c_2) > r_1 + r_2$ then the spheres do not intersect, so the hemispheres cannot intersect either.

Solution 3: The projectile's x- and z-coordinates vary linearly with time t and the y-coordinate follows an inverted parabola. Letting $g = 9.8\text{m}/\text{sec}^2$ denote the gravitational acceleration, we have

$$x(t) \;=\; p_x + t v_{0,x}, \qquad z(t) \;=\; p_z + t v_{0,z}, \qquad y(t) \;=\; -\frac{g}{2}t^2 + v_{0,y}t + h.$$

Let t denote the time when the projectile hits the wall plane ($x = 0$). We will compute the time of impact, determine whether this is a valid time, and if so, compute the coordinates where the projectile hits the wall. We will then determine whether these coordinates lie within the portal.

3.1 The time to hit the wall is determined by considering the x-coordinate of the projectile's position. We seek the value of t such that $x(t) = p_x + t v_{0,x} = 0$, that is $t^* = -p_x/v_{0,x}$. We may ignore t^* if either $v_{0,x} = 0$ (meaning it is moving parallel to the wall and so never hits it) or if $t^* < 0$ (meaning that the projectile is moving away from the wall and does not hit it in the future).

3.2 The point at which the projectile hits the wall is

$$q \;=\; (x(t^*), y(t^*), z(t^*)) \;=\; (0,\; -\frac{g}{2}(t^*)^2 + v_{0,y}t^* + h,\; p_z + t^* v_{0,z}).$$

3.3 The projectile passes through the portal, if $\text{dist}(q, c) \le r$. Expanding out the formulas this is equivalent to

$$\left(-\frac{g}{2}(t^*)^2 + v_{0,y}t^* + h - c_y\right)^2 + \left(p_z + t^* v_{0,z} - c_z\right)^2 \le r^2.$$

Thus, if $t^* \ge 0$ and the above inequality is satisfied, then the projectile flies through the portal, and otherwise it does not.

Solution 4:

4.1 The homogeneous coordinates of p with respect to each of the frames are:

(i) Propeller frame: $p_{[c]} = (5, 0, 1)^{\mathsf{T}}$

(ii) Copter frame: $p_{[b]} = (0, 13, 1)^{\mathsf{T}}$

(iii) World frame: $p_{[a]} = (10, 20, 1)^{\mathsf{T}}$

(The "T" superscript indicates that these are all column vectors.)

4.2 Relative to b's frame, we have $c.x = (0, 1, 0)$, $c.y = (-1, 0, 0)$ and $c.o = (0, 8, 1)$ (all column vectors in homogeneous coordinates). The matrix $T_{[b\leftarrow c]}$ has these as its three columns. Similarly, relative to a's frame, we have $b.x = (1, 0, 0)$, $b.y = (0, 1, 0)$ and $b.o = (10, 7, 1)$. The matrix $T_{[a\leftarrow b]}$ has these as its three columns.

$$T_{[b\leftarrow c]} \;=\; \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 8 \\ 0 & 0 & 1 \end{pmatrix} \quad T_{[a\leftarrow b]} \;=\; \begin{pmatrix} 1 & 0 & 10 \\ 0 & 1 & 7 \\ 0 & 0 & 1 \end{pmatrix}.$$

As a check of your work, you can verify that $p_{[b]} = T_{[b\leftarrow c]}p_{[c]}$ and t $p_{[a]} = T_{[a\leftarrow b]}p_{[b]}$.

4.3 To get from c to a, we go from c to b and then from b to a. Thus, $T_{[a \leftarrow c]} = T_{[a \leftarrow b]} T_{[b \leftarrow c]}$. To get from a back to c we invert this matrix: $T_{[c \leftarrow a]} = T^{-1}_{[a \leftarrow c]}$.

4.4 In order to achieve the desired motion, we first convert $p_{[a]}$ to $p_{[c]}$ by applying the transformation $T_{[c \leftarrow a]}$. We then apply the rotation transformation $\mathrm{Rot}(-\varphi)$ about the propeller's frame (negating φ because the rotation is clockwise), then apply $T_{[b \leftarrow c]}$ to convert to the copter frame and then apply a rotation of $\mathrm{Rot}(\theta)$, then apply $T_{[a \leftarrow b]}$ to convert from to the world's frame and apply a translation $\mathrm{Tr}(b' - b)$. We are now in the coordinate frame for the world's frame, so no further conversions are needed. Since all this proceeds from right to left, we have

$$p'_{[a]} \;=\; \big(\mathrm{Tr}(b' - b) \cdot T_{[a \leftarrow b]} \cdot \mathrm{Rot}(\theta) \cdot T_{[b \leftarrow c]} \cdot \mathrm{Rot}(-\varphi) \cdot T_{[c \leftarrow a]}\big) p_{[a]}.$$

The final matrix is the quantity in parentheses.

Solution 5:

5.1 The matrix $T_{[w \leftarrow f]}$ is the 4×4 matrix whose four columns are $f.x$, $f.y$, $f.z$, and $f.o$ (from left to right), all given as 4-element homogeneous column vectors with respect to world coordinates. For the example shown in the figure, note that $f.x$ is directed along the world's z-axis and $f.z$ is directed along the world's $-x$-axis. Hence, we have

$$T_{[w \leftarrow f]} \;=\; \begin{pmatrix} 0 & 0 & -1 & f.x \\ 0 & 1 & 0 & f.y \\ 1 & 0 & 0 & f.z \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

5.2 Analogous to 5.1, but using $g.x$, $g.y$, $g.z$, and $g.o$, instead. For the example shown in the figure, note that $g.x$ is directed along the world's $-x$-axis and $g.z$ is directed along the world's $-z$-axis. Hence, we have

$$T_{[w \leftarrow g]} \;=\; \begin{pmatrix} -1 & 0 & 0 & g.x \\ 0 & 1 & 0 & g.y \\ 0 & 0 & -1 & g.z \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

5.3 Just invert the matrices: $T_{[f \leftarrow w]} = T^{-1}_{[w \leftarrow f]}$ and $T_{[g \leftarrow w]} = T^{-1}_{[w \leftarrow g]}$. Computing inverses by hand is difficult, but for example, if you work out the case of $T_{[f \leftarrow w]}$, you get

$$T_{[f \leftarrow w]} \;=\; T^{-1}_{[w \leftarrow f]} \;=\; \begin{pmatrix} 0 & 0 & 1 & -f.z \\ 0 & 1 & 0 & -f.y \\ -1 & 0 & 0 & f.x \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

5.4 Here is the idea. First, convert q from world coordinates to entry-portal coordinates: $q_{[f]} = T_{[f \leftarrow w]} q_{[w]}$. Now, the key insight is that, since q' is the point corresponding to q in the exit portal, it's coordinates with respect to g are the same as q's coordinate with respect to f.

That is $q'_{[g]} = q_{[f]}$. All that remains is to convert back to world coordinates, which we can do with $q'_{[w]} = T_{[w\leftarrow g]} q'_{[g]} = T_{[w\leftarrow g]} q_{[f]}$. Putting this together, we have the final answer

$$q'_{[w]} \;=\; T_{[w\leftarrow g]} T_{[f\leftarrow w]} q_{[w]}.$$

Setting $M = T_{[w\leftarrow g]} T_{[f\leftarrow w]}$ gives the entry-to-exit transformation. It can be applied to points, vectors, or anything else that passes through the portal.

For example, in the example from the figure, this would yield the portal matrix $P = T_{[w\leftarrow g]} T_{[f\leftarrow w]} =$

$$\begin{pmatrix} -1 & 0 & 0 & g.x \\ 0 & 1 & 0 & g.y \\ 0 & 0 & -1 & g.z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 & -f.z \\ 0 & 1 & 0 & -f.y \\ -1 & 0 & 0 & f.x \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & -1 & g.x + f.z \\ 0 & 1 & 0 & g.y - f.y \\ 1 & 0 & 0 & g.z - f.x \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

As a quick check of correctness, it is easy to see that multiplying this matrix times the column vector $P(f.x, f, y, f.z, 1)^{\mathsf{T}} = (g.x, g.y, g.z, 1)^{\mathsf{T}}$, thus a point entering the center point of the entry portal is mapped to the center point of the exit portal, as desired.

That's Portal in a nutshell!