[Question 1]

(a)
```
while (true) {
     for (int deg = 0; deg < 360; deg++) {
        var rad = Mathf.Deg2Rad * deg;
        var x = (stA + stB) *Mathf.Sin(rad);
        var y = (stA + stB) *Mathf.Cos(rad);
        this.transform.position = new Vector2(x,y);
  }
```

(b)
```
deg = 0;
 while (true) {
     var rad = Mathf.Deg2Rad * deg;
     var x = (stD * 0.5 * 1/360) *Mathf.Sin(rad);
     var y = (stD * 0.5 * 1/360) *Mathf.Cos(rad);
     var z = stC--;
     this.transform.position = new Vector3(x,y,z);
     deg++;
  }
```

(c) To improve this, use Ellipsoid Particle Emitter to make circular motion, and flocking from highest position to 0 on Z.

[Question 2]

(a) RST = (4 - | 3 - stD |, 5 - | 5 - stE |, -8 - | 6 - stF |)
(b) Mario designed by 2D model, Olivia designed by 3D model. Mario applies light sources to the rendered planar model itself, but in Olivia's case, the amount of light sources received varies depending on the bending of the model.
(c) Use Shadow Distance property on Mario model with point light follow Mario.

[Question 3]

(a) P1 = (100cm cos ø, 100cm sin ø), K = ((100cm cos ø - 60 - stE) / 2, (100cm sin ø + 10 - stF) / 2)

(b) Beginning of Active Frames = t1, Beginning of Recovery = t2, End of Recovery = t3, Beginning of Active Frames 2 = t4. when Button pressed, do Active Frames, and if Button Realesed on t2, goto Startup, and immediately move to t4.

```
if (Input.KeyDown(PunchKey)) Punch();

void Punch() {
        if (isCurrentlyBusy) return;
        if (Input.KeyDown(Forward))
                PunchForward();
        else if (Input.KeyDown(Down))
                 PunchDown();
        else if (Input.KeyDown(Jump)) {
                PunchUp();
                Jump();
        }
        else RegularPunch();
}
```

Punch action - Track A.
Cancel punch action at a certain timeframe(suppose at 0.6s) -ΔT
Return to ideal action - Track B
Kick action - Track C

```
list = [a, b, c, ΔT]
[a, b, c, ΔT] -> press a
[b, c, ΔT] -> press b
[c, ΔT] -> press b
[a, b, c, ΔT] -> press a
[b, c, ΔT] -> press b
[c, ΔT] -> press c
[ΔT] -> press ΔT
[] done! -> ATTACK COMBO
```

This is also only a simple case, not exactly efficient, and a bit error-prone.If you were checking for a word like “IAACAB”, then pressing A in the wrong moment doesn’t kick you back to the start, but to position 1.

[Question 4]

(a) $Rx(ø) = [[1\ 0\ 0], [0\ \sqrt{3}/2, -1/2], [0, 1/2, \sqrt{3}/2]]$
$Ry(ø) = [[\sqrt{3}/2, 0, 1/2], [0\ 1\ 0], [-1/2, 0, \sqrt{3}/2]]$
$Rz(ø) = [[\sqrt{3}/2, -1/2, 0], [1/2, \sqrt{3}/2, 0], [0\ 0\ 1]]$

(b) $P1P2 = (5, 0, 0) - (0, 0, 5) = (5, 0, -5)$
$P1P3 = (5, 0, 0) - (10, 0, 5) = (-5, 0, -5)$,
$P1P2 \times P1P3 = 0$

[Question 5]

(a) Gonna use Answer of (a) of Question 4. Final result of that roation $R = Rx(ø)\ Ry(ø)\ Rz(ø)$. $R = [[3/4, -\sqrt{3}/4, (2+3\sqrt{3})/8], [\sqrt{3}/4, (1+2\sqrt{3})/8, (3-2\sqrt{3})/8], [-1/2, \sqrt{3}/4, 3/4]]$.
(b) The advantage of using a matrix is that multiple transformations can be combined into one via matrix multiplication.

Now, if the purpose is simply to bring translation on the table, then I'd say (x, y, z, 1) instead of (x, y, z, w) and make the last row of the matrix always [0 0 0 1], as done usually for 2D graphics. In fact, the 4-component vector will be mapped back to the normal 3-vector vector via this formula:

$$\begin{bmatrix} x(3D) \\ y(3D) \\ z(3D) \end{bmatrix} = \begin{bmatrix} x / w \\ y / w \\ z / w \end{bmatrix}$$

This is called homogeneous coordinates. Allowing this makes the perspective projection expressible with a matrix too, which can again combine with all other transformations.