

**Ejercicio 1 (15pts)**

- A. ¿Qué es el "pipeline de renderizado"? Describa brevemente las (3) etapas principales del mismo.
- B. ¿Qué datos de entrada y salida se utilizan habitualmente en cada una de ellas? (considere un caso simple como el del TP base de la materia, donde hay un modelo 3D con textura e iluminación).

**Ejercicio 2 (15pts)**

- A. Escriba la ecuación basada en el modelo de Phong utilizada habitualmente para calcular iluminación en OpenGL (solo la ecuación, en este punto no hace falta explicar las componentes).
- B. Si tengo una escena que modela el aula en la que está rindiendo (hay mucha luz del sol ingresando por las ventanas, y posiblemente varias luces encendidas en el interior). ¿Cómo modelaría esta situación con su ecuación? (es decir, qué valores asignaría a qué constantes para un caso como este)
- C. Si tengo en la misma escena dos objetos de un mismo color rojo, pero diferentes: la tapa de plástico del marcador que el docente utiliza en el pizarrón, y la remera roja de un alumno. ¿Cómo modelaría esta situación con su ecuación? ¿Qué cambia entre un objeto y el otro?

**Ejercicio 3 (25pts)**

- A. ¿Por qué es conveniente representar transformaciones como matrices?
- B. Considere las matrices *model*, *view* y *projection* que se utilizan habitualmente, ¿de qué espacio a qué otro espacio transforma cada una de ellas?
- C. Una aplicación permite mover puntos sobre el plano xy ( $z=0$ ). Cuando el usuario hace click, la aplicación debe determinar en qué punto de dicho plano hizo click. La vista de la aplicación es 3D, y usted dispone de las matrices utilizadas para renderizar (*model*, *view*, *projection*). ¿Cómo calcularía el punto?

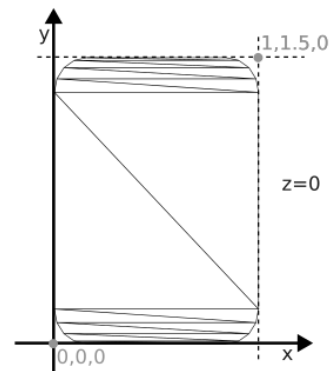
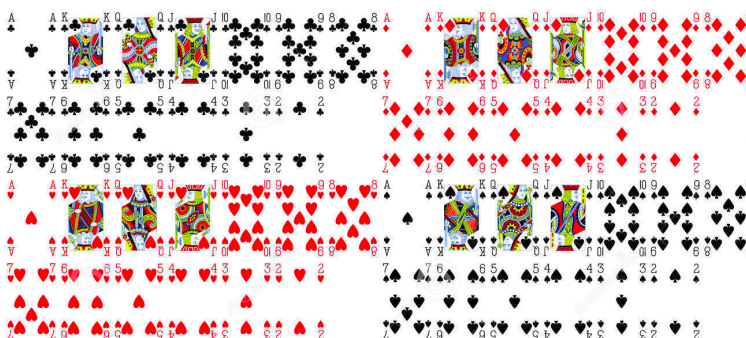
*Nota:* el dato es el punto en la ventana donde el usuario hace click ( $x$  e  $y$  del píxel).

*Ayuda:* hay un rayo que parte de la cámara y pasa por el "píxel" donde el mouse hizo click. Si encuentra cómo reconstruir dos puntos de ese rayo en el sistema de coordenadas 3D adecuado, podría reducir el problema a una "simple" intersección rayo-plano.

**Ejercicio 5 (20pts)**

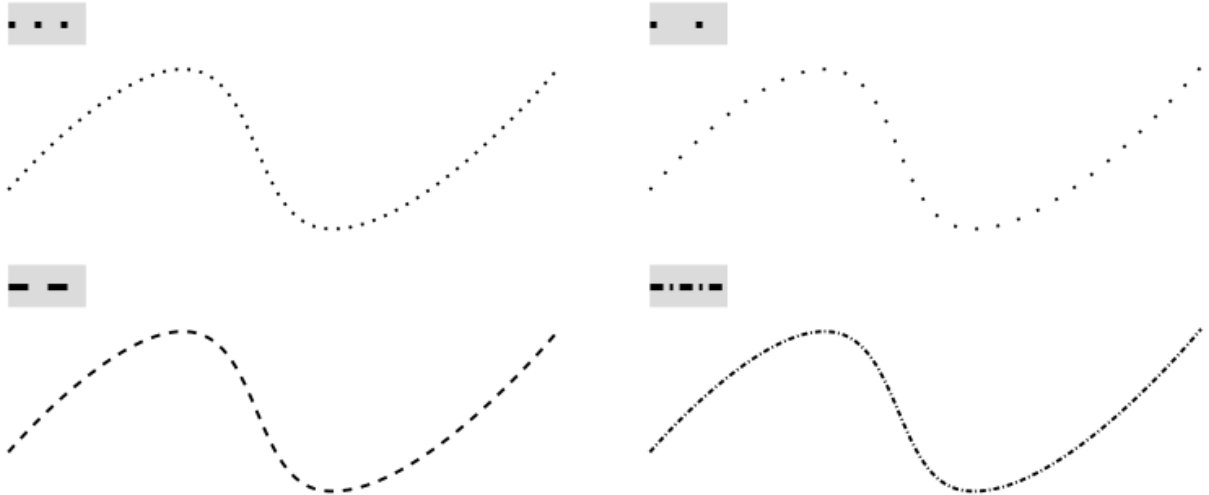
Si programo un FPCG (un juego de cartas en primera persona), donde las cartas se representan cada una con un modelo plano con la textura adecuada, y se colocan y deslizan sobre una mesa también plana.

- A. ¿Qué "artefactos" podría observar al renderizar la escena con OpenGL? ¿A qué se debe? ¿Cómo podría corregirlo?
- B. Si tengo una sola textura que tiene todas las cartas formando una grilla, el modelo de carta que uso la dibuja entre  $(0,0,0)$ ,  $(1,1.5,0)$ , y para aplicar la textura utilizo mapeo plano; cuando quiera dibujar la carta de la fila  $i$ , columna  $j$ , ¿cómo defino los planos?



**Ejercicio 6 (25pts) Ejercicio difícil/integrador... rasterización, curvas y transformación afín)**

A. Describa un algoritmo para rasterizar una curva utilizando un “estilo de línea” no continuo. Por ej:



Nota: Complete además la definición del problema describiendo cómo modelaría el patrón de la línea (cómo representaría ese dato de entrada para su algoritmo).

Ayuda: evalúe si es conveniente resolver este problema con una técnica vectorial o raster, y en función de su respuesta elija cómo representar el patrón.

B. Si tiene una función `drawMarker(int i)` que dibuja un “marcador” siempre orientado en la dirección +y, encerrado entre  $\{0,0\}$  y  $\{1,1\}$  (donde el argumento  $i$  permite elegir entre diferentes marcadores) ¿cómo la utilizaría para dibujar los “marcadores” en los extremos de la curva?



C. ¿Cómo modificar la función para poder “escalar” el patrón, el grosor de la línea, y los marcadores?

