Tarea 1

Instrucciones Generales

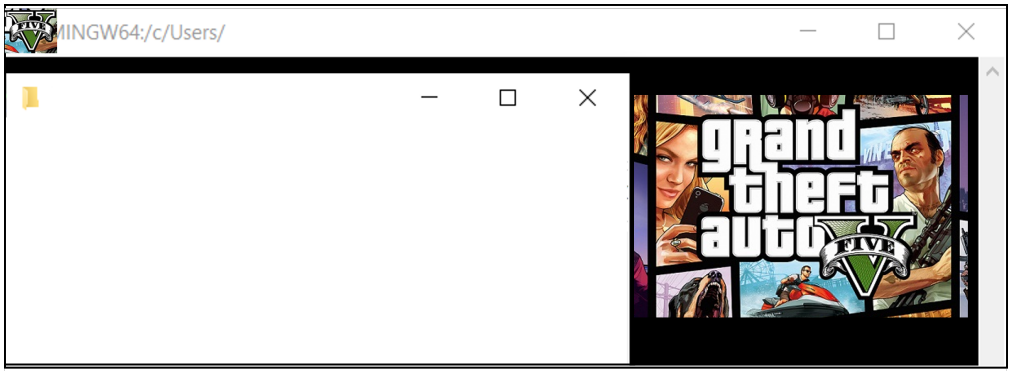
- La tarea es estrictamente individual.
- Este enunciado presenta varias opciones, escoja e implemente solo una de ellas.
- Guarde todo su trabajo para esta tarea en una carpeta de nombre tarea1x, con x indicando la opción que haya escogido y entregue eso en U-cursos.
- El uso de Git es altamente recomendado, aunque no es obligatorio.
- DEBE utilizar: Python 3.7 (o superior), Numpy, OpenGL core profile, GLFW.
- *Las imágenes presentadas son solo referenciales, utilice un estilo propio para su trabajo.*

Entregables

- Código que implemente su solución (4.5 puntos)
  - Su versión final DEBE utilizar archivos *.py, NO jupyter notebooks.
  - Incluya TODO lo que su código necesita, incluyendo archivos proporcionados en cátedras o auxiliares.
- Reporte de documentación de entre 1 y 2 planas. Formato pdf. Detalles disponibles en primera clase. (1.2 puntos)
- Video demostrativo de 20-30 segundos. (0.3 puntos)

Opción A: GUI Y Widgets!

La comunidad computina está cansada de los entornos de escritorio clásicos como GNOME y le pidieron al DCC desarrollar uno propio usando OpenGL. Como el equipo está ocupado planeando el próximo /PaseoDCC, decidieron dejarlo de tarea 1 para el ramo de Computación Gráfica el semestre de vuelta a la presencialidad. Ahora le toca a usted desarrollar esta tarea.



Su programa debe mostrar un escritorio con dos o más archivos. Uno de ellos abre una terminal que muestra cierto contenido, y otro muestra una una carpeta.

Considere lo siguiente:

- Todos los widgets (aplicación y directorios) deben tener un botón de cerrado X, cerrando la aplicación cuando se presiona. Además, se amplian al tamaño de la pantalla cuando se presiona el cuadrado. El botón de minimizado _ no es necesario. Se recomienda inicializar las ventanas con un tamaño estándar distinto del tamaño total para que se nota el efecto de la ampliación.
- Los widgets se pueden agrandar si se toman desde un extremo con el mouse, haciendo click primario y manteniendo el click hacia la derecha o hacia arriba. No es necesaria la ampliación diagonal (Aunque se considera un bonus si lo incluye)
- Si un widget se toma desde el borde superior, se puede trasladar manteniendo el botón primario del mouse apretado.
- Debe agregar una textura a la aplicación que abra. En el ejemplo de la imagen, se ve una textura del GTA V. Esta figura debe incluir una animación de rotación, cuya forma de rotación sea a su elección.
- Debe tener una textura en el escritorio para el directorio y para la aplicación, que al hacerles click de alguna forma que usted elija (puede ser doble click, click derecho, click izquierdo), se abre el widget respectivo.

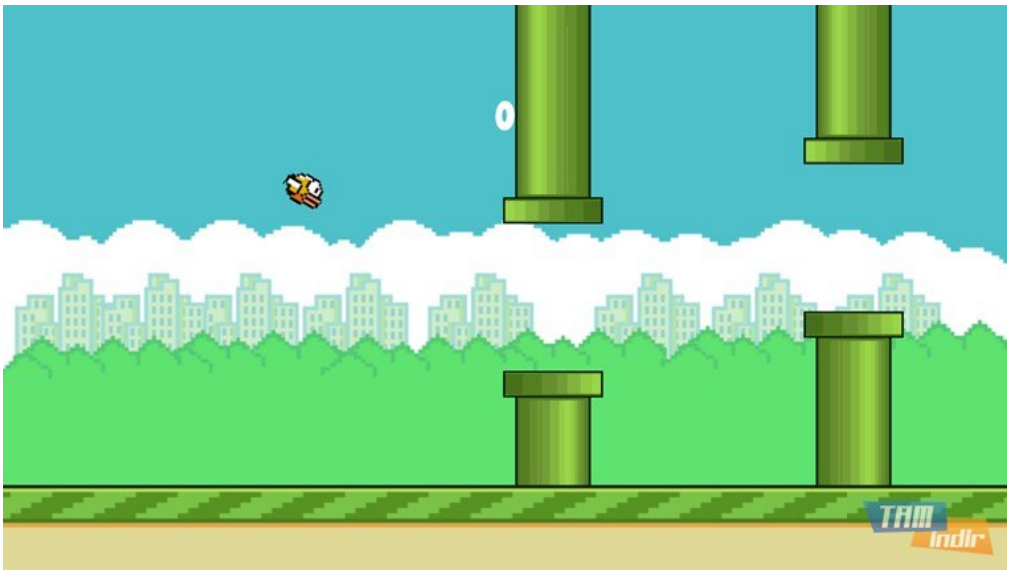
- En un extremo del escritorio (queda a su criterio cual) hay una barra que indica las aplicaciones que están abiertas (Taskbar). Esta barra muestra los logos de carpeta y/o la aplicación según sea el caso, y deben aparecer cuando se abre el programa, así como desaparecer cuando se cierra el widget con la X.

Puntuación

- Escritorio con texturas de directorio y aplicación: 0.6 puntos.
- Widgets se abren al interactuar con elementos del escritorio: 0.4 puntos.
- Widgets se pueden maximizar: 0.5 puntos
- Widgets se pueden cerrar: 0.4 puntos
- Widgets tienen una textura que indica la aplicación que está abierta en una de sus esquinas (Como el logo de GTA en la ventana de la imagen o la carpeta en la imagen): 0.3 puntos
- Widgets se pueden ampliar horizontal y verticalmente manteniendo el click: 0.7 puntos. (Bonus de 0.3 pts a toda la tarea si lo puede ampliar diagonalmente, por si comete errores en el informe o video)
- Aplicación contiene textura. Esta textura es la misma que se ve en el escritorio y se puede clickear, pero más grande: 0.4 puntos
- Textura de la aplicación gira de alguna manera: 0.3 puntos
- Barra de tareas (taskbar) en un extremo de la pantalla: 0.4 puntos
- Aplicación y carpeta aparecen en la barra de tareas cuando se abren, y desaparecen de ahí cuando se cierra el widget: 0.5 puntos

Opción B: Flappy Bird

Su mejor amigo le cuenta que a él le encantaría comprarse un juego nuevo, que se llama 'Flappy Bird', pero que es muy caro para él. Usted recuerda que el cumpleaños de su amigo es en 3 semanas, pero usted tampoco tiene el dinero suficiente para comprarlo. Como usted está tomando el curso de Computación Gráfica, decide aprovechar la oportunidad y desarrollar un videojuego muy similar, para así regalárselo a su amigo.



Su juego se debe ejecutar con la siguiente llamada:

```
python flappy_bird.py N
```

Siendo N el puntaje necesario para que el jugador gane.

Considere lo siguiente:

- Debe implementar los modelos necesarios para el juego, es decir, al personaje principal (no necesariamente puede ser el clásico pájaro), los pilares donde podría chocar el personaje y el suelo.
- El personaje principal debe 'volar' con la flecha hacia arriba.
- El fondo debe utilizar una textura y también considerar una transformación de traslación horizontal hacia la izquierda para simular el avance del personaje.
- Dado lo anterior, su personaje debe estar fijo horizontalmente, y moverse solo en el eje vertical.

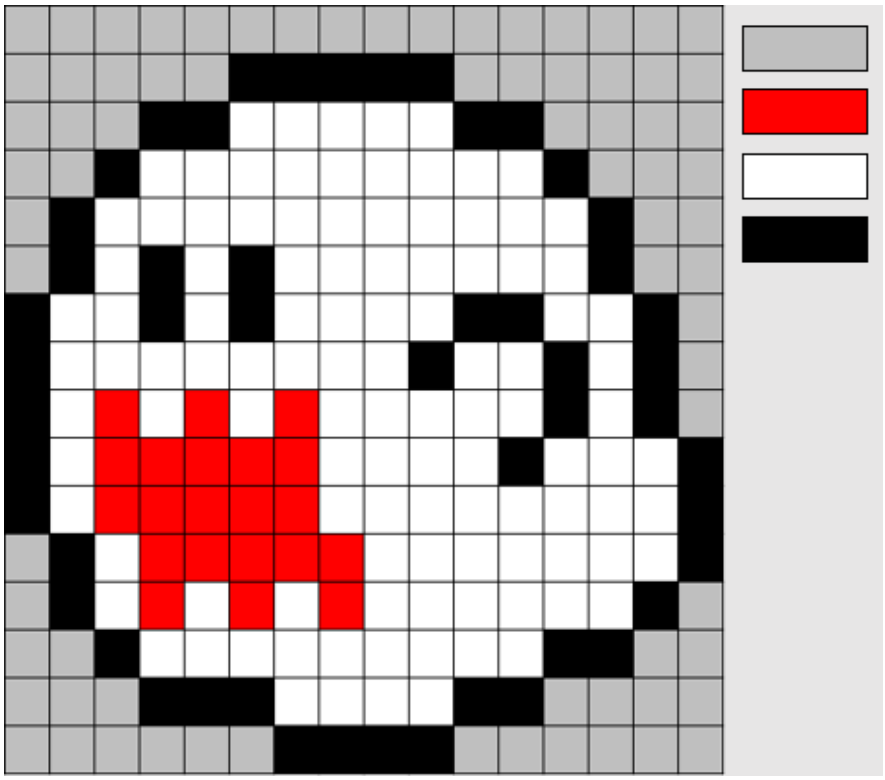
- Debe implementar un contador en el juego que indique cuantos obstáculos ha pasado el jugador.
- Para el movimiento del personaje, considere que la velocidad con que el personaje se mueve en el eje vertical es constante.
- Debe implementar los casos de colisión en el juego para que cuando el jugador toque ciertos elementos pierda la partida (por ejemplo el suelo, o los pilares).
- Usted defina una distancia apropiada horizontal y vertical entre los pilares.
- La altura de los pilares debe ser aleatoria, es decir, a veces puede haber un espacio entre pilares en la parte superior, pero a veces puede estar en el sector inferior.
- Cuando el jugador logre ganar, se debe mostrar una pantalla que indique que el jugador ganó. De la misma forma, debe existir una pantalla de game over.

Puntuación

- Modelos: 0.5 puntos
- Generación aleatoria de la altura de los pilares: 0.5 puntos
- Movimiento del personaje: 1 punto
- Colisión del personaje: 1 punto
- Fondo con textura y que simule el avance del personaje: 0.5 puntos
- Marcador de score: 0.5 puntos
- Pantalla de ganador y game over: 0.5 puntos

Opción C: Pixel-Paint

En esta tarea desarrollaremos un software estilo paint enfocado principalmente en imágenes de baja resolución. De esta forma, será idóneo para Pixel Art.



Su programa se debe ejecutar con la siguiente llamada:

```
python pixel_paint.py 16 pallete.json boo.png
```

- El primer argumento, 16, corresponde al tamaño de la grilla cuadrada de píxeles a utilizar. En este caso generará una grilla de 16×16.
- El segundo argumento, *pallete.json* corresponde a un archivo externo donde se especifica la paleta de colores disponible para pintar. El archivo se encuentra en formato json, por lo que puede ser leído fácilmente.
- El tercer y último argumento corresponde al nombre con el cual se guardará el archivo. Siempre será de extensión *png*.

Ejemplo de archivo *pallete.json*:

```
{
    "transparent" : (0.5, 0.5 ,0.5),
    "pallete" : [ (1, 0, 0), (1, 1, 1), (0, 0, 0)]
}
```

Considere que:

- Los valores dados son solo de ejemplo, su programa debe funcionar para cualquier grilla y nombres de archivos.
- Solo debe utilizar GLFW y OpenGL. No debe utilizar librerías adicionales para manejo de ventanas o botones.
- Debe identificar donde se hace clic con el mouse. Si es un color, se pondrá como color activo; si es un pixel de la imagen, se pintará del color activo.
- Internamente, debe almacenar la información en una matriz cuadrada numpy.
- Al presionar la tecla s o g se deberá guardar la imagen con el nombre especificado como tercer argumento.
- El tamaño de la ventana puede ajustarse automáticamente según el valore de N.
- Asuma que existirá un máximo de 10 colores, para configurar apropiadamente la distribución de la ventana.
- En la barra de colores, el primer color corresponde al color especificado como transparente. Al momento de guardar la imagen, este color debe ser escrito como tal.
- Siéntase libre de utilizar otro diseño para presentar la paleta de colores.

Puntuación

- Tamaño de grilla de píxeles especificada por el usuario: 0.7 puntos
- Paleta de colores especificada por el usuario: 0.8 puntos
- Seleccionar colores y pintar: 2 puntos
- Almacenar imagen, incluyendo transparencia: 1 punto