

Programación Avanzada

IIC2233 2024-1

Hernán Valdivieso - Daniela Concha - Francisca Ibarra - Dante Pinto - Francisca Cattán



Experiencia 3

Interfaces Gráficas II

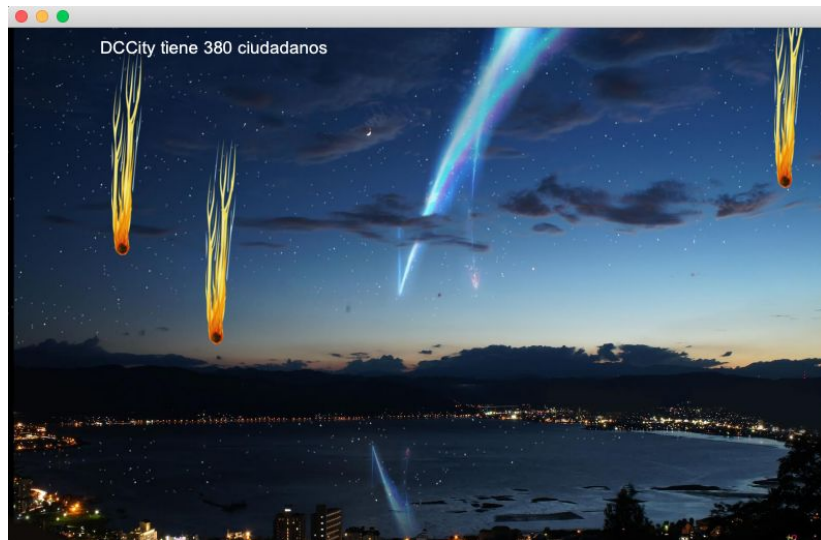
Arreglando un mini juego

Experiencia 3: ¿Qué vamos a hacer?

1. Entenderemos la programación de un mini juego mediante su **Diagrama de Clases**
2. **Arreglaremos el código** que presenta 16 errores que no permiten jugarlo correctamente.

Advertencia: estos errores son algunos típicos que ocurren en la tarea de Interfaz Gráfica 😊

Salvemos DCCity - Contexto



Contexto

Se solicitó diseñar un pequeño juego, pero la persona encargada de hacerlo programó todo sin ejecutar nunca su código y luego, cuando quedaban segundos para hacer *push*, se dió cuenta que no funciona.

Ahora nos pide a nosotros arreglar todo el código.

Salvemos DCCity - Parte 1



Antes de arreglar el código, vamos a entender:

1. Formato y reglas del juego
2. El diagrama de clases que corresponde al boceto original que se esperaba programar.

Salvemos DCCity - Formato y reglas

1. Al principio se solicita seleccionar una dificultad del juego, presionando ENTER u oprimiendo un botón de la ventana se enviará dicha información.
2. Una vez empezado el juego, sonará una linda canción de fondo y aparecerá nuestra ventana de juego. Además, se creará un meteorito desde arriba que comenzará a caer. Estos meteoritos aparecerán cada N segundos donde el N depende de la dificultad seleccionada.
3. Con el *mouse* le puedes disparar al centro del meteorito para destruirlo.
4. Si el meteorito llega a la ciudad (borde de abajo), algunos ciudadanos perderán su hogar y deberán abandonar DCCity.

Salvemos DCCity - Diagrama de clases

Analicemos cómo fue planificado este juego mediante el diagrama de clases.

Junto al código hay un archivo llamado “diagrama_clases.jpeg” que vamos a ver y entender ahora.

¡A ver el diagrama!

Salvemos DCCity - Parte 2

 Ahora vamos a arreglar el código .

Para esto, se definieron “**rondas de errores**”.

- Para cada ronda se indicará cuál es el resultado esperado a llegar tras corregir los errores.
- Luego se describirán 3 o 4 errores en el PPT que intentaremos entender cómo se gatilló y solucionarlo.

Al final del día se publicará la solución donde se detalla cada error en el código y cómo se solucionó.

Salvemos DCCity - Parte 2 (ronda 1)

Resultado esperado tras corregir errores

- Que se abra la ventana de Inicio sin ningún error en el proceso 🎉

Salvemos DCCity - Parte 2 (ronda 1)

1. Intentamos ejecutar main.py y el código se cae de inmediato con un extraño error:

SystemError: <method-wrapper '__init__' of VentanaInicio ...> returned a result with an exception set

2. Solucionado el error 1, el código se vuelve a caer de inmediato con otro error:

TypeError: Juego cannot be converted to PyQt6.QtCore.QObject

3. Solucionado el error 2, el código se vuelve a caer de inmediato con otro error:

TypeError: native Qt signal is not callable

4. Finalmente no hay error, pero nunca se muestra la ventana de inicio 😞. Hay que cerrar el proceso o abrir una nueva terminal para poder volver a ejecutar 🤔

¡A programar/arreglar!  

Salvemos DCCity - Parte 2 (ronda 2)

Resultado esperado tras corregir errores

- Con ENTER o presionando el botón, se envíe la información al *backend*.
- En la terminal debe aparecer, para ambos casos (botón o ENTER), el mensaje:

[Front] Empezar juego

Salvemos DCCity - Parte 2 (ronda 2)

1. Oprimir ENTER no hace nada. Incluso, no se detecta ninguna tecla del teclado cuando se presiona.
2. Solucionado el error 1, cuando se oprime ENTER aparece un *print* en consola de que se envía la información, pero sale un error

TypeError: native Qt signal is not callable

3. Solucionado el error 2, el *backend* nunca da aviso que recibió la información.
4. Si intento oprimir el botón de enviar información de la ventana de Inicio, no se manda la información al *backend*.

¡A programar/arreglar!  

Salvemos DCCity - Parte 2 (ronda 3)

Resultado esperado tras corregir errores

- Aparezca la ventana de Juego y reproduzca música.
- Que no aparezca un error sobre QThread en la terminal.
- Ver un meteorito caer.

Salvemos DCCity - Parte 2 (ronda 3)

1. Se indica que empieza el juego, pero aparece un error en la terminal

AttributeError: 'QMediaPlayer' object has no attribute 'start'

2. Solucionado el tema de 'QMediaPlayer', se sigue sin ver la ventana del juego.
3. Cuando se logra ver la ventana, no aparece nunca un meteorito que notifique que comienza su caída (debería salir un *print* al menos)
4. Solucionado el error 3, intentamos ejecutar y se cae el juego

QThread: Destroyed while thread is still running

¡A programar/arreglar!  

Salvemos DCCity - Parte 2 (ronda 4)

Resultado esperado tras corregir errores

- Aparezcan muchos meteoritos y los podamos destruir cuando presionamos en el centro de ellos.
- Desaparezcan los meteoritos correctamente.
- En resumen: podemos jugar.

Salvemos DCCity - Parte 2 (ronda 4)

Ahora saldrán errores de “jugabilidad” que no cierran el juego, pero no nos permiten jugar correctamente.

1. Hago *click* con el *mouse para destruir* y sale un error;

NameError: name 'x' is not defined. Did you mean: 'self.x'?

2. Solucionado el error 1, aparece un meteorito que nunca desaparece 🤔
3. Solucionado el error 2, el juego solo tiene 1 meteorito, ¿donde están los demás?
4. Solucionado el error 3, por fin veo muchos meteoritos, pero desde el segundo en adelante, si llegan al final... no desaparecen 😱

¡A programar/arreglar!  

Experiencia 3

Interfaces Gráficas II

Cierre de la experiencia

Salvemos DCCity - Resumen de errores

- Olvidar conectar señales o escribirlas mal.
- No hacer show de alguna ventana y quedarse sin nada en la pantalla.
- Usar incorrectamente una señal (no usar `.connect` o no usar `.emit`).
- No guardar en memoria cualquier objeto de PyQt que fue creado en un método.
- No liberar el lock (`QMutex`).
- Usar incorrectamente el `singleShot` de `QTimer`.
- No escribir correctamente los métodos propios de PyQt (`keyPressEvent`, `mousePressEvent`, etc).
- Olvidar heredar de las clases de PyQt.

Salvemos DCCity - Desafíos importante

A continuación se detallan 2 desafíos que puede que sean muy relevantes para la tarea de interfaz gráfica

1. Actualmente escondemos los meteoritos que queremos "eliminar", pero eso lentamente llenará nuestra RAM. Investigar cómo sería el proceso para eliminarlos en *frontend* y *backend*.
2. Cuando cierro el juego mientras hay meteoritos cayendo (ALT+F4, CMD+Q), según el Sistema operativo, puede salir un error en la terminal.

Esto pasa porque destruimos de golpe un QThread que se estaba ejecutando. Implementa un botón de **Cerrar** que primero detenga y destruya los QThread y QTimer antes de cerrar el juego.

Salvemos DCCity - Desafíos extras

- Agregar un contador para saber cuánto tiempo logramos sobrevivir.
- Actualmente no se gana, solo se sobrevive la mayor cantidad de tiempo.... intenta agregar un “Tiempo máximo” que permita decir que ganamos en caso de sobrevivir hasta dicho tiempo.
- Cuando empiece el juego (post seleccionar dificultad), cerrar la ventana de Inicio.
- Implementar una tecla que permita pausar/detener la música mientras jugamos.
- Implementar una tecla que permita pausar y despausar el juego.
- Implementar una tecla que permita aumentar o reducir la velocidad de caída del meteorito.

Programación Avanzada

IIC2233 2024-1

Hernán Valdivieso - Daniela Concha -

- Dante Pinto - Francisca Cattán

