

## Ejercicio 2

### Problema 1

Partiendo del código de ejemplo

[https://github.com/dantros/beginning\\_cpp/tree/main/basics/04\\_static\\_lib/source](https://github.com/dantros/beginning_cpp/tree/main/basics/04_static_lib/source)

- a) Defina la clase Path que internamente almacene un `std::vector` de puntos. Esta clase modela un camino que se intentará recorrer. La clase debe tener las funciones miembro:
  - a. `Path& addDestination(const Point& point);` // agregando un nuevo vértice
  - b. `Path& reverse();` // change the direction of the path to the opposite way.
  - c. `Float length() const;` // calcula el largo total del camino (suma total del largo de cada segmento contiguo).
  - d. `Point travel(float t) const;` // t es un parámetro entre 0 y 1, se utiliza para obtener el punto P que se encuentra a razón t respecto de la posición inicial avanzando por el camino (Path)
- b) Ejercite su static library imprimiendo la secuencia de puntos que se obtienen al caminar por un path/camino de unos 4 o más puntos.

### Problema 2

- a) Implemente una static library distinta para manipular Texturas. Cada textura está definida por una matriz 2D de colores. Para esto, utilizaremos internamente un `std::vector` definida para el tipo template Color.
- b) Una versión de Color, digamos ColorRGB, puede ser definido con valores de un tipo template para r,g,b. Ya que estamos aquí... ¿Por qué querríamos que Color no siempre fuera definido en base a r, g y b?
- c) La clase TextureT debe construirse para un alto y ancho dado con un color por defecto dado en el constructor.
- d) Redefina el operador () para asignar en la posición i,j el color dado como argumento.
- e) El mismo operador, en su versión constante, puede ser utilizado para consultar el color ya almacenado.
- f) Revisar: <https://stackoverflow.com/questions/2351148/explicit-template-instantiation-when-is-it-used>
- g) Utilice explicit instantiation para definir:
  - a. `using TextureB = TextureT<ColorRGB<std::byte>>;`
  - b. `using TextureI = TextureT<ColorRGB<int>>;`
  - c. `using TextureF = TextureT<ColorRGB<float>>;`

### Problema 3

Defina un struct GameObject que almacene posición y una textura de 2x2 pixeles. En un ejecutable, establezca un loop donde se mueva este GameObject a lo largo de un path (tal como en 1b), en cada paso, modifique el color de un pixel de la textura cambiando sus componentes rgb.