COMANDI DI SFML:N.B., useful for project

Classi :

* THREAD -> DONNA CHE SA FARE IL MULTITASKING (sf::Thread ): un “vettore di funzioni ” che avvengono contemporaneamente

1. sf::Thread::Thread(F function) non runna, ha bisogno del metodo launch

|  |
| --- |
|  |

1. sf::Thread::Thread ( F function, A argument)
2. sf::Thread::Thread ( void(C::\*)() function, C object )

N.B. Negli argomenti dei costruttori metti il koala che si aggrappa all’albero(referenza )

- Member function:

class MyClass

{

public:

void func() { }

};

MyClass object; //creo oggetto della classe

sf::Thread thread(&MyClass::func, &object);// faccio partire il thread

1. void sf::Thread::launch()-> avvia il multitasking
2. void sf::Trhead::terminate()-> blocca il multitasking, non importa se ha finito l’operazione
3. void sf::Thread::wait()-> è la pazienza -> aspetta finchè non finisce

* WINDOW-> FINESTRA (sf::Window)

<https://www.sfml-dev.org/documentation/2.5.1/classsf_1_1Window.php>

1. Costruttore
2. Creare (in aiuto al costruttore)
3. Chiudere(overloading)
4. E’ aperto(overloading)
5. Impostazioni dell’open graphic library
6. Dimensioni della rendering finestra -> solo quadro no cornice
7. PollEvent-> interrogatorio agli eventi;
8. Geometria varia(posizioni, dimensioni tramite vettori bidimensionali); nei vettori 2d le componenti sono attributi pubblici

* EVENT
* Keyboard sf::Keyboard
* Mouse sf::Mouse
* Inserimento del testo

**PER I BOTTONI**

* FONT -> CARICAMENTO E MANIPOLAZIONE DEI CARATTERI

<https://www.sfmldev.org/documentation/2.5.1/classsf_1_1Font.php#ab020052ef4e01f6c749a85571c0f3fd1>

I Seguenti sono tutti getters, tranne il 5

1. Recupera la rappresentazione (simbolo) del carattere ;

[ const Glyph& sf::Font::getGlyph ( Uint32[è un tipo di numeri interi con un determinato range] codePoint [è un numero intero corrispondente un tipo di carattere], unsigned int characterSize, bool bold[una rappresentazione grafica specifica ed originale; rappresentazione artistica ], float outlineThickness = 0 ) const]

1. I caratteri voglio un loro spazio vitale e lo decidi tu ; spazio sta caratteri [Kerning]
2. Anche le linee di testo voglio lo spazio vitale [LineSpacing/UnderlinePosition]
3. Assorbimento del font da file[bool sf::Font::loadFromFile(const std::string & filename)], da file in memoria [bool sf::Font::loadFromMemory( const void \* data, std::size\_t sizeInBytes )], da input (?) [bool sf::Font::loadFromStream( InputStream & stream ) ]
4. Allineamenti [Font& sf::Font::operator=(const Font & right )]

=>TEXT-> “disegno i caratteri”(è più leggero di un oggeto Font, ma lavorano in società)

<https://www.sfml-dev.org/documentation/2.5.1/classsf_1_1Text.php>

1. Tutto ciò che concerne il disegno del carattere, posizione e orientazione -> setters, getters

N.B. Text prende anche lui il font con un getter [getFont], ritorna un puntatore, non fa la copia come fa Font