

Simple Drawing FrameWork

Y.Nakaue

2022 年 3 月 11 日

1 プログラムの開始と終了

1.1 ライブラリ機能の初期化

```
void init()
```

引数 無し

返回值 無し

1.2 ライブラリ機能の終了処理

```
void quit()
```

引数 無し

返回值 無し

2 メインループ

2.1 描画内容の更新

```
bool System::update
```

引数 無し

返回值 メインループの更新可否

描画内容を最新の状態に更新する。常に true を返す。

3 描画のための設定

3.1 ウィンドウの作成

```
int32_t openWindow(uint32_t width, uint32_t height)
```

引数 width: 横幅, height: 高さ

返回值 作成したウィンドウ ID

描画を行うためのウィンドウを作成し、画面前面に表示する。引数には作成するウィンドウの横幅と高さを指定する。1つのプログラムの中で複数のウィンドウを作成することが可能で、返回值として作成したウィンドウに割り当てられたウィンドウ ID を返す。ウィンドウ ID は、作成した順に 0 からの連番で整数値が返される。

3.2 ウィンドウを閉じる

```
void closeWindow(int32_t win_id)
```

引数 win_id: 閉じるウィンドウの ID

返回值 無し

4 主な描画関数

4.1 背景色を変更

```
void setBackground(Color color, int32_t win_id = 0)
```

引数 color: 背景色, win_id: 背景色を設定するウィンドウの ID

返回值 無し

背景色を引数で指定した色に設定する。新たな背景色は、この関数によって設定した次のフレーム (System::update() が呼び出されたタイミング) で反映される。

5 イベントに関する機能

以下の機能は System::update() が呼び出されたタイミングで状態が更新されているため、少し古い情報が得られる場合がある。

5.1 マウス入力

5.1.1 マウスボタン入力状態の取得（押されているかを取得）

`bool pressed(int8_t button)`

引数 `button`: マウスボタン（マクロ）

返り値 指定したマウスボタンが押されているか

入力状態を得たいマウスボタンの指定には、マクロによるビットマスクを使用する。ここで使用するマクロは以下の表 1 のように定義されている。また、これらをビット OR で結合して引数に指定することで、複数のボタンが同時に入力されているかを得ることもできる。

表 1 マウスボタンを表すマクロ名

ボタン	マクロ名
マウス左ボタン	LEFT
マウス中ボタン	MIDDLE
マウス右ボタン	RIGHT

ソースコード 1 使用例

```
1 // マウス左ボタンが押されているかを取得
2 sdfw::Mouse::pressed(LEFT);
3
4 // マウス左・右ボタンが両方押されているかを取得
5 sdfw::Mouse::pressed(LEFT | RIGHT);
```