

```
1 //*****
2 // File name:  SDLApp.h
3 // Author:     Chadd Williams
4 // Date:       1/31/2017
5 // Class:      CS 485
6 // Assignment: 01_Assignment_Shapes
7 // Purpose:    Provide a wrapper around SDL and a simple game loop
8 //*****
9 #pragma once
10 #include "SDL.h"
11 #include "SDL2_gfxPrimitives.h"
12 #include "SDL_ttf.h" // true type font
13 #include <string>
14 #include <vector>
15 #include "Color.h"
16 #include <map>
17 #include "ISDLWidgetTextable.h"
18 #include "ISDLWidgetClickable.h"
19
20 class SDLTextWidget;
21 class SDLSprite;
22
23 class SDLApp
24 {
25 public:
26     SDLApp (unsigned int framesPerSecond = 60);
27     SDLApp (const SDLApp&) = delete;
28
29     virtual ~SDLApp ();
30
31     SDLApp& operator=(const SDLApp&) = delete;
32
33     bool createWindow (std::string windowName,
34         int windowXPos = 100,
35         int windowYPos = 100,
36         int windowSizeX = 640,
37         int windowSizeY = 640);
38     void setWindowBackgroundColor (Color cRgba);
39
40     void cleanup ();
41
42     void setColor (Color cRgba);
43     Color getColor () const;
44     void drawLine (int x1, int y1, int x2, int y2);
45     void drawCircle (int x, int y, int radius);
46     void drawBox (int x, int y, int x2, int y2);
47
48     void displayText (int x, int y, std::string msg, Color cRgba);
49
50     int loadFont (std::string fontname, int size);
51     bool displayText (int x, int y, std::string msg, Color cRgba, int font);
52     bool displayTextClickable (int x, int y, std::string msg, Color cRgba,
```

```
53     int font, int &rW, int &rH);
54
55     bool loadSprite (std::string name, int &w, int &h);
56     void displaySprite (const SDLSprite &rcSprite);
57
58     void gameLoop ();
59     unsigned int getFramesPerSecond ();
60
61
62     void enableTextInput ();
63     void disableTextInput ();
64
65     void registerTextWidget (ISDLWidgetTextable *pcWidget);
66     void registerClickableWidget (ISDLWidgetClickable *pcClickable);
67
68     // subclasses should implement each of the
69     // following methods as necessary to customize
70     // their App.
71     virtual void initGame ();
72     virtual void update ();
73     virtual void handleEvent (SDL_Event event);
74
75     virtual void render () = 0;
76
77 private:
78
79     SDL_Window *mpWindow = nullptr;
80     SDL_Renderer *mpRenderer = nullptr;
81
82     // current color
83     Color mcCurrentColor;
84
85     // background color
86     Color mcBackgroundColor;
87
88     std::vector<TTF_Font *> mcFonts;
89
90     std::vector<ISDLWidgetClickable *> mcClickWidgetVector;
91     std::vector<ISDLWidgetTextable *> mcTextWidgetVector;
92     ISDLWidgetTextable *mpFocusTextWidget = nullptr;
93
94     bool mbTextInputActive = false;
95     bool mbEnableTextInput = false;
96
97     std::map<std::string, SDL_Texture*> mcSpriteMap;
98     void handleTextEvent (SDL_Event event);
99
100     unsigned int mFramesPerSecond;
101 };
```

```
1 //*****
2 // File name: Color.h
3 // Author: Chadd Williams
4 // Date: 3/30/2017
5 // Class: CS 485
6 // Assignment: SDLApp2
7 // Purpose: Provider wrapper for SDL
8 //*****
9 #pragma once
10
11 #include "SDL.h"
12 #include <iostream>
13
14 class Color
15 {
16 public:
17
18     Color ();
19     Color (Uint8 r, Uint8 g, Uint8 b, Uint8 a=0);
20
21     Uint8 getR () const;
22     Uint8 getG () const;
23     Uint8 getB () const;
24     Uint8 getA () const;
25
26     void setOpacity (float opacity);
27
28     friend std::istream& operator >> (std::istream &in, Color &rcColor);
29
30 private:
31     Uint8 mR, mG, mB, mA;
32 };
```

```
1 //*****
2 // File name: ISDLWidget.h
3 // Author: Chadd Williams
4 // Date: 3/30/2017
5 // Class: CS 485
6 // Assignment: SDLApp2
7 // Purpose: Provider wrapper for SDL
8 //*****
9 #pragma once
10 #include <string>
11 #include "SDLApp.h"
12
13 class ISDLWidget
14 {
15 public:
16
17     virtual ~ISDLWidget () = default;
18
19     virtual void draw (SDLApp &rcApp) = 0;
20
21     virtual bool isVisible () { return mbVisible; };
22
23     virtual void setVisible (bool bVisible) { mbVisible = bVisible; };
24
25 private:
26     bool mbVisible = true;
27 };
28
```

```
1 //*****
2 // File name: ISDLWidgetClickable.h
3 // Author: Chadd Williams
4 // Date: 3/30/2017
5 // Class: CS 485
6 // Assignment: SDLApp2
7 // Purpose: Provider wrapper for SDL
8 //*****
9 #pragma once
10 #include "SDLApp.h"
11 #include <functional>
12
13 class ISDLWidgetClickable
14 {
15 public:
16
17 // must return true or false to determine
18 // if the widget was clicked
19 // may optionally call clickEventHandler()
20 virtual bool clicked (int x, int y) = 0;
21
22 virtual void registerClickEventHandler (
23     std::function<void (void)> cFunction)
24 { mcClickHandler = cFunction;}
25
26 virtual void clickEventHandler()
27 {
28     if (nullptr!=mcClickHandler)
29     {
30         mcClickHandler ();
31     }
32 }
33
34 private:
35     std::function<void (void)>
36     mcClickHandler = nullptr;
37 };
```

```
1 //*****
2 // File name: ISDLWidgetTextable.h
3 // Author: Chadd Williams
4 // Date: 3/30/2017
5 // Class: CS 485
6 // Assignment: SDLApp2
7 // Purpose: Provider wrapper for SDL
8 //*****
9 #pragma once
10 #include <string>
11 #include "SDLApp.h"
12 #include <functional>
13
14 #include "ISDLWidgetClickable.h"
15
16 class ISDLWidgetTextable : public ISDLWidgetClickable
17 {
18 public:
19
20 virtual ~ISDLWidgetTextable () = default;
21 virtual void setData (std::string cData) { mData = cData;}
22
23 virtual std::string getData () const {return mData;}
24 virtual void setLabel (std::string cLab) { mLabel = cLab; }
25 virtual std::string getLabel () const { return mLabel; }
26 virtual int getLength () const = 0;
27
28 virtual void setEditable (bool bEditable) { mbEditable = bEditable; };
29 virtual bool isEditable () const { return mbEditable; };
30 virtual void setFocus (bool bFocus) { mbHasFocus = bFocus; };
31 virtual bool hasFocus () const {return mbHasFocus;};
32
33 virtual void stateChangeEvent () {
34     if (nullptr != mcStateChangeHandler)
35     {
36         mcStateChangeHandler ();
37     }
38 };
39
40 virtual void registerStateChangeEventHandler (
41     std::function<void (void)> cFunction)
42 {
43     mcStateChangeHandler = cFunction;
44 };
45
46
47 private:
48     std::string mData;
49     std::string mLabel;
50
51     bool mbEditable = true;
52     bool mbHasFocus = false;
```

---

```
53
54     std::function<void (void)>
55         mcStateChangeHandler = nullptr;
56
57
58     };
```

```
1 //*****
2 // File name:  SDLSprite.h
3 // Author:    Chadd Williams
4 // Date:      3/30/2017
5 // Class:     CS 485
6 // Assignment: SDLApp2
7 // Purpose:   Provider wrapper for SDL
8 //*****
9 #pragma once
10
11 #include "SDLApp.h"
12 #include <functional>
13 #include "ISDLWidgetClickable.h"
14 #include "ISDLWidget.h"
15
16 class SDLSprite : public ISDLWidgetClickable, public ISDLWidget
17 {
18 public:
19
20     SDLSprite () = default;
21     SDLSprite (SDLApp *pcApp, std::string spriteName, int x ,int y);
22
23     bool loadSprite (SDLApp *pcApp, std::string spriteName, int x, int y);
24
25     int getX () const;
26     int getY () const;
27     int getW () const;
28     int getH () const;
29     std::string getName () const;
30
31     void draw (SDLApp &rcApp) ;
32
33     bool clicked (int x, int y);
34
35     void setScale (double scale);
36
37     void setRotation (double rotation);
38     double getRotation () const;
39
40 private:
41
42     std::string mSpriteName;
43     int mX, mY;
44     int mH, mW;
45     double mScale = 1.0;
46     double mRotation = 0.0;
47
48 };
49
```



```
1 //*****
2 // File name:  SDLTextWidget.h
3 // Author:    Chadd Williams
4 // Date:      3/30/2017
5 // Class:     CS 485
6 // Assignment: SDLApp2
7 // Purpose:   Provider wrapper for SDL
8 //*****
9 #pragma once
10 #include "ISDLWidget.h"
11 #include <functional>
12 #include "ISDLWidget.h"
13 #include "ISDLWidgetClickable.h"
14 #include "ISDLWidgetTexttable.h"
15
16 class SDLTextWidget : public ISDLWidget, public ISDLWidgetTexttable
17 {
18
19 public:
20
21     SDLTextWidget (std::string label, std::string data, int x, int y,
22                   int font, Color cColor);
23
24     virtual ~SDLTextWidget () = default;
25
26     void draw (SDLApp &rcApp) ;
27
28     int getLength () const;
29
30     bool clicked (int x, int y);
31
32     int getWidth () const;
33     int getHeight () const;
34
35     void setColor (Color cColor);
36
37
38 private:
39
40     int mX;
41     int mY;
42     int mH=0;
43     int mW=0;
44     int mFont;
45     Color mcColor;
46
47 };
```