



Outline

- Quick review
 - Resources
 - Scenes
 - Objects & Sprites
 - **Objects & Controls**
 - Template & Code structure
 - Goal & Grading Policy
- 

Objects & Controls

- A simple button requires too much code.

```
void on_mouse_down(int btn, int x, int y) {  
    if (btn == 1 && pnt_in_rect(x, y, btnX, btnY, btnW, btnH)) {  
        // Button clicked.  
    }  
}  
  
void game_draw() {  
    if (pnt_in_rect(mouse_x, mouse_y, btnX, btnY, btnW, btnH))  
        al_draw_bitmap(img_btn_in, btnX, btnY, btnW, btnH);  
    else  
        al_draw_bitmap(img_btn_out, btnX, btnY, btnW, btnH);  
}
```

Objects & Controls

- We can ignore the drawing and mouse-in detection. For buttons, we only want to know when it is clicked. Declaring a variable just for the button is also unnecessary: higher abstraction.



```
void SceneA::BtnOnClick() { // Button clicked. }  
void SceneA::Initialize() {  
    ImageButton* btn = new ImageButton("img_out.png", "img_in.png", 0, 0);  
    btn->SetOnClickCallback(std::bind(&SceneA::BtnOnClick, this));  
    AddNewControlObject(btn);  
}
```

The background is a light gray gradient. It is decorated with numerous realistic water droplets of various sizes, some clustered in the top-left and bottom-right corners. In the upper center, there is a faint, circular logo or watermark that appears to contain a stylized 'E' or a similar symbol.

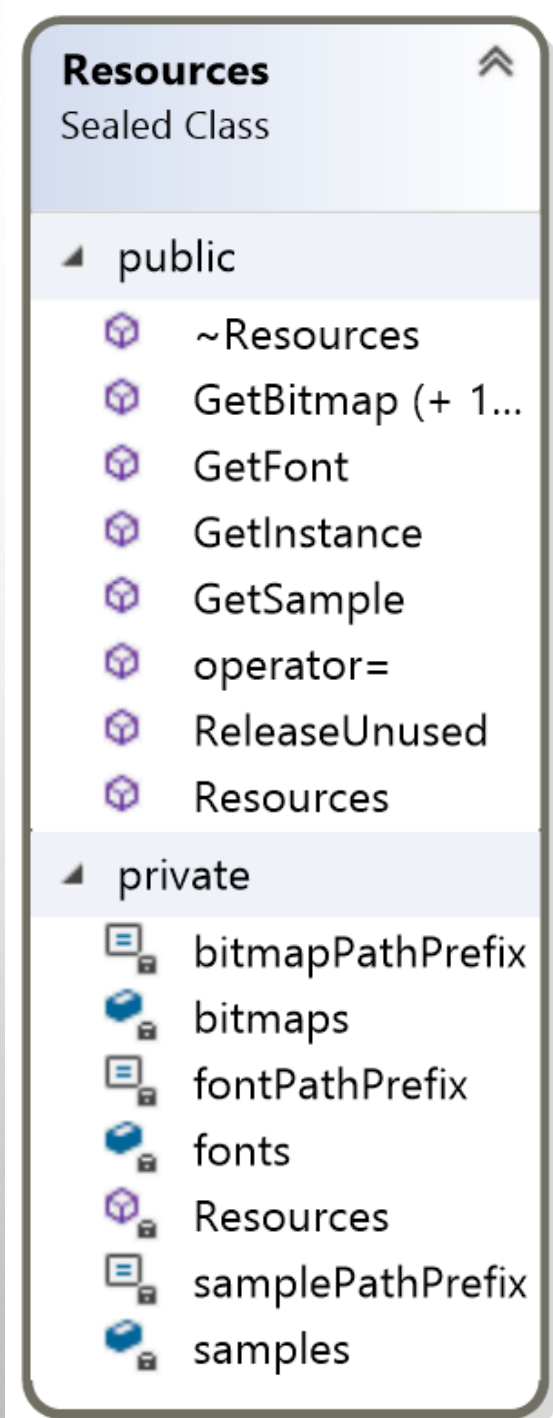
Engine code

Tower Defense

Template: Resources

Engine::Resources

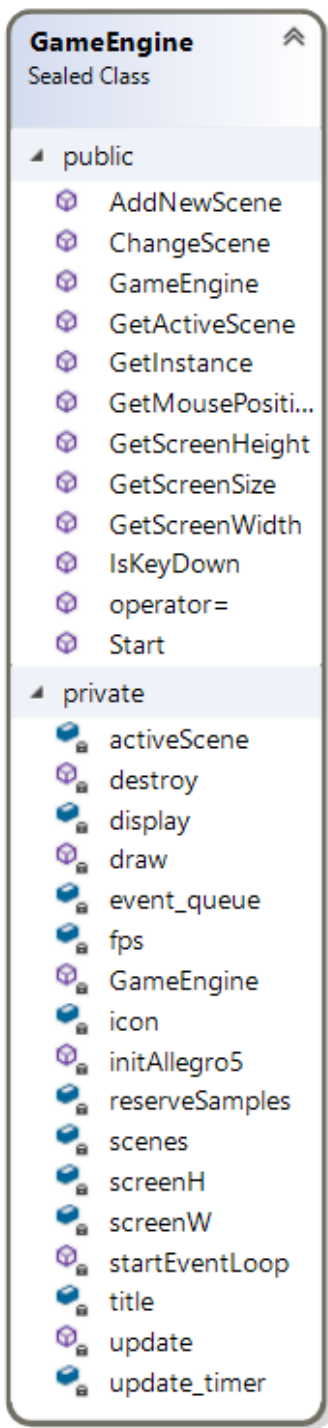
- Abstracts all resources loading and destroy.
- Resources can be retrieved from this class directly.



Template: Game Engine

`Engine::GameEngine`

- Abstracts the entire message loop
- Manages current scene and scene changes.



Template: IScene, Group

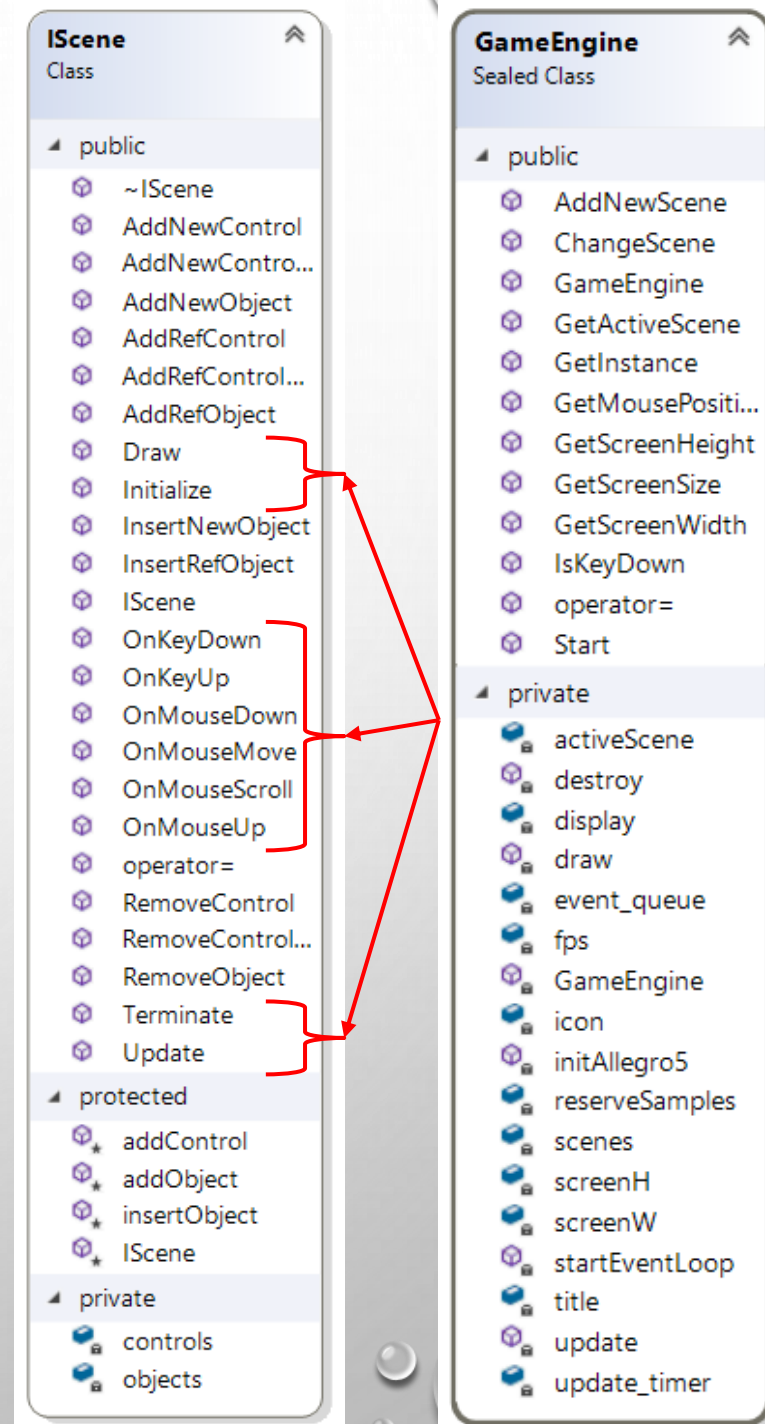
Engine::IScene

- Encapsulates a scene, must be inherited and customized.

Engine::Group

- Draw and update everything for you.

Note: We combined Group and IScene in this diagram



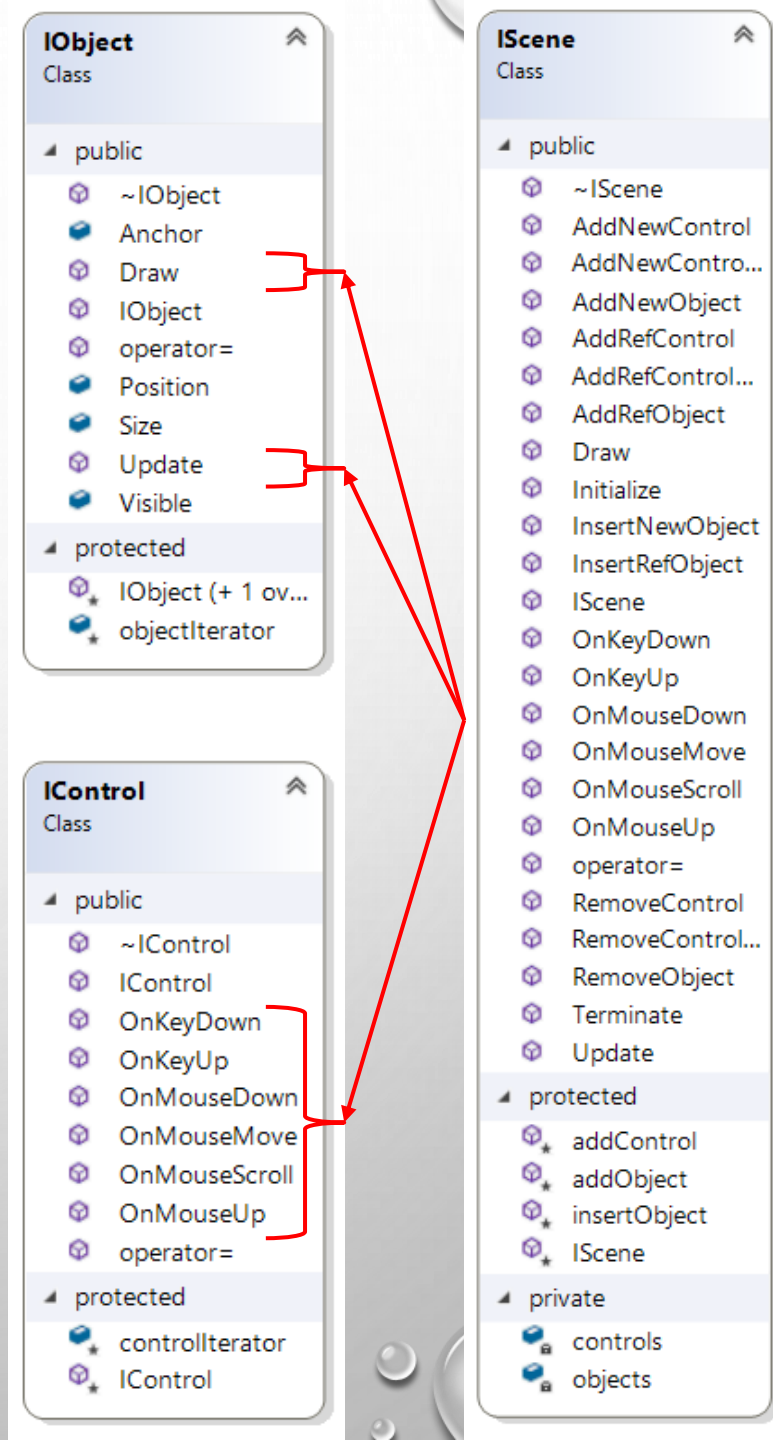
Template: IObject, IControl

Engine::IObject

- The base class of everything that can be drawn.

Engine::IControl

- The base class of everything that can receive events.



Template: Image, Sprite

Engine::Image :

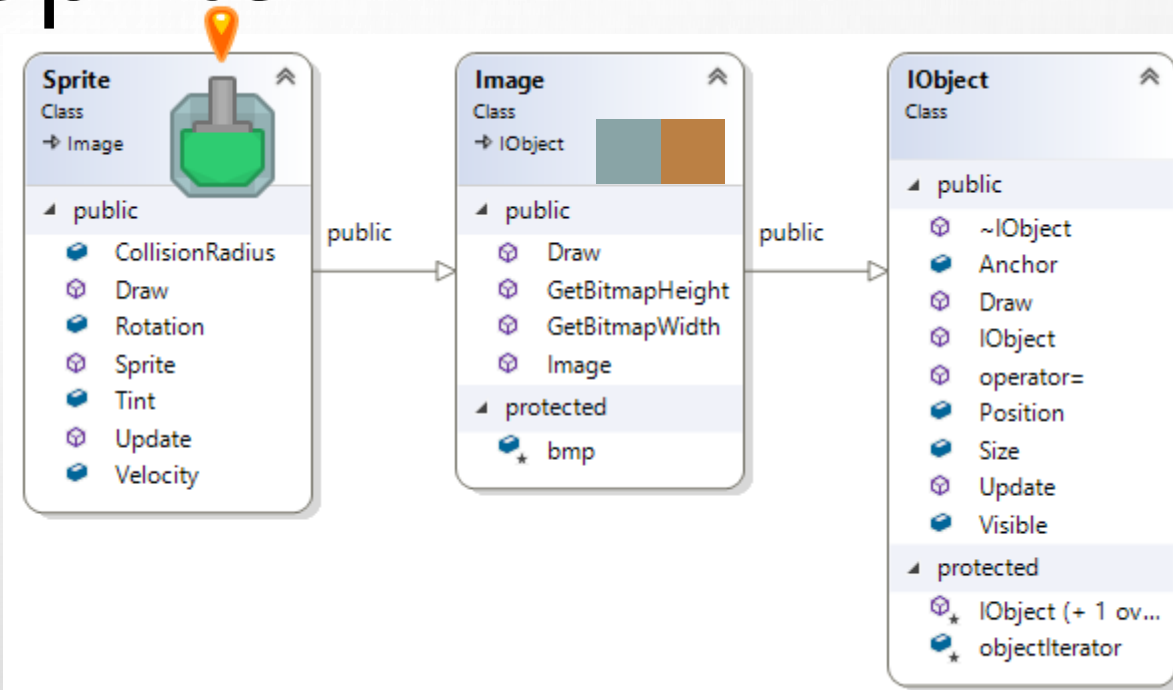
public Engine::IObject

- A simple static image object.

Engine::Sprite :

public Engine::Image

- Supports rotation, velocity, tint, and collision radius.



Template: Label, ImageButton

Engine::Label :

public Engine::IObject

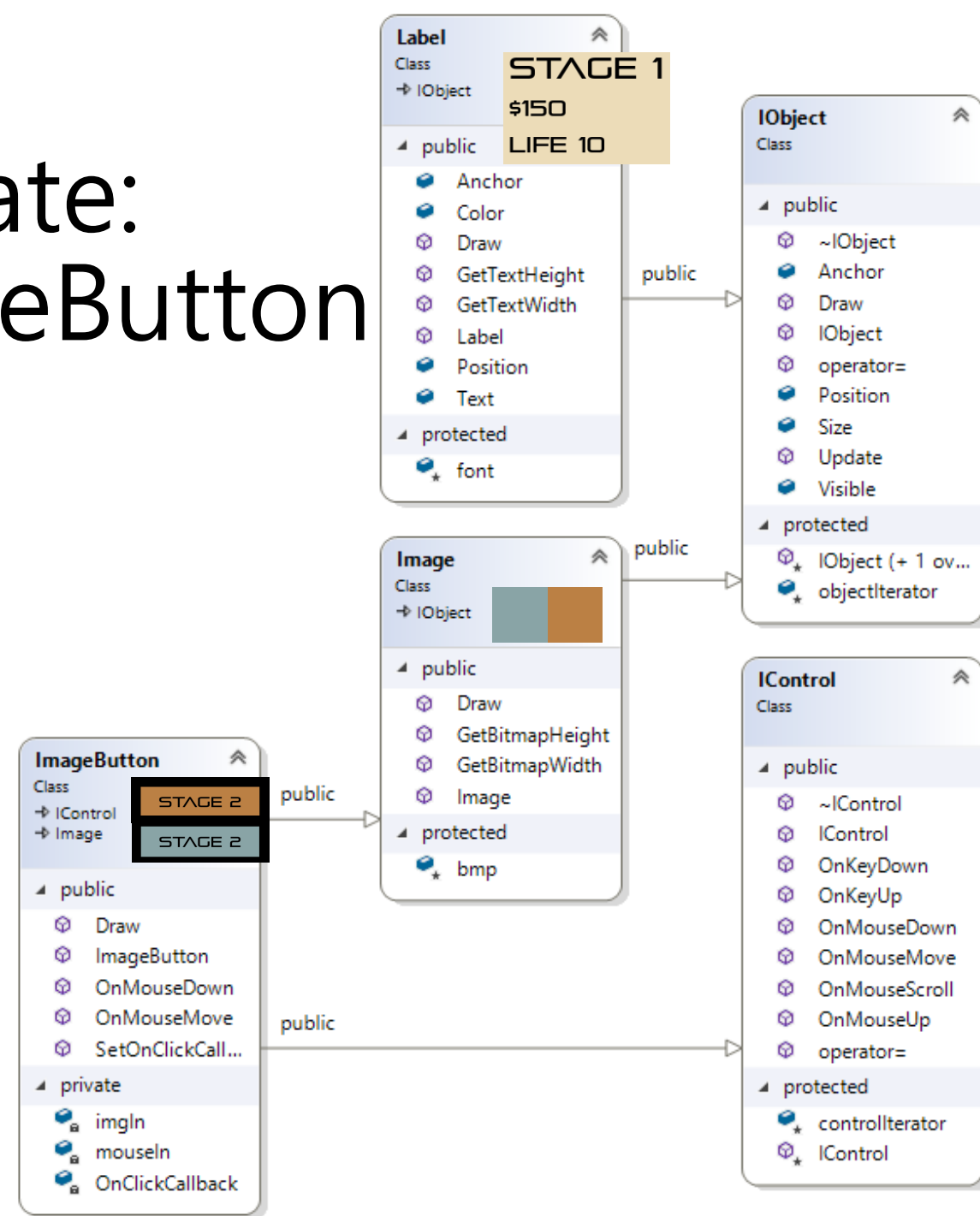
- A simple static text object.

Engine::ImageButton :

public Engine::IObject

public Engine::IControl

- A clickable button, changes image when mouse move.



Engine Diagram (Minimized)

(OnMouseMove,
OnMouseDown, ...)

(Update, Draw)

