

# raylib-apl

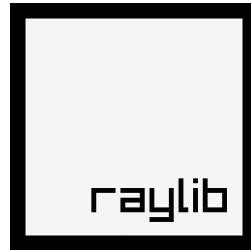
A Simple Cross-Platform Library to  
develop applications



*Presented by*  
Brian Ellingsgaard

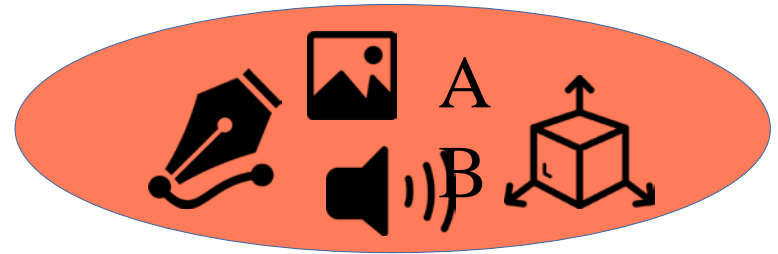
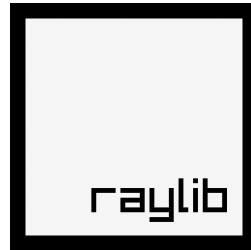
# raylib

A Simple Cross-Platform Library to  
develop applications



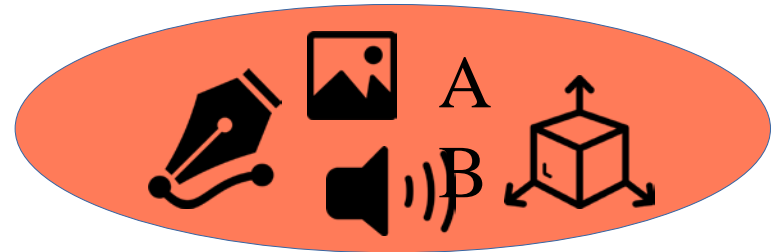
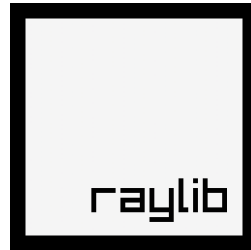
# raylib

A Simple Cross-Platform Library to  
develop applications



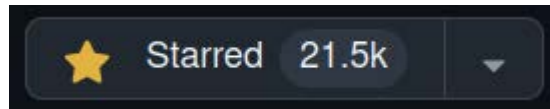
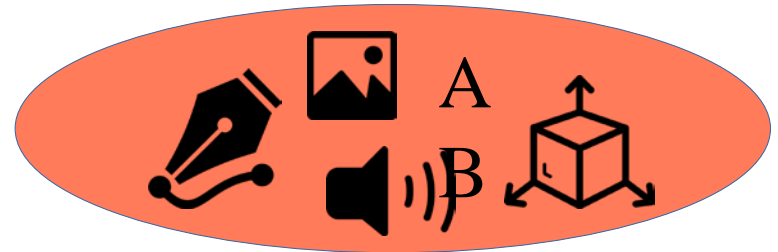
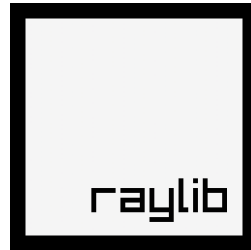
# raylib

A Simple Cross-Platform Library to  
develop applications

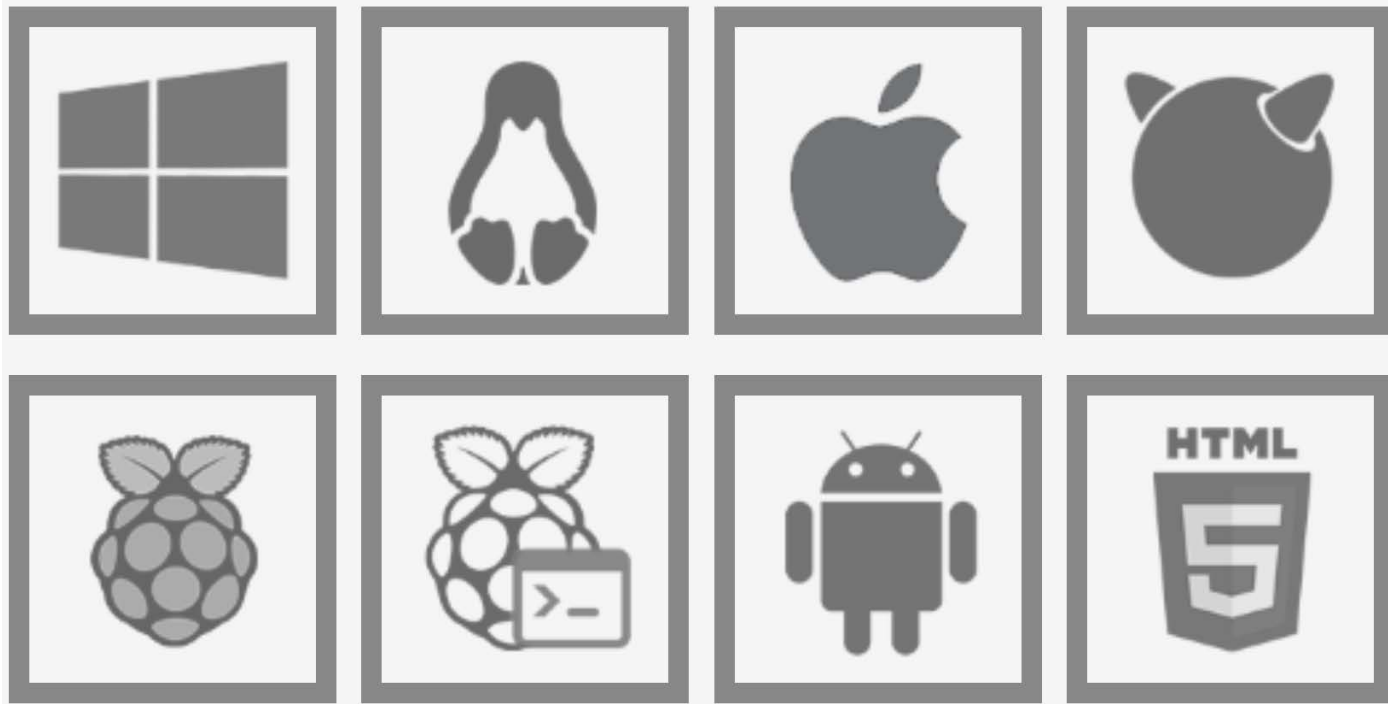


# raylib

A Simple Cross-Platform Library to  
develop applications



# raylib platforms



# My passion



# My passion

answer  $\leftarrow$  6?6

Ask  $\leftarrow$  {+/answer= $\omega$ }



# My passion

answer  $\leftarrow$  6?6

Ask  $\leftarrow$  {+/answer= $\omega$ }

Ask 1 2 2 2 2

1

# My passion

answer ← 6?6

Ask ← {+/answer=ω}

Ask 1 2 2 2 2 2

1

Ask 3 4 4 4 4 4

2

# My passion

answer  $\leftarrow$  6?6

Ask  $\leftarrow$  {+/answer= $\omega$ }

Ask 1 2 2 2 2 2

1

Ask 3 4 4 4 4 4

2

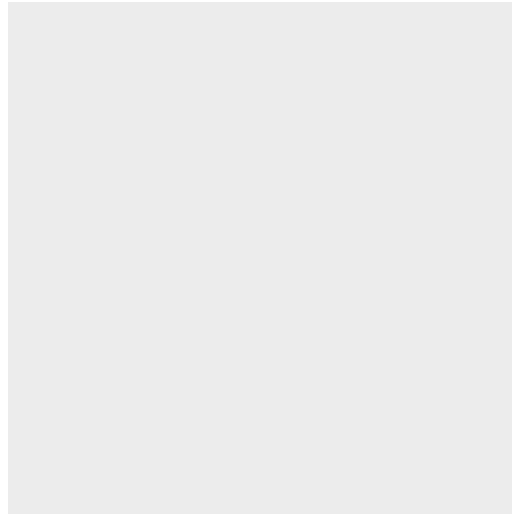
Ask 3 4 5 2 1 6

11<sup>6</sup>



□ WC

□ WC



□ WC

black blue ← 0 255

stripes ← 16 / 16/16 16 p black blue

'f' □ WC 'Form'

'b' □ WC 'Bitmap' ('CBits' stripes)

'f.i' □ WC 'Image'(0 0)('Picture' 'b')

{b.CBits ⊕ ~ ← 1 ◊ □ DL ÷ 60} ~ 200

□ WC

black blue ← 0 255

stripes ← 16 / 16/16 16 ρ black blue

'f' □ WC 'Form'

'b' □ WC 'Bitmap' ('CBits' stripes)

'f.i' □ WC 'Image'(0 0)('Picture' 'b')

{b.CBits ⊕ ~ ← 1 ◊ □ DL ÷ 60} " l200

□ WC

black blue ← 0 255

stripes ← 16 / 16/16 16 ρ black blue

'f' □ WC 'Form'

'b' □ WC 'Bitmap' ('CBits' stripes)

'f.i' □ WC 'Image'(0 0)('Picture' 'b')

{b.CBits ⊕ ~ ← 1 ⋄ □ DL ÷ 60} ⌈ 200



□ WC

black blue ← 0 255

stripes ← 16 / 16/16 16 ρ black blue

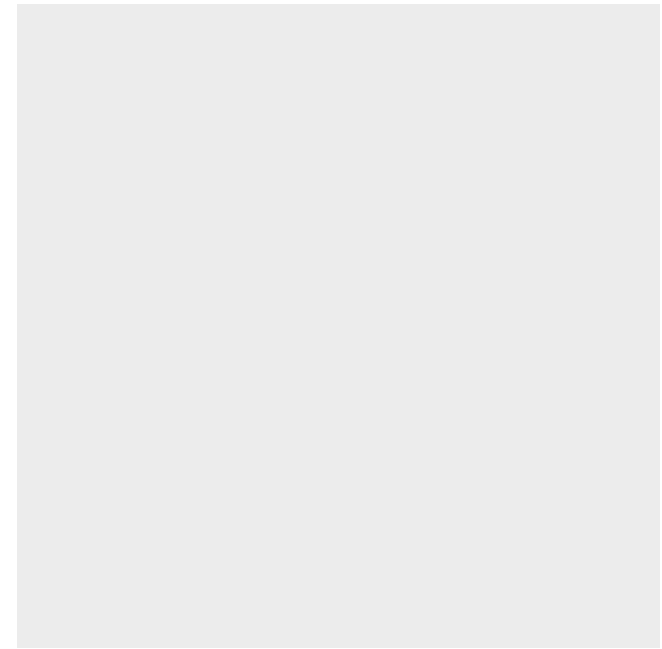
'f' □ WC 'Form'

'b' □ WC 'Bitmap' ('CBits' stripes)

'f.i' □ WC 'Image'(0 0)('Picture' 'b')

{b.CBits ⊕ ~ ← 1 ♦ □ DL ÷ 60} ι 200

□ WC



{b.CBits  $\phi \sim \leftarrow 1 \diamond \square DL \div 60\}$  200

# raylib-apl

Now showing stripes video made with  
raylib-apl

Screen  
2

```
rl ← 0 □ Fix'path/to/raylib.apln'  
rl.Init 0
```

```
blue ← 8 ↑ 0 0 255 255
```

```
black ← 8 ↑ 0 0 0 255
```

```
poses ← 32 × 1 - ~ 11
```

```
rl.InitWindow 800 800 'Hello!!!'
```

```
rl ← 0 □ Fix'path/to/raylib.apln'  
rl.Init 0
```

```
blue ← 8 ↑ 0 0 255 255
```

```
black ← 8 ↑ 0 0 0 255
```

```
poses ← 32 × 1 - ~ 11
```

```
rl.InitWindow 800 800 'Hello!!!'
```

⌚ For every frame till user closes the window

:While ~rl.WindowShouldClose

poses +← 1

poses |~← 32×9

rl.BeginDrawing ⌚ Drawing a new frame

⌚ We want black background

rl.ClearBackground ⊂black

rl.BeginScissorMode 0 0 256 256 ⌚ Only draw in area

rl.DrawRectangle" (poses-32), " ⊂0 16 256 blue

rl.EndScissorMode

rl.DrawFPS 260 0

rl.EndDrawing

:EndWhile

rl.CloseWindow

`rl.SetTargetFPS 314159`

# raylib-apl

Now showing FAST stripes video made  
with raylib-apl

Screen  
3



# raylib-apl

Now showing duck with raylib-apl

Screen  
4

```
rl.InitWindow 800 800 'Hello!!!'
```

```
rl.DisableCursor
```

```
proj ← rl.CameraProjection.CAMERA_PERSPECTIVE
```

```
camera ← (50 50 50) (0 10 0) (0 1 0) 45 proj 0 0 0 0
```

⦿ Define the camera to look into our 3d world

⦿ Camera position

⦿ Camera looking at point

⦿ Camera up vector (rotation towards target)

⦿ Camera field-of-view Y

⦿ Camera mode type

⦿ Duck model

```
model _ ← rl.LoadModelRetPtr Ⓟ 'OBJ/RubberDuck_LOD0.obj'
```

⌚ For every frame till user closes the window

:While ~rl.WindowShouldClose

⌚ Look for keystrokes from user, move the camera

camera ← rl.UpdateCamera(, camera)

Drawing...

:EndWhile

⌚ Unloading model, freeing up memory

rl.UnloadModel, camera

⌚ Application finished, window closed

rl.CloseWindow

⌚ draw the frame with white background

rl.BeginDrawing

rl.ClearBackground ← 8 ↑ rl.color.white

rl.DrawFPS 40 40 ⌚ Draws FPS at x=40 y=40

⌚ Draw 3d stuff relative to the camera

rl.BeginMode3D ← camera

⌚ draws grid

rl.DrawGrid 20 10

⌚ Draw the duck model at position

position ← 0 0 0 ♦ scale ← 1

rl.DrawModel model position scale (8 ↑ rl.color.white)

rl.EndMode3D ⌚ Stop drawing 3d

⌚ wait till next frame is needed, then switch to our drawing

rl.EndDrawing

# raylib-apl

Now again showing duck  
with raylib-apl

Screen  
4

# VR showcase video

Screen  
5

# raylib showcase video

Screen  
6

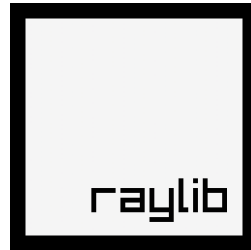
# Extra bits

raylib-apl adapts to breaking changes



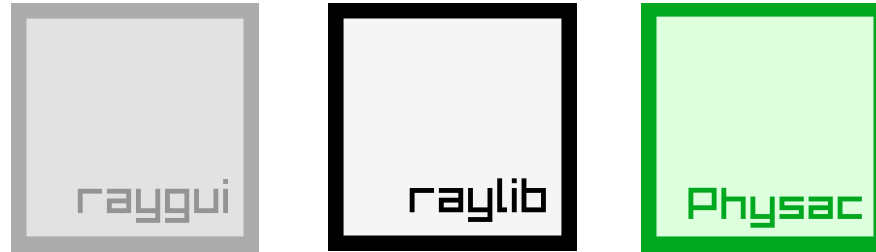
# Extra bits

raylib-apl adapts to breaking changes



# Extra bits

raylib-apl adapts to breaking changes



# When raylib-apl, and when not.

## Good reasons

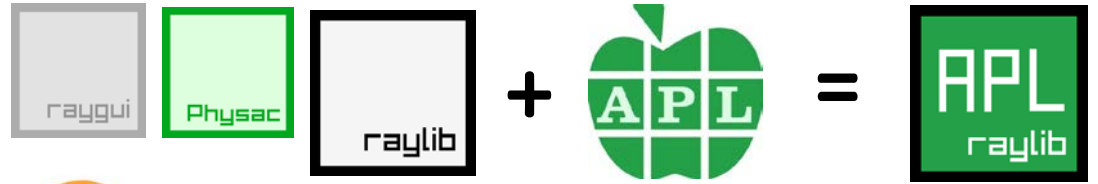
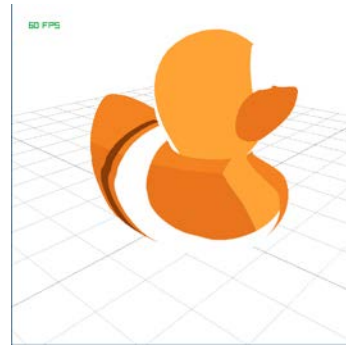
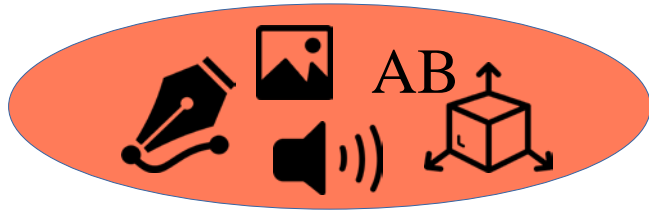
- Game
- 3D
- Control

## Bad reasons

- OOP
- GUI
- In production now

# Recap

A Simple Cross-Platform Library to develop applications



# Questions?

