

Blender

Jeroen Bakker
Blender Institute
jeroen@blender.org

Wat is Blender?

Wie gebruikt Blender?

Hoe is Blender georganiseerd?

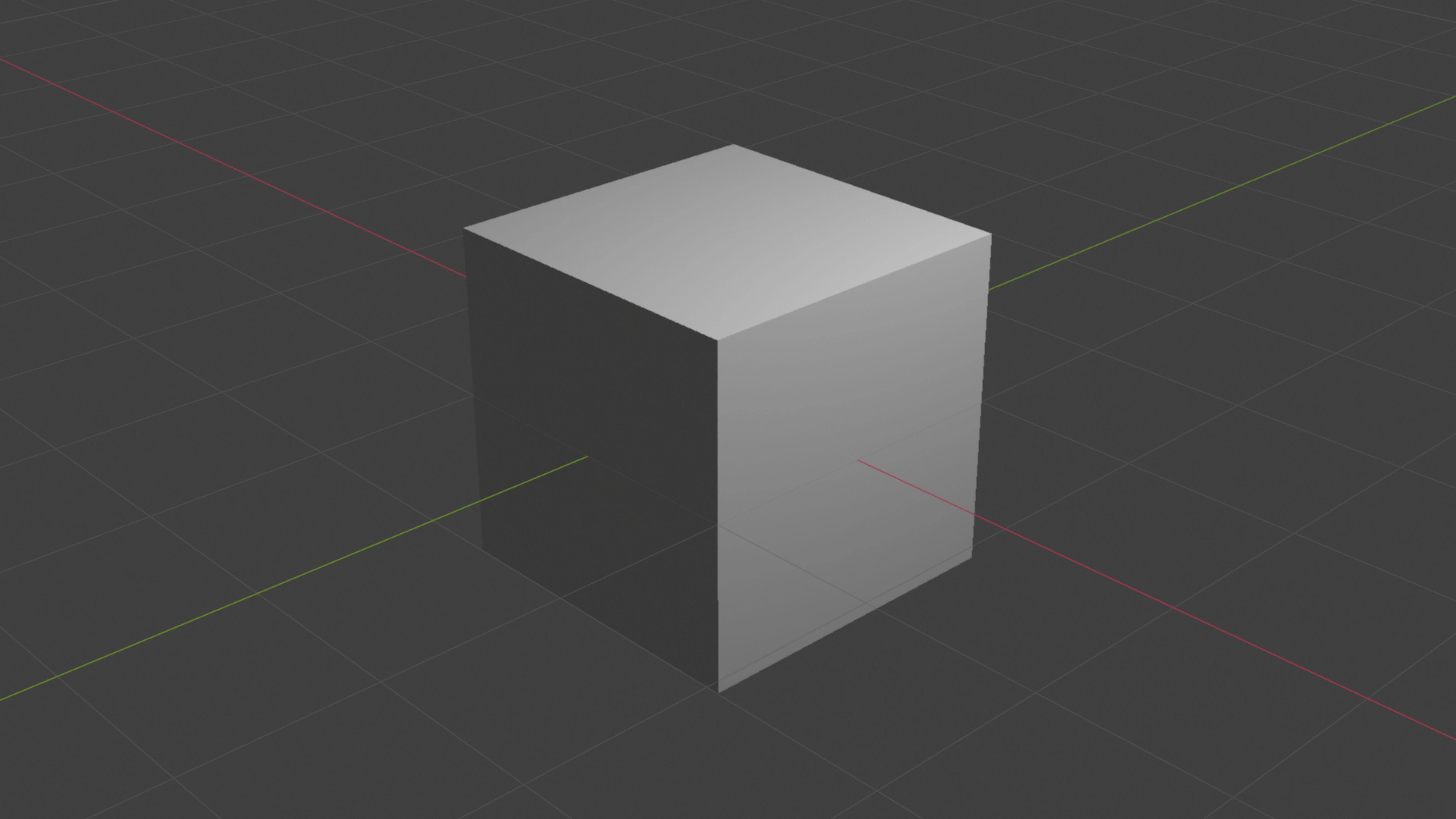
GNU GPL licenties

Blender is the
community driven
open source
(3D) content creation suite

Omvat A tot Z om een
productie te maken.

Van concept art to finale
encoding.

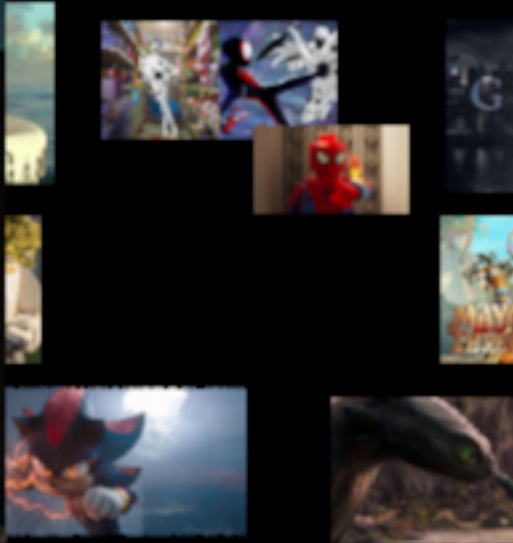
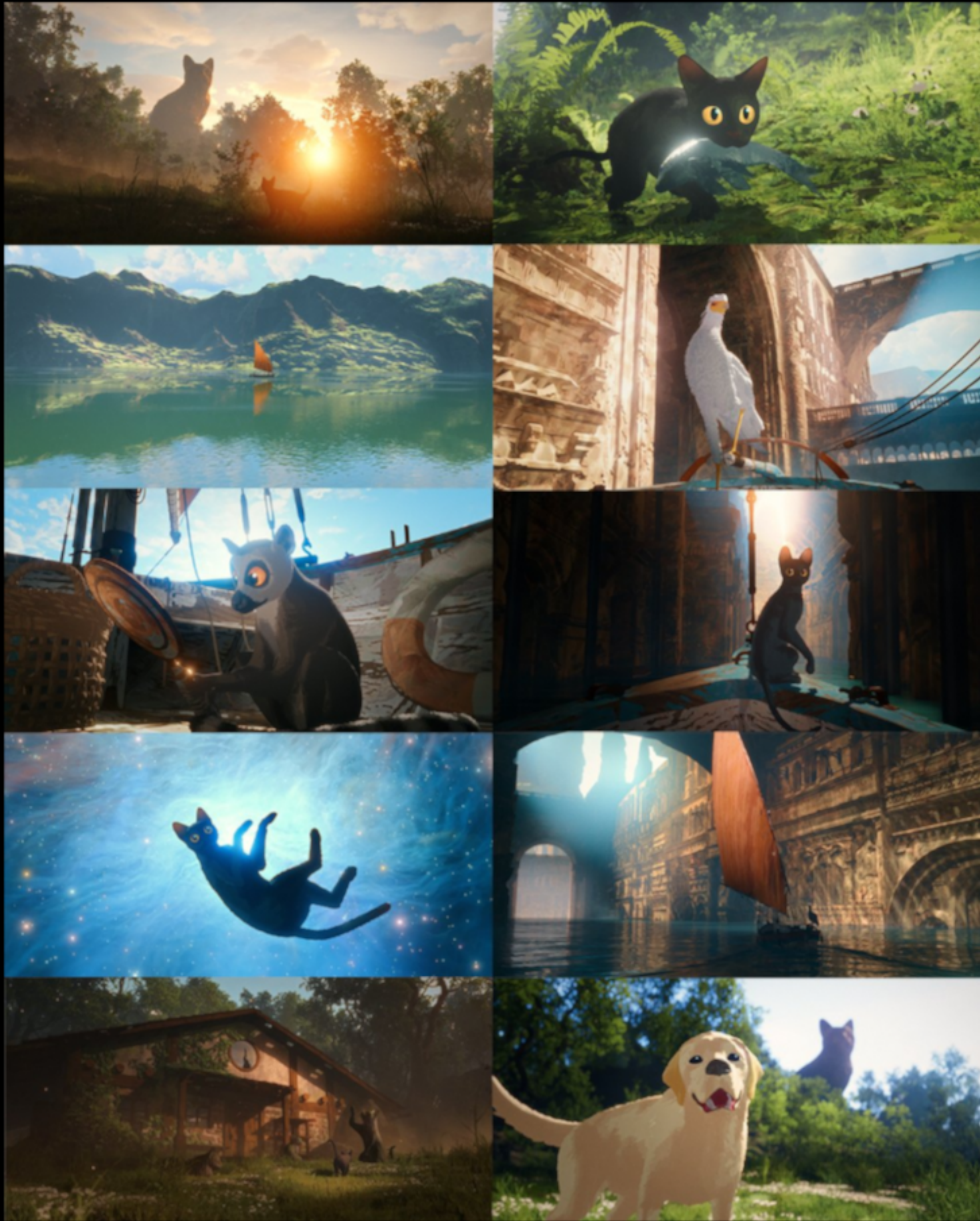
Via een volledig geïntegreerd
data model.



OKG







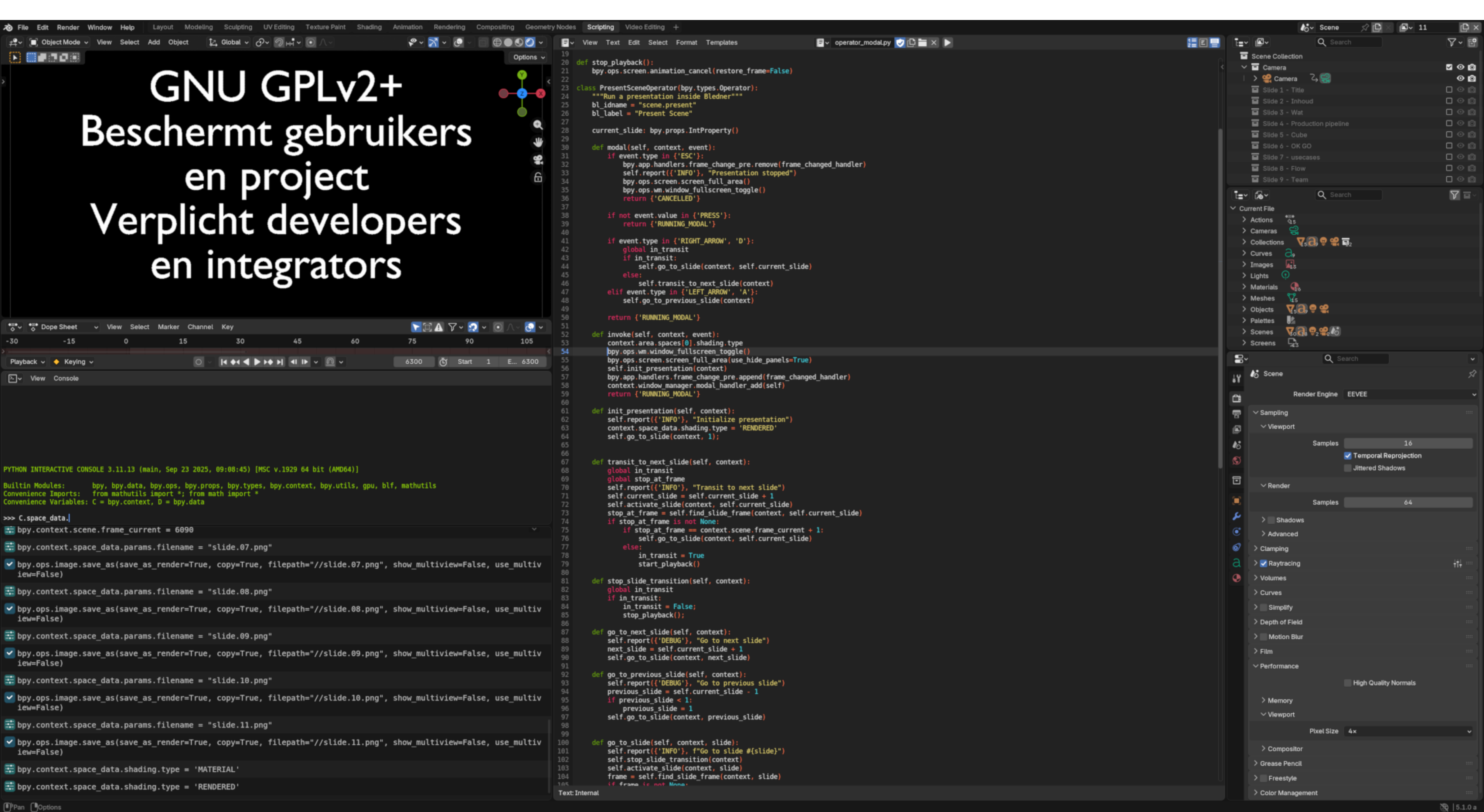


Blender Foundation
Blender Institute
Blender Studio

GNU GPLv2+

Beschermt gebruikers
en project

Verplicht developers
en integrators



GNU GPLv2+
Beschermt gebruikers
en project
Verplicht developers
en integrators

```
19
20 def stop_playback():
21     bpy.ops.screen.animation_cancel(restore_frame=False)
22
23 class PresentSceneOperator(bpy.types.Operator):
24     """Run a presentation inside Bledner"""
25     bl_idname = "scene.present"
26     bl_label = "Present Scene"
27
28     current_slide: bpy.props.IntProperty()
29
30     def modal(self, context, event):
31         if event.type in {'ESC'}:
32             bpy.app.handlers.frame_change_pre.remove(frame_changed_handler)
33             self.report({'INFO'}, "Presentation stopped")
34             bpy.ops.screen.screen_full_area()
35             bpy.ops.wm.window_fullscreen_toggle()
36             return {'CANCELLED'}
37
38         if not event.value in {'PRESS'}:
39             return {'RUNNING_MODAL'}
40
41         if event.type in {'RIGHT_ARROW', 'D'}:
42             global in_transit
43             if in_transit:
44                 self.go_to_slide(context, self.current_slide)
45             else:
46                 self.transit_to_next_slide(context)
47         elif event.type in {'LEFT_ARROW', 'A'}:
48             self.go_to_previous_slide(context)
49
50         return {'RUNNING_MODAL'}
51
52 def invoke(self, context, event):
53     context.area.spaces[0].shading.type
54     bpy.ops.wm.window_fullscreen_toggle()
55     bpy.ops.screen.screen_full_area(use_hide_panels=True)
56     self.init_presentation(context)
57     bpy.app.handlers.frame_change_pre.append(frame_changed_handler)
58     context.window_manager.modal_handler_add(self)
59     return {'RUNNING_MODAL'}
60
61 def init_presentation(self, context):
62     self.report({'INFO'}, "Initialize presentation")
63     context.space_data.shading.type = 'RENDERED'
64     self.go_to_slide(context, 1)
65
66
67 def transit_to_next_slide(self, context):
68     global in_transit
69     global stop_at_frame
70     self.report({'INFO'}, "Transit to next slide")
71     self.current_slide = self.current_slide + 1
72     self.activate_slide(context, self.current_slide)
73     stop_at_frame = self.find_slide_frame(context, self.current_slide)
74     if stop_at_frame is not None:
75         if stop_at_frame == context.scene.frame_current + 1:
76             self.go_to_slide(context, self.current_slide)
77         else:
78             in_transit = True
79             start_playback()
80
81 def stop_slide_transition(self, context):
82     global in_transit
83     if in_transit:
84         in_transit = False
85         stop_playback()
86
87 def go_to_next_slide(self, context):
88     self.report({'DEBUG'}, "Go to next slide")
89     next_slide = self.current_slide + 1
90     self.go_to_slide(context, next_slide)
91
92 def go_to_previous_slide(self, context):
93     self.report({'DEBUG'}, "Go to previous slide")
94     previous_slide = self.current_slide - 1
95     if previous_slide < 1:
96         previous_slide = 1
97     self.go_to_slide(context, previous_slide)
98
99
100 def go_to_slide(self, context, slide):
101     self.report({'INFO'}, f"Go to slide #{slide}")
102     self.stop_slide_transition(context)
103     self.activate_slide(context, slide)
104     frame = self.find_slide_frame(context, slide)
105     if frame is not None:
```

PYTHON INTERACTIVE CONSOLE 3.11.13 (main, Sep 23 2025, 09:08:45) [MSC v.1929 64 bit (AMD64)]

Builtin Modules: bpy, bpy.data, bpy.ops, bpy.props, bpy.types, bpy.context, bpy.utils, gpu, blf, mathutils
Convenience Imports: from mathutils import *; from math import *
Convenience Variables: C = bpy.context, D = bpy.data

>>> C.space_data.

bpy.context.scene.frame_current = 6090

bpy.context.space_data.params.filename = "slide.07.png"

bpy.ops.image.save_as(save_as_render=True, copy=True, filepath="//slide.07.png", show_multiview=False, use_multiview=False)

bpy.context.space_data.params.filename = "slide.08.png"

bpy.ops.image.save_as(save_as_render=True, copy=True, filepath="//slide.08.png", show_multiview=False, use_multiview=False)

bpy.context.space_data.params.filename = "slide.09.png"

bpy.ops.image.save_as(save_as_render=True, copy=True, filepath="//slide.09.png", show_multiview=False, use_multiview=False)

bpy.context.space_data.params.filename = "slide.10.png"

bpy.ops.image.save_as(save_as_render=True, copy=True, filepath="//slide.10.png", show_multiview=False, use_multiview=False)

bpy.context.space_data.params.filename = "slide.11.png"

bpy.ops.image.save_as(save_as_render=True, copy=True, filepath="//slide.11.png", show_multiview=False, use_multiview=False)

bpy.context.space_data.shading.type = 'MATERIAL'

bpy.context.space_data.shading.type = 'RENDERED'

Pan Options

5.1.0 a