



Forum: Python & Plugins

Topic: Pouvoir choisir l'outil actif avec python

Subject: Re: Pouvoir choisir l'outil actif avec python

Posté par: pierre-y

Contribution le : 25/5/2020 12:20:16

Merci a testeur pour l'aide :

```
import bpy bl_info = { "name": "Grease Pencil switch Draw<->select", "author": "Django950",
"version": (1, 0), "blender": (2, 82, 0), "location": "View3D > UI> GP select draw",
"description": "switch draw select, takes draw and select in activ tool", "warning": "", "wiki_url":
"", "category": "Animation", } # ----- #
creation of an addon that will switch select to draw in Grease Pencil #
----- #
----- # PROPERTY # Armature name
bpy.types.WindowManager.bool_select = bpy.props.BoolProperty(name = "select", default=True)
bpy.types.WindowManager.bool_draw = bpy.props.BoolProperty(name = "draw", default=True) #
----- #
OP_Switch_GP : switch between select and draw of the grease pencil def Switch() : print("def
Switch start") context = bpy.context scene = context.scene select
=context.window_manager.bool_select draw=context.window_manager.bool_draw if
context.mode == '#039;EDIT_GPENCIL#039;:
bpy.ops.object.mode_set(mode='#039;PAINT_GPENCIL#039;) if draw :
bpy.ops.wm.tool_set_by_id(name='#039;builtin_brush.Draw#039;) return elif
context.mode == '#039;PAINT_GPENCIL#039;:
bpy.ops.object.mode_set(mode='#039;EDIT_GPENCIL#039;) if select :
bpy.ops.wm.tool_set_by_id(name='#039;builtin.select_box#039;) return class
OP_Switch_GP (bpy.types.Operator) : bl_idname = "object.oper_select_draw" bl_label =
"switch" bl_description = "switch select to draw in Grease Pencil" def execute(self, context) :
Switch() return {'#039;FINISHED#039;} #
----- # PANEL GREASE PENCIL SWITCH : SELECT <-> DRAW
class InterfacePanel (bpy.types.Panel) : bl_label = "Switch Draw <-> select" # bl_label =
"Switch" bl_idname = "VIEW_3D_PT_SELECT_DRAW" bl_space_type =
'#039;VIEW_3D#039; bl_region_type = '#039;UI#039; bl_category = "GP select-draw"
bl_description = "Grease Pencil switch Draw <-> select" # --- draw --- #
def draw(self, context) : window_m = context.window_manager l = self.layout
c=l.column(align = True) c.operator(OP_Switch_GP.bl_idname) c.label(text
="edit mode :") c.prop(window_m, "bool_select") c.label(text ="draw :")
c.prop(window_m, "bool_draw") # ----- # REGISTER
classes = (InterfacePanel, OP_Switch_GP) def register() : for cls in classes:
bpy.utils.register_class(cls) def unregister() : for cls in classes:
bpy.utils.unregister_class(cls) if __name__ == "__main__" : register()
```