

```

diff --git a/scripts/startup/bl_ui/node_add_menu_compositor.py b/scripts/startup/bl_ui/node_add_menu_compositor.py
index 28a243b751a..6c0187d610c 100644
--- a/scripts/startup/bl_ui/node_add_menu_compositor.py
+++ b/scripts/startup/bl_ui/node_add_menu_compositor.py
@@ -100,6 +100,7 @@ class NODE_MT_category_compositor_color(Menu):
     layout.separator()
     node_add_menu.add_node_type(layout, "CompositorNodeInvert")
     node_add_menu.add_node_type(layout, "CompositorNodeRGBToBW")
+    node_add_menu.add_node_type(layout, "CompositorNodeCrazy")

     node_add_menu.draw_assets_for_catalog(layout, self.bl_label)

diff --git a/source/blender/blenkernel/BKE_node.h b/source/blender/blenkernel/BKE_node.h
index ef3ec6466a8..5c52d8c9cd0 100644
--- a/source/blender/blenkernel/BKE_node.h
+++ b/source/blender/blenkernel/BKE_node.h
@@ -1054,6 +1054,7 @@ void BKE_nodetree_remove_layer_n(struct bNodeTree *ntree, struct Scene *scene, i
#define CMP_NODE_TONEMAP 302
#define CMP_NODE_LENSDIST 303
#define CMP_NODE_SUNBEAMS 304
+#define CMP_NODE_CRAZY 305

#define CMP_NODE_COLORCORRECTION 312
#define CMP_NODE_MASK_BOX 313
diff --git a/source/blender/compositor/CMakeLists.txt b/source/blender/compositor/CMakeLists.txt
index 4b7bcda220e..78e4cdc0c1d 100644
--- a/source/blender/compositor/CMakeLists.txt
+++ b/source/blender/compositor/CMakeLists.txt
@@ -233,6 +233,8 @@ if(WITH_COMPOSITOR_CPU)
    nodes/COM_ConvertAlphaNode.h
    nodes/COM_ConvertColorSpaceNode.cc
    nodes/COM_ConvertColorSpaceNode.h
+   nodes/COM_CrazyNode.cc
+   nodes/COM_CrazyNode.h
    nodes/COM_GammaNode.cc
    nodes/COM_GammaNode.h
    nodes/COM_HueSaturationValueCorrectNode.cc
@@ -496,6 +498,8 @@ if(WITH_COMPOSITOR_CPU)
    operations/COM_ColorBalanceASCCDL0operation.h
    operations/COM_ColorBalanceLGG0operation.cc
    operations/COM_ColorBalanceLGG0operation.h
+   operations/COM_CrazyOperation.cc
+   operations/COM_CrazyOperation.h
    operations/COM_InvertOperation.cc
    operations/COM_InvertOperation.h
    operations/COM_MapRange0operation.cc
diff --git a/source/blender/compositor/intern/COM_Converter.cc b/source/blender/compositor/intern/COM_Converter.cc
index 59aa6413ee3..c00b539e6ac 100644
--- a/source/blender/compositor/intern/COM_Converter.cc
+++ b/source/blender/compositor/intern/COM_Converter.cc
@@ -38,6 +38,7 @@
#include "COM_ConvertOperation.h"
#include "COM_Converter.h"
#include "COM_CornerPinNode.h"
+#include "COM_CrazyNode.h"
#include "COM_CropNode.h"
#include "COM_CryptomatteNode.h"
#include "COM_DefocusNode.h"
@@ -440,6 +441,9 @@ Node *COM_convert_bnode(bNode *b_node)
    case CMP_NODE_KUWAHARA:
        node = new KuwaharaNode(b_node);
        break;
+   case CMP_NODE_CRAZY:
+       node = new CrazyNode(b_node);
+       break;
}
return node;
}

diff --git a/source/blender/compositor/nodes/COM_CrazyNode.cc b/source/blender/compositor/nodes/COM_CrazyNode.cc
new file mode 100644
index 00000000000..b860e06d272
--- /dev/null
+++ b/source/blender/compositor/nodes/COM_CrazyNode.cc
@@ -0,0 +1,23 @@
+/* SPDX-FileCopyrightText: 2011 Blender Authors
+ *

```

```

+ * SPDX-License-Identifier: GPL-2.0-or-later */
+
+#include "COM_CrazyNode.h"
+#include "BKE_node.hh"
+#include "COM_CrazyOperation.h"
+
+namespace blender::compositor {
+
+void CrazyNode::convert_to_operations(NodeConverter &converter,
+                                     const CompositorContext &context) const
+{
+  CrazyOperation *operation = new CrazyOperation();
+
+  /*Map the input socket to operation socket.*/
+  converter.add_operation(operation);
+  converter.map_input_socket(get_input_socket(0), operation->get_input_socket(0));
+  converter.map_input_socket(get_input_socket(1), operation->get_input_socket(1));
+  converter.map_output_socket(get_output_socket(0), operation->get_output_socket());
+}
+
+} // namespace blender::compositor
\ No newline at end of file
diff --git a/source/blender/compositor/nodes/COM_CrazyNode.h b/source/blender/compositor/nodes/COM_CrazyNode.h
new file mode 100644
index 000000000000..422f27fc919
--- /dev/null
+++ b/source/blender/compositor/nodes/COM_CrazyNode.h
@@ -0,0 +1,22 @@
+/* SPDX-FileCopyrightText: 2011 Blender Authors
+ *
+ * SPDX-License-Identifier: GPL-2.0-or-later */
+
+#pragma once
+
+#include "COM_Node.h"
+
+namespace blender::compositor {
+
+/**
+ * \brief CrazyNode
+ * \ingroup Node
+ */
+class CrazyNode : public Node {
+ public:
+  CrazyNode(bNode *editor_node) : Node(editor_node) {}
+  void convert_to_operations(NodeConverter &converter,
+                            const CompositorContext &context) const override;
+};
+
+} // namespace blender::compositor
\ No newline at end of file
diff --git a/source/blender/compositor/operations/COM_CrazyOperation.cc b/source/blender/compositor/operations/COM_CrazyOperation.cc
new file mode 100644
index 000000000000..baa358edc56
--- /dev/null
+++ b/source/blender/compositor/operations/COM_CrazyOperation.cc
@@ -0,0 +1,52 @@
+/* SPDX-FileCopyrightText: 2023 Blender Authors
+ *
+ * SPDX-License-Identifier: GPL-2.0-or-later */
+
+#include "COM_CrazyOperation.h"
+
+namespace blender::compositor {
+
+CrazyOperation::CrazyOperation()
+{
+  this->add_input_socket(DataType::Value); // Craziness.
+  this->add_input_socket(DataType::Color);
+
+  this->add_output_socket(DataType::Color);
+
+  /*Set the input canvas to socket(1), which is the color input.*/
+  set_canvas_input_index(1);
+}
+
+void CrazyOperation::init_execution()
+{

```



```

+                                     Span<MemoryBuffer *> inputs) override;
+};
+
+} // namespace blender::compositor
diff --git a/source/blender/makesdna/DNA_node_types.h b/source/blender/makesdna/DNA_node_types.h
index ab9571240ee..8ebe61983e0 100644
--- a/source/blender/makesdna/DNA_node_types.h
+++ b/source/blender/makesdna/DNA_node_types.h
@@ -1050,6 +1050,10 @@ typedef struct NodeKuwaharaData {
    float eccentricity;
} NodeKuwaharaData;

+typedef struct NodeCrazyData {
+ float craziness;
+} NodeCrazyData;
+
+typedef struct NodeAntiAliasingData {
+ float threshold;
+ float contrast_limit;
diff --git a/source/blender/makesrna/intern/rna_nodetree.cc b/source/blender/makesrna/intern/rna_nodetree.cc
index 053de45e33a..bb18bc29bd6 100644
--- a/source/blender/makesrna/intern/rna_nodetree.cc
+++ b/source/blender/makesrna/intern/rna_nodetree.cc
@@ -8616,6 +8616,24 @@ static void def_cmp_kuwahara(StructRNA *srna)
    RNA_def_property_update(prop, NC_NODE | NA_EDITED, "rna_Node_update");
}

+static void def_cmp_crazy(StructRNA *srna)
+{
+ PropertyRNA *prop;
+
+ RNA_def_struct_sdna_from(srna, "NodeCrazyData", "storage");
+
+ prop = RNA_def_property(srna, "craziness", PROP_FLOAT, PROP_NONE);
+ RNA_def_property_float_sdna(prop, nullptr, "craziness");
+ RNA_def_property_range(prop, -10.0f, 10.0f);
+ RNA_def_property_ui_range(prop, -10.0f, 10.0f, 0.1, 3);
+ RNA_def_property_ui_text(
+     prop,
+     "Craziness",
+     "Controls how crazy the output is.");
+ RNA_def_property_update(prop, NC_NODE | NA_EDITED, "rna_Node_update");
+}
+
+static void def_cmp_antialiasing(StructRNA *srna)
+{
+ PropertyRNA *prop;
diff --git a/source/blender/nodes/NOD_static_types.h b/source/blender/nodes/NOD_static_types.h
index 7e9658646c4..0b47e7a473d 100644
--- a/source/blender/nodes/NOD_static_types.h
+++ b/source/blender/nodes/NOD_static_types.h
@@ -225,6 +225,7 @@ DefNode(CompositorNode, CMP_NODE_SEPARATE_XYZ, 0, "SEPARATE_XYZ", SeparateXYZ, "SeparateXYZ")
DefNode(CompositorNode, CMP_NODE_SEPARATE_COLOR, def_cmp_combsep_color, "SEPARATE_COLOR", SeparateColor, "SeparateColor")
DefNode(CompositorNode, CMP_NODE_COMBINE_COLOR, def_cmp_combsep_color, "COMBINE_COLOR", CombineColor, "CombineColor")
DefNode(CompositorNode, CMP_NODE_KUWAHARA, def_cmp_kuwahara, "KUWAHARA", Kuwahara, "Kuwahara")
+DefNode(CompositorNode, CMP_NODE_CRAZY, def_cmp_crazy, "CRAZY", Crazy, "Crazy")
+
DefNode(TextureNode, TEX_NODE_OUTPUT, def_tex_output, "OUTPUT", Output, "Output")
DefNode(TextureNode, TEX_NODE_CHECKER, 0, "CHECKER", Checker, "Checker")
diff --git a/source/blender/nodes/composite/CMakeLists.txt b/source/blender/nodes/composite/CMakeLists.txt
index 054bbc11a07..6eb7e1d70d7 100644
--- a/source/blender/nodes/composite/CMakeLists.txt
+++ b/source/blender/nodes/composite/CMakeLists.txt
@@ -46,6 +46,7 @@ set(SRC
    nodes/node_composite_composite.cc
    nodes/node_composite_convert_color_space.cc
    nodes/node_composite_cornerpin.cc
+   nodes/node_composite_crazy.cc
    nodes/node_composite_crop.cc
    nodes/node_composite_cryptomatte.cc
    nodes/node_composite_curves.cc
diff --git a/source/blender/nodes/composite/node_composite_register.cc b/source/blender/nodes/composite/node_composite_register.cc
index 89c3639d730..9e01952e6f6 100644
--- a/source/blender/nodes/composite/node_composite_register.cc
+++ b/source/blender/nodes/composite/node_composite_register.cc
@@ -35,6 +35,7 @@ void register_composite_nodes()

```

```

register_node_type_cmp_composite();
register_node_type_cmp_convert_color_space();
register_node_type_cmp_cornerpin();
+ register_node_type_cmp_crazy();
register_node_type_cmp_crop();
register_node_type_cmp_cryptomatte_legacy();
register_node_type_cmp_cryptomatte();
diff --git a/source/blender/nodes/composite/node_composite_register.hh b/source/blender/nodes/composite/node_composite
index 39e55e3d20d..5f940947ef2 100644
--- a/source/blender/nodes/composite/node_composite_register.hh
+++ b/source/blender/nodes/composite/node_composite_register.hh
@@ -31,6 +31,7 @@ void register_node_type_cmp_combyuva();
void register_node_type_cmp_composite();
void register_node_type_cmp_convert_color_space();
void register_node_type_cmp_cornerpin();
+void register_node_type_cmp_crazy();
void register_node_type_cmp_crop();
void register_node_type_cmp_cryptomatte_legacy();
void register_node_type_cmp_cryptomatte();
diff --git a/source/blender/nodes/composite/nodes/node_composite_crazy.cc b/source/blender/nodes/composite/nodes/node_c
new file mode 100644
index 00000000000..fd16d139889
--- /dev/null
+++ b/source/blender/nodes/composite/nodes/node_composite_crazy.cc
@@ -0,0 +1,78 @@
+/* SPDX-FileCopyrightText: 2023 Blender Authors
+ *
+ * SPDX-License-Identifier: GPL-2.0-or-later */
+
+/** \file
+ * \ingroup cmpnodes
+ */
+
+#include "BLI_system.h"
+
+#include "MEM_guardedalloc.h"
+
+#include "UI_interface.hh"
+#include "UI_resources.hh"
+
+#include "DNA_node_types.h"
+
+#include "COM_node_operation.hh"
+
+#include "node_composite_util.hh"
+
+namespace blender::nodes::node_composite_crazy_cc {
+
+NODE_STORAGE_FUNCS(NodeCrazyData)
+
+static void cmp_node_crazy_declare(NodeDeclarationBuilder &b)
+{
+  b.add_input<decl::Float>("Craziness").default_value(0.5);
+  b.add_input<decl::Color>("Color")
+    .default_value({1.0f, 1.0f, 1.0f, 1.0f})
+    .compositor_domain_priority(0);
+  b.add_output<decl::Color>("Color");
+}
+
+static void node_composit_init_crazy(bNodeTree * /*ntree*/, bNode *node)
+{
+  NodeCrazyData *ncd = MEM_cnew<NodeCrazyData>(__func__);
+  ncd->craziness;
+  node->storage = ncd;
+}
+
+static void node_composit_butts_crazy(uiLayout *layout, bContext * /*C*/, PointerRNA *ptr)
+{
+  // uiItemR(layout, ptr, "craziness", UI_ITEM_R_SPLIT_EMPTY_NAME, nullptr, ICON_NONE);
+}
+
+using namespace blender::realtime_compositor;
+
+class CrazyOperation : public NodeOperation {
+public:
+  using NodeOperation::NodeOperation;
+}

```

```
+ void execute() override {}
+};
+
+static NodeOperation *get_compositor_operation(Context &context, DNode node)
+{
+    return new CrazyOperation(context, node);
+}
+
+} // namespace blender::nodes::node_composite_crazy_cc
+
+void register_node_type_cmp_crazy()
+{
+    namespace file_ns = blender::nodes::node_composite_crazy_cc;
+
+    static bNodeType ntype;
+
+    cmp_node_type_base(&ntype, CMP_NODE_CRAZY, "Crazy", NODE_CLASS_OP_COLOR);
+    ntype.declare = file_ns::cmp_node_crazy_declare;
+    ntype.draw_buttons = file_ns::node_composit_butts_crazy;
+    ntype.initfunc = file_ns::node_composit_init_crazy;
+    node_type_storage(
+        &ntype, "NodeCrazyData", node_free_standard_storage, node_copy_standard_storage);
+    ntype.get_compositor_operation = file_ns::get_compositor_operation;
+
+    nodeRegisterType(&ntype);
+}
+
+ \ No newline at end of file
```