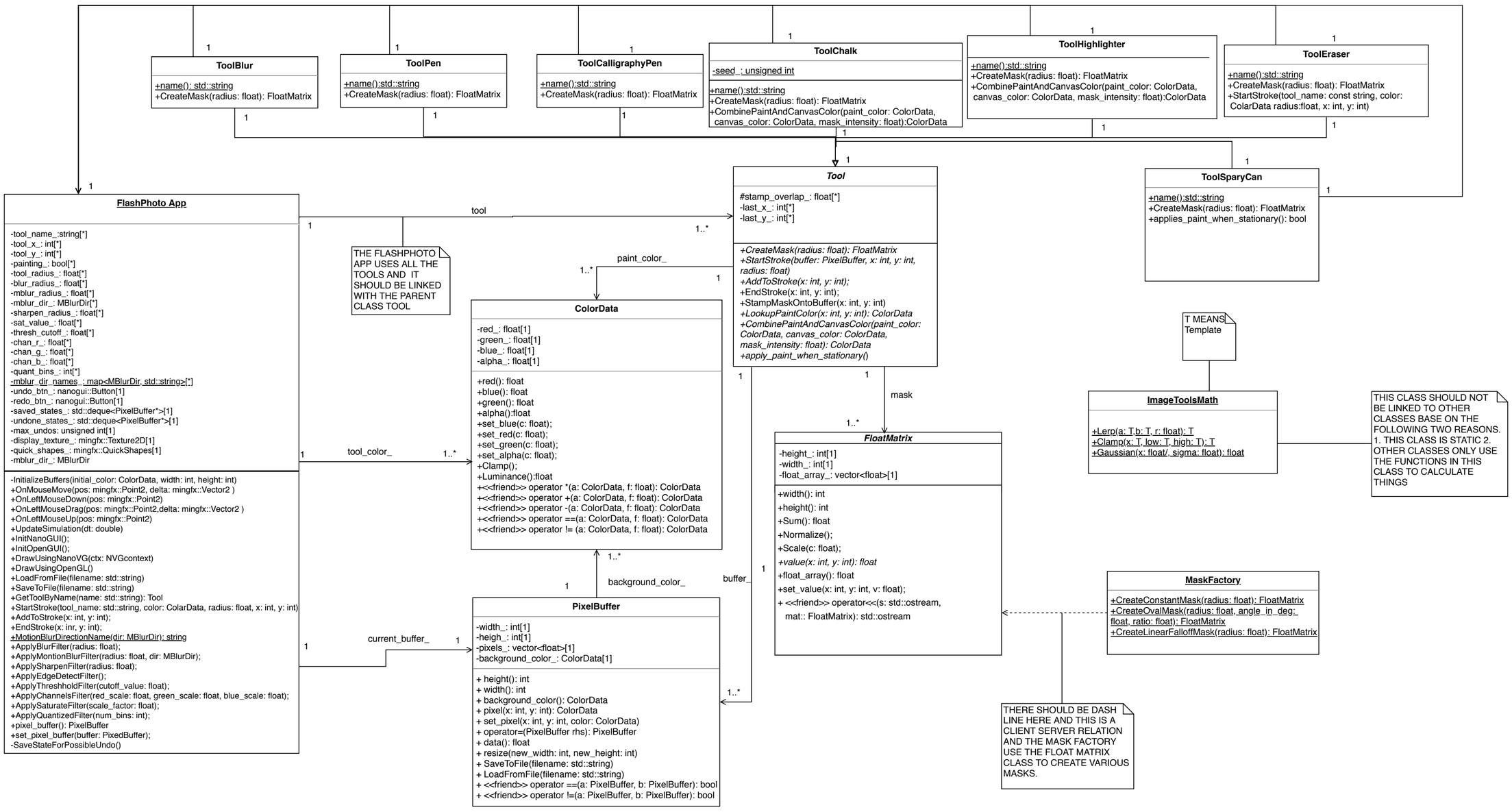
ToolBlur
+name(): std::string
+CreateMask(radius: float): FloatMatrix
ToolPen
+name():std::string
+CreateMask(radius: float): FloatMatrix
ToolCalligraphyPen
+name():std::string
+CreateMask(radius: float): FloatMatrix
ToolChalk
-seed_ : unsigned int
+name():std::string
+CreateMask(radius: float): FloatMatrix
+CombinePaintAndCanvasColor(paint_color: ColorData, canvas_color: ColorData, mask_intensity: float):ColorData
ToolHighlighter
+name():std::string
+CreateMask(radius: float): FloatMatrix
+CombinePaintAndCanvasColor(paint_color: ColorData, canvas_color: ColorData, mask_intensity: float):ColorData
ToolEraser
+name():std::string
+CreateMask(radius: float): FloatMatrix
+StartStroke(tool_name: const string, color: ColarData radius:float, x: int, y: int)
FlashPhoto App
-tool_name_:string[*]
-tool_x_: int[*]
-tool_y_: int[*]
-painting_: bool[*]
-tool_radius_: float[*]
-blur_radius_: float[*]
-mblur_radius_: float[*]
-mblur_dir_: MBlurDir[*]
-sharpen_radius_: float[*]
-sat_value_: float[*]
-thresh_cutoff_: float[*]
-chan_r_: float[*]
-chan_g_: float[*]
-chan_b_: float[*]
-quant_bins_: int[*]
-mblur_dir_names_: map<MBlurDir, std::string>[*]
-undo_btn_: nanogui::Button[1]
-redo_btn_: nanogui::Button[1]
-saved_states_: std::deque<PixelBuffer*>[1]
-undone_states_: std::deque<PixelBuffer*>[1]
-max_undos: unsigned int[1]
-display_texture_: mingfx::Texture2D[1]
-quick_shapes_: mingfx::QuickShapes[1]
-mblur_dir_: MBlurDir
-InitializeBuffers(initial_color: ColorData, width: int, height: int)
+OnMouseMove(pos: mingfx::Point2, delta: mingfx::Vector2)
+OnLeftMouseDown(pos: mingfx::Point2)
+OnLeftMouseDrag(pos: mingfx::Point2,delta: mingfx::Vector2)
+OnLeftMouseUp(pos: mingfx::Point2)
+UpdateSimulation(dt: double)
+InitNanoGUI();
+InitOpenGUI();
+DrawUsingNanoVG(ctx: NVGcontext)
+DrawUsingOpenGL()
+LoadFromFile(filename: std::string)
+SaveToFile(filename: std::string)
+GetToolByName(name: std::string): Tool
+StartStroke(tool_name: std::string, color: ColarData, radius: float, x: int, y: int)
+AddToStroke(x: int, y: int);
+EndStroke(x: inr, y: int);
+MotionBlurDirectionName(dir: MBlurDir): string
+ApplyBlurFilter(radius: float);
+ApplyMontionBlurFilter(radius: float, dir: MBlurDir);
+ApplySharpenFilter(radius: float);
+ApplyEdgeDetectFilter();
+ApplyThreshholdFilter(cutoff_value: float);
+ApplyChannelsFilter(red_scale: float, green_scale: float, blue_scale: float);
+ApplySaturateFilter(scale_factor: float);
+ApplyQuantizedFilter(num_bins: int);
+pixel_buffer(): PixelBuffer
+set_pixel_buffer(buffer: PixedBuffer);
-SaveStateForPossibleUndo()
tool
1
1..*
THE FLASHPHOTO APP USES ALL THE TOOLS AND IT SHOULD BE LINKED WITH THE PARENT CLASS TOOL
Tool
#stamp_overlap_: float[*]
-last_x_: int[*]
-last_y_: int[*]
+CreateMask(radius: float): FloatMatrix
+StartStroke(buffer: PixelBuffer, x: int, y: int, radius: float)
+AddToStroke(x: int, y: int);
+EndStroke(x: int, y: int);
+StampMaskOntoBuffer(x: int, y: int)
+LookupPaintColor(x: int, y: int): ColorData
+CombinePaintAndCanvasColor(paint_color: ColorData, canvas_color: ColorData, mask_intensity: float): ColorData
+apply_paint_when_stationary()
ToolSparyCan
+name():std::string
+CreateMask(radius: float): FloatMatrix
+applies_paint_when_stationary(): bool
paint_color_
1..*
1
ColorData
-red_: float[1]
-green_: float[1]
-blue_: float[1]
-alpha_: float[1]
+red(): float
+blue(): float
+green(): float
+alpha():float
+set_blue(c: float);
+set_red(c: float);
+set_green(c: float);
+set_alpha(c: float);
+Clamp();
+Luminance():float
+<<friend>> operator *(a: ColorData, f: float): ColorData
+<<friend>> operator +(a: ColorData, f: float): ColorData
+<<friend>> operator -(a: ColorData, f: float): ColorData
+<<friend>> operator ==(a: ColorData, f: float): ColorData
+<<friend>> operator != (a: ColorData, f: float): ColorData
mask
FloatMatrix
-height_: int[1]
-width_: int[1]
-float_array_: vector<float>[1]
+width(): int
+height(): int
+Sum(): float
+Normalize();
+Scale(c: float);
+value(x: int, y: int): float
+float_array(): float
+set_value(x: int, y: int, v: float);
+ <<friend>> operator<<(s: std::ostream, mat:: FloatMatrix): std::ostream
T MEANS Template
ImageToolsMath
+Lerp(a: T,b: T, r: float): T
+Clamp(x: T, low: T, high: T): T
+Gaussian(x: float/, sigma: float): float
THIS CLASS SHOULD NOT BE LINKED TO OTHER CLASSES BASE ON THE FOLLOWING TWO REASONS. 1. THIS CLASS IS STATIC 2. OTHER CLASSES ONLY USE THE FUNCTIONS IN THIS CLASS TO CALCULATE THINGS
tool_color_
background_color_
buffer_
current_buffer_
PixelBuffer
-width_: int[1]
-heigh_: int[1]
-pixels_: vector<float>[1]
-background_color_: ColorData[1]
+ height(): int
+ width(): int
+ background_color(): ColorData
+ pixel(x: int, y: int): ColorData
+ set_pixel(x: int, y: int, color: ColorData)
+ operator=(PixelBuffer rhs): PixelBuffer
+ data(): float
+ resize(new_width: int, new_height: int)
+ SaveToFile(filename: std::string)
+ LoadFromFile(filename: std::string)
+ <<friend>> operator ==(a: PixelBuffer, b: PixelBuffer): bool
+ <<friend>> operator !=(a: PixelBuffer, b: PixelBuffer): bool
MaskFactory
+CreateConstantMask(radius: float): FloatMatrix
+CreateOvalMask(radius: float, angle_in_deg: float, ratio: float): FloatMatrix
+CreateLinearFalloffMask(radius: float): FloatMatrix
THERE SHOULD BE DASH LINE HERE AND THIS IS A CLIENT SERVER RELATION AND THE MASK FACTORY USE THE FLOAT MATRIX CLASS TO CREATE VARIOUS MASKS.