Protobyte API

Contributors: Brittni L. Watkins

***API Key:***

* ***Headers***
* **/Directories/and/Paths.ext**
* *Code*
* Menu Options
* Problem/Question

***Variables:***

* Project Title
  + The title of your project
* projectName
  + Name of your project. Will be used to name main project .cpp and .h files

***Beginning a New Project:***

1. Creating Project Files
   1. Open terminal window
   2. Navigate to **project\_wizard** directory of Protobyte
      1. Current Absolute Path
         1. **/Dropbox/ira\_dev/Protobyte\_0.1.0\_quark/Protobyte/project\_wizard/**
   3. Execute command *./wizard projectName*
      1. Creates .cpp and .h files in **/Protobyte\_0.1.0\_quark /Protobyte/src**
2. Adding Project Files to Protobyte Environment
   1. Open Visual Studios
   2. Open project file
      1. Current Absolute Path
         1. **/Dropbox/ira\_dev/Protobyte\_0.1.0\_quark/protobyte\_dev\_WIN/Zhlong\_dev\_WIN/Shlonh\_dev\_WIN.vcxproj**
   3. Add new project files to Visual Studios Project
      1. Right click **srcProtobyte** folder under **Protobyte\_Project**
      2. Hover over Add menu option
      3. Select Existing Item
      4. Navigate to **src** directory
         1. Current Absolute Path
            1. **/Dropbox/ira\_dev/Protobyte\_0.1.0\_quark/Protobyte/src/**
      5. Select and Add projectName.h and projectName.cpp
3. Running new project
   1. Open ProtoMain.cpp
   2. At top of document
      1. *#include “projectName.h”*
   3. In main()
      1. ijg::ProtoPlasm objectName(windowWidth, windowHeight, “Project Title”, new projectName());

***General Information about Visual Environment:***

* Origin (0, 0) is at the center of the window
* Default background color is black
* Default stoke color is black
* Default fill color is white
* RGB is on a 0-1 scale
* Greyscale
  + 0 = black
  + 1 = white

***Visual Environment:***

Variables:

std::string appTitle

int width

* Is equal to the width of the window
  + Set in ProtoMain.cpp
* Alternate name for canvasWidth

int height

* Is equal to the height of the window
  + Set in ProtoMain.cpp
* Alternate name for canvasHeight

int appWidth

int appHeight

int canvasWidth

* Is equal to the width of the canvas
* Set in ProtoMain.cpp
* Alternate name for width

int canvasHeight

* Is equal to width of the window
* Set in ProtoMain.cpp
* Alternate name for height

Dim2i canvasSize

* Is equal to the dimensions of the canvas
* Set in ProtoMain.cpp
* Alternate name for windowFrameSize

Dim2i windowFrameSize

* Is equal to the dimensions of the window
* Set in ProtoMain.cpp
* Alternate name for canvasSize

int frameCount

float frameRate

bool isStroke

bool isFill

float linewidth

Col3f bgColor

Col4f fillColor

Col4f strokeColor

Functions:

void setWindowFrameSize(const Dim2i& windowFrameSize)

int getWidth();

int getHeight();

Dim2i getSize();

void background(float color)

void setBackground(float color)

void stroke(float color)

void strokeWeight(float lineWidth)

void noStroke()

void fill(float color)

void noFill()

getFrameCount()

setFrameCount()

getFrameRate()

setFrameRate()

push()

pop()

translate()

rotate()

scale()

***Objects:***

Col4f

Col4

Col3f

Col3

Vec2f

Vec2

***Mouse/Key Events:***

Global Variables:

float mouseX

float mouseY

float mouseLastFrameX

float mouseLastFrameY

int mouseButton

int mouseAction

int mouseMods

bool isMousePressed

Functions:

void keyPressed()

void mousePressed()

void mouseRightPressed()

void mouseReleased()

void mouseRightReleased()

void mouseMoved()

void mouseDragged()

***Two Dimensional Primitives:***

Rectangle

void rect(float radius1, float radius2);

* rect(width, height);
* Draws a rectangle centered at origin with width of *width* and height of *height*

void rect(const Vec2 &pt0, const Vec2 &pt1);

void rect(float x, float y, float w, float h);

***Suggestions:***

General Information about Visual Environment

* Default background color should be different from default stroke color
  + Maybe have default background color be a shade of grey

Visual Environment

* Overload = operator for Col4f = Col3f so that when using global variables to set fill and stroke color, users may pass the variable a Col4f object or a Col3f object

***Problems:***

Visual Environment

* appTitle = null
* appWidth = 0
* appHeight = 0

Mouse/Key Events

* Origin for mouseX and mouseY calculations is in the top left corner of the screen
  + Should be in the center

Two Dimensional Primitives

* Rectangle and Ellipse
  + x, y, width, and height measurements are not pixel accurate
    - Max width and height of Screen
      * 1800 x 1000
    - Max width and height of rectangle so that it fills the screen
      * 1380 x 770