CompArt

A language for creating digital art masterpieces

Aaron Priven, Julia Reichel, Asher Willner, Evan Zauderer

I: Team Members and Positions

Aaron Priven <ahp2154@columbia.edu> | Language Guru
Julia Reichel <jrr2189@barnard.edu> | Manager
Asher Willner <ajw2210@columbia.edu> | Tester
Evan Zauderer <emz2118@columbia.edu> | System Architect

*Roles are not binding for now, we will finalize later on

II: Introduction and Goals

Our language is meant to allow users to create beautiful digital canvases in an easy and efficient manner. We will build our language on top of the Simple DirectMedia Layer (SDL2) library, which makes drawing to a canvas accessible to a novice coder. Our goal is to design and develop a language that can assist people in drawing images and perhaps building simple quick animations. Artists can use our basic shapes to create more complex figures by defining new objects.

III: Language and Syntax

A. Data Types & Objects

```
a. int
b. string
c. Color
     a. Black      = (0,0,0) //Default Color for all Objects
     b. White      = (255,255,255)
     c. Red        = (255,0,0)
     d. Lime       = (0,255,0)
     e. Blue       = (0,0,255)
     f. Yellow     = (255,255,0)
     g. Aqua       = (0,255,255)
     h. Magenta    = (255,0,255)
     i. Silver     = (192,192,192)
     j. Gray       = (128,128,128)
     k. Maroon     = (128,0,0)
     l. Olive      = (128,128,0)
     m. Green      = (0,128,0)
     n. Purple     = (128,0,128)
     o. Teal       = (0,128,128)
```

p. Navy = (0,0,128)

d. Canvas (int height, int width, [Color background])

e. Point (int x, int y, [Color color])

f. Line (int x1, int y1, int x2, int y2, [Color color])

g. Circle (int xcenter, int ycenter, int radius, [Color color])

h. Ellipse (int xcenter, int ycenter, int width, int height, [Color color])

i. Square (int xcenter, int ycenter, int side, [Color color])

j. Rect (int xcenter, int ycenter, int width, int height, [Color color])

k. Arc (int xcenter, int ycenter, int radius, int startDegree, int endDegree, [Color color])

Note: [] are optional parameters

B. Operations

a. +

b. -

c. *

d. /

e. %

f. >, <, >=, <=, ==, !=

g. =

h. &&

i. ||

C. Syntax

- Our syntax will be Java inspired
- Semicolons are required at the end of every statement
- Use // for inline comments and /* */ for multiline comments.

D. Reserved Keywords

function, for, if, else, continue, break, new, while, true, false, class, return, this

E. Functions

a. Draw Functions

a. Circle

1. drawCirlce(Canvas, Circle)
2. drawCircle(Canvas, int xcenter, int ycenter, int radius, [Color color])

        b. Line
            1. `drawLine(Canvas, Line)`
            2. `drawLine(Canvas, int x1, int y1, int x2, int y2, [Color color])`
        c. Arc
            1. `drawArc(Canvas, Arc)`
            2. `drawArc(Canvas, int xcenter, int ycenter, int radius, int startDegree, int endDegree, [Color color])`
        d. Point
            1. `drawPoint(Canvas, Point)`
            2. `drawPoint(Canvas, int x, int y, [Color color])`
        e. Ellipse
            1. `drawEllipse(Canvas, Ellipse)`
            2. `drawEllipse(Canvas, int xcenter, int ycenter, int width, int height, [Color color])`
        f. Square
            1. `drawSquare(Canvas, Square)`
            2. `drawSquare(Canvas, int xcenter, int ycenter, int side, [Color color])`
        g. Rectangle
            1. `drawRect(Canvas, Rect)`
            2. `drawRect(Canvas, int xcenter, int ycenter, int width, int height, [Color color])`
    b. `Render(Canvas)`
        - Prints the canvas for the user to see
    c. `delay(int)` → helpful for rendering moving images
        - Delays the program for int milliseconds

F. Control Flow

    a. `If-else statements`
    b. `For-loop`
    c. `While loop`

G. Libraries

We plan on linking/implementing the following libraries:

    a. SDL2
        - https://wiki.libsdl.org/CategoryAPI
    a. SDL2_gfx
        - https://www.ferzkopp.net/Software/SDL2_gfx/Docs/html/files.html

H. Two Example Codes

```
//Example Code 1: Drawing a smiley face :)

function Eye (int x, int y) {
     int this.x = x;
     int this.y = y;
     int radius = 5;
     Color eyeColor = Blue;
     return new Circle(this.x, this.y, radius, eyeColor);
}

Canvas myFirstCanvas = new Canvas(height=100, width=100,
                                      background=Yellow);
Circle leftEye = Eye(20, 35);
Circle rightEye = Eye(20, 65);

// left and right eye
drawCircle(myFirstCanvas,
           leftEye.x,
           leftEye.y,
           leftEye.radius,
           leftEye.color);
//Can be written drawCircle(myFirstCanvas, leftEye);

drawCircle(myFirstCanvas,
           rightEye.x,
           rightEye.y,
           rightEye.radius,
           rightEye.color);
//Can be written drawCircle(myFirstCanvas, rightEye);

// smile
drawArc(myFirstCanvas, 50, 70, 20, 270, 90);

render(myFirstCanvas);
```

```
/*
  Example Code 2:
  Ball moving from left to right
*/

int canvasHeight = 720;
int canvasWidth = 400;

// Starts in the middle on the left border of screen
int x = 0;
int y = canvasHeight/2;

while (x<canvasWidth){
  Canvas myCanvas = new Canvas(canvasHeight, canvasWidth, Black);

  drawCircle(myCanvas, x, y, 20, Blue);

  render(myCanvas);

  // Moving right at a constant speed
  x = x + 10;

  delay(100);
}
```