**MIPS PIXEL PAINT DOCUMENTATION:**

**INSTRUCTIONS:**   
Connect bitmap display:   
Set pixel dim to 8x8  
Set display dim to 256x256  
Set $gp as base address  
Connect keyboard and run  
Use w (up), s (down), a (left), d (right), space (exit)  
Use numbers 0-9 to change colors  
Use r, c, b for bitmap functions

**OVERVIEW:** MIPS pixel paint is a lite version of Microsoft paint but simulated in a pixel format. While I was contemplating what to do for this project that would be creative, I thought outside the box. Why limit the creativity when I could create a program that promotes the user’s creativity. The program allows you to use the pixel brush to create complex images in the simplest form. Furthermore, to add more depth to the program I added welcome and closing animations to provide a more entertaining experience.

**PSEUDOCODE:**

welcomeAnimation {

set current pixel to the top left  
 loop:  
 paint each row red

set current pixel to the top left  
 loop:  
 paint each row orange

set current pixel to the top left  
 loop:  
 paint each row yellow

set current pixel to the top left  
 loop:  
 paint each row green

set current pixel to the top left  
 loop:  
 paint each row blue

set current pixel to the top left  
 loop:  
 paint each row purple

set current pixel to the top left  
 loop:  
 paint each row magenta

}  
 Print the welcome message

Print the info for the program  
set the current pixel to the center of the bitmap

mainLoop {

set the pixel brush to grey  
wait and read keyboard input

if w:  
 draw the next pixel above the current one  
if s:  
 draw the next pixel below the current one   
if a:  
 draw the next pixel to the left of the current one  
id d:  
 draw the next pixel to the right of the current one  
if 0:  
 change current color to black  
 display color change confirmation  
if 1:  
 change current color to red  
 display color change confirmation  
if 2:  
 change current color to green  
 display color change confirmation  
if 3:  
 change current color to blue  
 display color change confirmation  
if 4:  
 change current color to white  
 display color change confirmation  
if 5:  
 change current color to yellow  
 display color change confirmation  
if 6:  
 change current color to cyan  
 display color change confirmation  
if 7:  
 change current color to magenta  
 display color change confirmation   
if 8:  
 change current color to purple   
 display color change confirmation  
if 9:  
 change current color to orange  
 display color change confirmation  
if b:  
 backgroundChange()  
if r:  
 resetCanvas()  
if c:  
 center pixel()  
if space:  
 exit()  
}

exit {  
 set current pixel to the top left  
 loop:  
 paint each row red

set current pixel to the top left  
 loop:  
 paint each row orange

set current pixel to the top left  
 loop:  
 paint each row yellow

set current pixel to the top left  
 loop:  
 paint each row green

set current pixel to the top left  
 loop:  
 paint each row blue

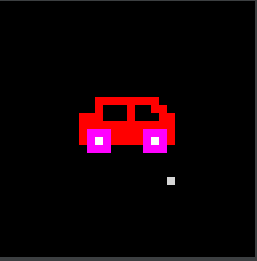
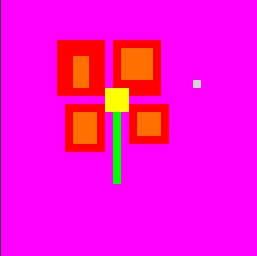
set current pixel to the top left  
 loop:  
 paint each row purple

set current pixel to the top left  
 loop:  
 paint each row magenta  
 print out the exit message  
 close the program  
 }

backgroundChange {  
 save the current location of the pixel  
 set a limit so the painting loop stops to 1024  
 if current color is red  
 print out red background message  
 if current color is blue  
 print out blue background message  
 if current color is green  
 print out green background message  
 etc…  
 set the current pixel to the top left  
 loop till limit:  
 paint each row the current color  
 restore the original pixel location  
 }

resetCanvas {  
 set limit so the painting loop stops to 1024  
 load in black as the current color  
 set the current pixel to the top left  
 backgroundChange()  
 centerPixel()  
 prints reset message  
 }  
 centerPixel {  
 finds the center pixel on the bitmap  
 replace the current pixel coordinates with the new ones  
 prints center pixel message  
 }

**SCREENSHOTS:**

**WARNINGS:** Changing the background color erases your current drawing