

Program Final Graphics Animation

Honors Computer Science C++

Mr. Clausen

PROGRAM Final Graphics Animation (100 points with a possibility of up to 20 extra credit points)

This Program must be rated "G": no violence, no controlled substances, no sex (gender is OK), no profanity, etc.

Start by drawing your "background scenes" on the graphics grid sheet. **Remember to keep the area of the background that you wish to have your animated object move over to be only one solid color.**

Once you have your backgrounds ready, it's time to add animation to your project. Remember that animation is drawing an object in one position, erasing it (drawing it in the background color at that position), and then redrawing the object in a new position (avoid using `cleardevice()` in the animation process). This works best if you use variables for every point in the object you wish to move across the screen and a "loop to move the object. Think of this as a "mini movie", so plan on having 3 different background scenes with animation occurring in each scene. You will need to use the `cleardevice();` function to change from one background scene to the next. The object that you are animating cannot be a simple shape that is a part of Borland's predefined shapes (circle, ellipse, rectangle, etc.) unless you add several details to the shape to create a new object.

This is your final project, so the effort that you put into this program should reflect that fact, and will be reflected in your grade. Make sure you have a separate function for each and every part of your graphic and animation, complete with Pre and Post conditions in the function implementations, and the 5 lines of comments for each function declaration as illustrated below. Remember, Pre-conditions tell what must be true before the function is called and Post-conditions essentially describe what is true or has occurred after the function has executed.

```
//Function: Type the function name here
//Type a brief description of what the function does here
//
//Input: list the variables that are passed into the function here
//Output: list the variables that are returned from the function here
```

Don't forget to use descriptive variable names for self-documenting code, global symbolic constants rather than local numeric constants, and use comments freely to explain your program.

Information about graphics can be found at the end of every chapter in our textbook. In particular Chapter 6 in our text (Page 340) begins the discussion about animation. There is also a nice summary of Borland's graphics commands in Appendix 5 of our textbook, pages A.16 - A.18. You can also find lectures about graphics and animation on my website where all the lectures for the entire year are located.

If you choose to do interactive graphics, you can earn up to 20 extra credit points, as long as it is true interaction (as in a game) that controls the object being animated. If you use interactive graphics, your game must start with a text screen menu, which gives directions on how to use your program (specifically indicating which keys control the interaction, and which key QUITs your program). If I press the key that is supposed to quit your program, then the program should actually quit. I should not have to press this key more than once to exit your program.

First very important note:

When you use `initgraph` and tell the compiler where Borland's `bgi` module is **use:** `"c:\\bc5\\bgi"` an absolute directory reference, instead of relative directory reference (what the book uses... `"c:..\\bgi"`) I will deduct 10 points off of anyone who does not make this change.

Instead of using this command to set the graphics mode:

```
int graphdriver = DETECT, graphmode;
```

You must use this graphics mode instead:

```
int graphdriver = VGA, graphmode= VGAHI;
```

Second important note: This program should be written in C++ entirely, and use Borland's `bgi`, no assembly code allowed, no information or routines downloaded from the Internet are allowed either. I will submit your source code to "TurnItIn.com" to verify that your code is uniquely yours as opposed to using code from the Internet or from another student in class. If you choose to create (or recreate) a game, no other student in our class can write the same game for their final project.

This is part of your "hands on" final exam, so be prepared to present your program to the class. Be ready to explain the algorithms that you used, and why you chose those particular algorithms. Be prepared to answer other questions that may be asked on any and every line of code. Make sure that you follow the programming practices that we have learned in class all year and don't use topics in C++ that were not covered in class without prior approval from your instructor.