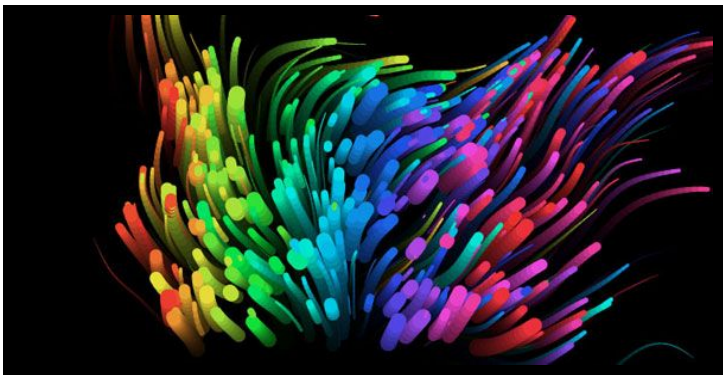
AP Computer Science 2018!
Final Project! May 20-June 7
Description, Milestones, and Process



It's over! Time to make your code come alive. 2D Arrays become game boards, recursion becomes code art that replicates natural forms, logic gets put to work in a way that contributes to the social good of the community. The possibilities are endless and the opportunity is priceless. Do not squander it.

Unlike a focus on a test, this is a focus on a project. Both skills are important in the software world. Here, you may choose a path that leads you to something meaningful both for you and the RHS community. You must also work with a partner who has chosen the same kind of project/environment. You can make independent projects, but you are also expected to work as "thought partners" to help each other along the way. With that in mind, consider developing a project for the last half of the trimester that will fulfill all requirements below.

We will be working in Java using *Processing* as the IDE. *Processing* facilitates the use of graphics and animation all using code that you studied very hard this year. You must log your progress each week and make a plan at the end of each week. Your grade will be determined both by lab time and by the product you produce and share at the end of the trimester.

There will be ample support available on the Moodle and as has been the case most of the year, you should view a few videos or read a bit before coming into the lab to try it out. I will also be available to help. You may work with a partner on this or alone.

To use Processing, save your classes to the desktop. When necessary, simply drag and drop your folders to GitHub…HOWEVER…you will first need to create a new branch!!

The Proposal

The first step in a project is to propose a system. You must write and submit a proposal and present the proposed system to the class. In all cases, your project must be *TV Worthy*.

.

Both the written and oral proposals should contain very little technical information. They simply explain what the project will do when it is completed. The proposal might specify languages used, or details such as that the project runs on two computers connected across a network, or that the project will run as an applet. It should not include any technical words or class names such as, "My project uses a class that extends JPanel." The proposal must include at least the sections listed below, and may contain others as is appropriate.

- Introduction

This is kind of like an abstract. In a few sentences, and in layman's terms, explain what this project will be when it is completed.

- Description

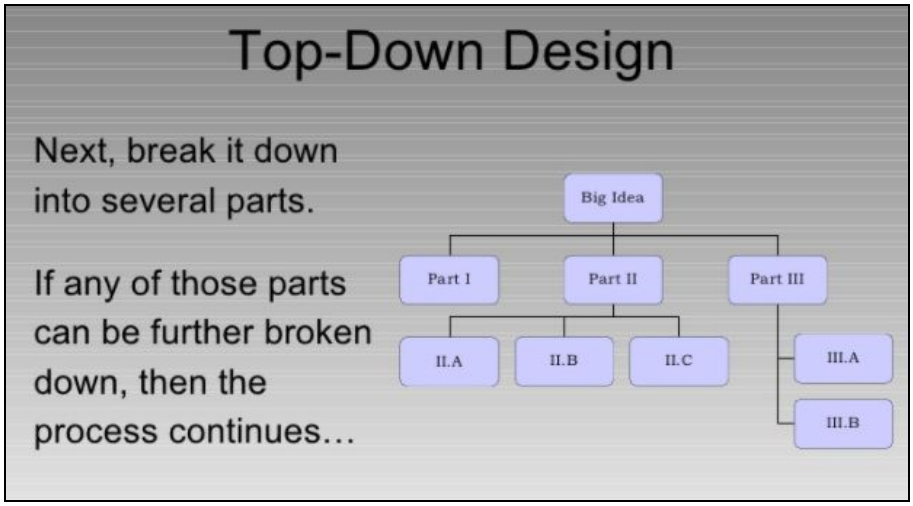
In more detail, describe the functionality of the system. Refer to prototypes where useful. Prototypes may be embedded in description text. Explain what the system will do, not how it will be done.

- Milestones (Weekly Goals)

Break the project into several smaller parts that can be completed and tested individually. Give a timeline with estimated milestone completion dates.

- Prototypes and Top Down Design Chart

Include a series of color pictures of screens that will be seen by users. These pictures may be incorporated into the other sections, or appended. If there are controls, specify how they work.



Your proposal will be graded based on content, quality, and writing (including grammar and punctuation). The goal is to produce a professional quality document that gives the reader a clear understanding of what you're proposing.

There will be ample support available on the Schoology and as has been the case most of the year, you should view a few or read a bit before coming into the lab to try it out. I will also be available to help. You may work with a partner on this or alone.

The Process

During this process, which will be monitored daily, we will acknowledge a “flipped classroom” environment. In other words, before you come to class each day, you should have reviewed the resources on the Moodle to facilitate better use of class time.

You will also need to complete a series of “MileStones”

1. Choose a team, set up your repo and logs on Github,
2. Begin research
3. Develop a project description including a rough design in graphic form
4. Develop the project
5. Present clearly

AboutTheLog

Every week you must write a few sentences about what you (and, I mean you yourself, not other people in the team) did for the project. I expect that some weeks you will do less for the project, than other weeks, but you must demonstrate progress.

The goals for this Personal Log are:

- To allow me, the "manager", to keep track of what each one of you is doing.
- To facilitate coordination among your group. Since you can read all members' log files you can quickly check on who is doing/did what.
- Your personal log will be a single Wiki page on the wiki pages of your group project. All you need is the date, followed by a couple of sentences on what you worked on that week. Be specific.
- And, write the most recent entry first, that way you and I don't have to scroll so much.

The Grade

Here is how you will be graded:
Product (3 pts each)
_____Quality materials packed and turned in on github and a flash drive. Includes a README file (see the calendar)

_____Used, challenged, and advanced your coding skills

_____Demonstrated knowledge of code beyond a cursory level

_____Difficulty of attempted project, uses programming constructs and environment appropriately, project reaches beyond a one-week or brief assignment.

_____/12

Process
_____Classroom process/preparedness with in-class focused effort and daily sustained interested and inquiry on this topic (9 pts)

_____Weekly checkups/logs with me. You seek me out, I do not seek you. (3 pts)

_____Ability and willingness to problem solve complexity along with willingness to assist each other and others. (3 pts)

_____Met projected milestones and scope and submitted a proposal here: https://goo.gl/2XOxFx and a final link here https://tinyurl.com/y6h88aea (3 pts)

_____/18

_____/30 Total

The Reflection

AP Computer Science Reflection

Name _____Emmett Durigan_____
Date _____

1. Write a short paragraph describing your project. This should include the purpose of the program as well as how it works.

My program is supposed to be flappy bird. The bird or circle in my program is supposed to jump through the lines. If it touches a line you have to start over. Your score is on the top left.

2. Describe the section of the program for which you had the main responsibility and briefly describe your successes and difficulties. How did you overcome these problems?

I did the whole thing. The hardest parts for me was getting it so it knew when it hit a line. The other hardest part was getting the score to count up. It took a lot of time and i searched some of it uo but i got it to work.

3. Write a short paragraph describing your group. Topics to discuss include but are not limited to: 1) Group dynamics; 2) Group "leader"; 3) Give each member of your group a participation and effort percentage. The total percentage will equal 100%. Include your own effort; 4) Include any other information about your group that will help me award each member an equitable grade for the project.

4. Reflection: On the back of this sheet, write a brief paragraph or two reflecting your year in AP Computer Science. Include things you enjoyed about the course, things that you would like to see revised, how well prepared for the AP exam you

were, your greatest achievement, and your feelings about this final project. You may frame it as a letter to next year's incoming AP students.

My year in Ap Comp sci was good. I enjoyed this class the most out of my classes. I was well prepared for most the ap exam. I liked the final project, we got to use everything we learned to make a final project.

Name(s)
Period:

As part of your final project submission, create a "README" file and, in 500 or fewer words, include the following:

1. Identify the purpose of your program. (Approximately 150 words)
   It moves down the tree. As it moves down the tree it looks right and left. It looks for leaves and when it finds leaves it records it.
2. Describe the difficulties and/or opportunities you encountered and how they were resolved or incorporated. In your description, clearly indicate whether the development described was collaborative or independent. (Approximately 150 words)
3. Format your code. Then capture and paste the most significant piece of code into a document and describe how the algorithm functions. Include comments or citations for program code that have been written by someone else. (Approximately 200 words)
4. Turn in your final project on a USB drive with this "README" included. Be sure to print another copy to turn in the old fashioned way.