

(20 pts) Approx. 3 days

At this point in our AP Computer Science class, we know pretty much 95% of the content we need to do some really big, awesome things with our own computer programs! We started with control structures (if, while, for), and added data types (int, double, arrays, Strings, ArrayLists), covered object oriented design (classes, objects, methods, variables), and now have talked about the details of class structure and organization (inheritance & polymorphism).

With all those ideas in mind, it's time to build the framework for *the most awesomest video game ever!* Think about huge video games that you've played or seen: World of Warcraft, Minecraft, Halo, Skyrim, Call of Duty, Fortnite, Overwatch. Each of those video games has a massive program with a huge class hierarchy behind it. In this last part of the unit, you get to design your own video game class hierarchy and structure to illustrate your knowledge of inheritance and polymorphism

1. Use your engineering notebook to brainstorm and plan out the different games you could create, and how you want to organize your class structure once you've chosen a game! Do some brainstorming and planning before you start programming!
2. VIDEO GAME DESIGN: Make a new project in BlueJ (so you've got a nice blank work space). Create your class structure using proper inheritance and polymorphism to illustrate how everything would be related. Make sure that your design fits the following criteria:
 - a. Your hierarchy should have at least 4 levels
 - b. Your hierarchy should have at least 15 classes in it
 - c. There should be a runner class separate from the main hierarchy
 - d. You should include at least 2 interfaces
 - e. You should include at least 2 abstract classes each with at least 1 abstract method
 - f. Your entire project should be able to compile without error
 - g. Your classes should all have at least 1 method in them that has a name that implies relevant functionality, but does not need to actually *do* anything
3. *Log 7: Inheritance & Polymorphism* – To wrap things up, take some time to write a full-page response to the ideas of inheritance and polymorphism. What do you think is coolest about those ideas? What's most confusing? What do the keywords **extends**, **super**, **implements**, and **abstract** do for a program? How can you see inheritance and/or polymorphism work in other programs? What do YOU think the difference is between an abstract class and an interface?

Part 3: Tasks	5 points	4-3 points	2-1-0 points
<input checked="" type="checkbox"/> Video Game Design Brainstorm	+ You made a plan in your notebook for how to organize your game structure + Your brainstorming is recorded	- You wrote less than a page - Your notes do not outline a coherent plan	- There is no plan for your program
	15-10 points	9-5 points	4-0 points
<input checked="" type="checkbox"/> Video Game Design Class Framework	+ You completed your entire game design framework + Your framework meets all criteria listed above	- Your game design framework is mostly complete	- Your framework is incomplete
	10-8 points	7-4 points	3-0 points
<input checked="" type="checkbox"/> Log 7: Inheritance & Polymorphism	+ You wrote a complete page in your engineering notebook	- You wrote less than a full page	- You wrote less than half a page