

An overview of any changes I made to my class design after submitting Project 2A, as well as my reasoning for these changes

When I submitted project 2A and got to see the sample solution, I realised how flawed my design pattern was. Previously all I had was simply every class connected to Sprite and for certain classes I added an interface for them. I was confused on the use of them and this made me make interfaces even if one class only used it. This design made little use of the functionality of inheritance, abstract classes and the use of overriding or overloading.

This is why I decided to rethink through the design process with the sample solution for 2A as a base. I tried to mostly follow the sample solution in implementing the design and I realised how much more logical it was and better in terms of the ability of polymorphism where I could freely reuse or change the methods that the parent classes had.

Although some parts of my code is very messy I found it to be better than calling multiple for loops to find each different sprite at a given position and reinitializing them/casting them to the type.

Difficulties I had during this project

I had an difficult time when I was first starting out on the project, I didn't know how to exactly get the coordinates of the different classes. I first tried creating a new object of another class in each sprite when I wanted to access another classes coordinates but it obviously didn't make sense. It was only after some thinking that I found out I could use the array list of sprites to access each of the classes.

Similarly, due to unfamiliarity of doing object oriented programing I had confusion of how to properly overload or overwrite in certain classes to make sure that the objects move correctly. On that note, I had a lot of trouble when certain blocks were stacking onto each other when they were not supposed to or when my player was going into blocks.

The undoing of blocks was also a weird one for me, coming from two previous subjects which always asked me to optimize the speed of the program, I unnecessarily kept trying to find different ways to make it such that perhaps I only needed to store to history a single block when it is moved rather than storing every other movable block on each move.

How slick is as a program is also very intriguing and what it was able to do was very weird at first and I found it hard to understand how the update and rendering worked, since we were working with multiple nested loops and different loops running at the same time.

I am unsure how I should be initialising and calling for each object neatly, with the use of an array list, it was hard for me to think of a way to go around needing to loop through the whole array to perhaps only return a single sprite.

One key piece of knowledge I learned completing this project

The ability to have clean and reusable code with polymorphism which are easy to access and understand in object oriented programming. Without the use of inheritance I would assume it would be almost impossible or much harder to create a game in any other language which doesn't have this. This is a very useful tool to be able to use as the usage of multiple classes is much easier to understand as it will be very clear which method it will be using from the parent class.

Anything I would do differently if I did a similar project

I would think through and make methods which could be reused in multiple instances of subclasses my top priority first, rather than first tunnelling and only thinking of how to solve each problem for a specific case and thus needing to rewrite code when more stuff need to be implemented.

I definitely would start much earlier on the project as well, especially since making a game takes a lot more time than expected.

I would look at different stuff I could use in the standard java libraries to help assist me more and make life easier.

I would look at how someone designed a similar game on YouTube before jumping right into the project, this would really help in seeing the thought process and the common design amongst most games in terms of usage of sprites and the altering of them.

