Go, Erik Rupert GDSTRUCT Final Exam Essay

Algorithms are essentials for speedy programs. Without algorithms, we wouldn’t have fast devices, we wouldn’t even have optimized games. Our games would be very limited to low resolution textures, as we don’t have good algorithms to sort the big heavy texture files. That is why triple A games always have great optimizations. Because of fast algorithms, games have evolved into a realistic environment.

Having a big understanding of data structures and algorithms can give a higher range of games that you can create. You can even make your own data structure for the use of your own specific games. If you only know one data structure. It will limit the games that you can create. Even top studios will hire you if you create a really good data structure.

In the video “Fast Inverse Square Root — A Quake III Algorithm”, a programmer from id Software developed an algorithm that finds the square root of a number faster than the traditional method. I’m not going to explain how it works but only the use of it. When we calculate the light projections, the shadows, we don’t know it, but the computer calculates finds the square root of a number billions of times every second, just to render a scene. This algorithm, takes the normal method of just doing a sqrt, and made a new function / algorithm just for calculating a very efficient and fast method of finding the square root. This takes game development to a whole new level, as having a great understanding of math, and how algorithms work, it would greatly benefit anyone who wants to create a game, a very optimized one.

In relation to data structures, queues are a great structure to use in multiplayer games, although the modern matchmaking algorithm structures in multiplayer games are different, having the mastery of the simple queue can let you decide what extra features it can have, and how it queues other players based on their ranks.

Another one example of data structures is hashtables. Hashtables are a great way store data, as it is very accessible and has a really fast way of getting data since it is O(1) notation. The ideal usage of this would be a very big inventory system. If you were to use arrays for inventory system, it would take longer to find a certain item. An example of a game that could use, or is already using them is Minecraft, think about the creative inventory search bar, since the game has hundreds of items, it would really take a long time to find an item if its an array, especially if it’s a mudpack, that would take minutes to find.

Stack is a very simple data structure, yet it’s also very useful. Stack can store information of what you did recently. You can use it as an event list, say, for example, you did something in the day cycle, it adds the task to the stack, and once you end the day, the program counts the number of tasks you did in that day, and rewards you for it.

Dictionaries are another one of good structures to use in games, if the game has vast amount of items, and most of them relate to each other, a dictionary is a good way to find the relation between the two objects. For example, in Minecraft modpacks, there will be different overlapping types, one in a different, another in a different mod. If there would be a dictionary, like “Ore-Dictionary” mod that would tell you that the two objects are interchangeable, and can craft a common item.

References:  
<https://www.youtube.com/watch?v=p8u_k2LIZyo>

<http://enemyhideout.com/2016/05/games-101-data-structures-in-games/>

https://mcforge.readthedocs.io/en/1.12.x/utilities/oredictionary/