Discussion-2: Graphic Programming

Discussion Topic:

Delving inside a black box often reveals intricate algorithms and processes.
How does the black box concept allow us to focus on high-level functionalities and outcomes while abstracting away the underlying complexities?
Can you think of other domains where similar abstraction is used?

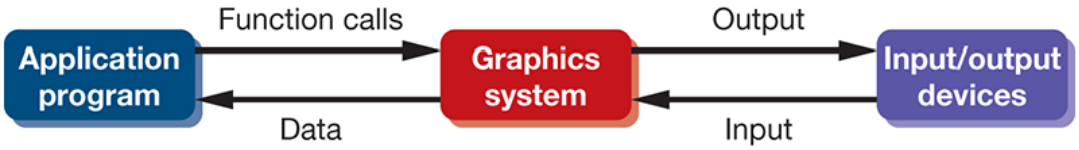
My Post:

Hello class,

The concept of a black box refers to a system in engineering that is characterized solely by its inputs and outputs, and we may not know its internal workings (Angel & Shreiner, 2020). In graphic programming, this concept can be defined as developers interacting with the graphics pipeline, through a graphical API such as WebGl and OpenGl by declaring specific inputs, such as vertex data and shaders, and receiving the rendered output, without needing to understand or interact with the processes handled internally by the operating system or the GPU, see Figure 1.

Figure 1

Graphics Pipeline as a Black Box



Note: From 2.3 Webgl Application Programming Interface, *Interactive Computer Graphics (8th edition)*, Figure 2.3 Graphic System as a Black Box by Angel and Shreiner (2020).

Notice that the flow chart from Figure 1 has three elements, application program, graphics system, and input or output devices. The flow chart can be described as follows:

- Function calls from application program to graphics system.
- Output from graphics system to input or output devices.
- Input from input or output devices to graphics system.
- Data from graphics system to application program.

This abstraction allows graphic programmers to focus on developing and optimizing visual outputs and user experiences rather than getting bogged down in the complexities of the graphics system architecture and processing details. In other words, application interfaces such as WebGL, OpenGL, and DirectX allow graphic programmers to create sophisticated graphics without worrying about the low-level operations managed by the operating system and the functionalities of the GPU, enabling them to concentrate on the creative aspects of their work.

Another domain where the black box concept is used is in Artificial Intelligence (AI), particularly in Deep Learning (DL). Just as in graphics programming with WebGL and OpenGL, DL developers interact with

APIs using Python libraries such TensorFlow and PyTorch , without needing to understand or interact with the processes handled internally by the operating system or the GPU. Furthermore, AI systems often function as black boxes themselves making decisions based on complex algorithms and vast amounts of data that are opaque to users. This is known as the “black box problem” in AI, where the inputs and outputs are clear, but the pathway that the AI took (thinking-reasoning) from input to output is not easily understood or visible, even to the developers who created the systems (Rawashdeh, 2023).

The black box concept is a powerful abstraction tool as shown by its application in graphic programming and DL development, enabling programmers and developers to be more efficient and to concentrate on the creative and engineering aspects of their work

-Alex

References:

Angel, E., & Shreiner, D. (2020). *Interactive computer graphics*. 8th edition. Pearson Education, Inc. ISBN: 9780135258262

Rawashdeh, S. (2023, March 6) *AI's mysterious ‘black box’ problem, explained.* University of Michigan-Dearborn. https://umdearborn.edu/news/ais-mysterious-black-box-problem-explained