Question

Should Game Developers commit to a Specific Engine or Toolset?

Abstract:

The issue of software port

Introduction:

This essay will be reviewing to see whether or not game developer should commit to a specific game engine or toolset when trying to develop games. This all stems from the issue that is software portability. In the field of computing the issue of portability has been around since the late 1950s [1]. Software can be classed as portable if it has the ability to be used on another system with efficient cost and effort [1]. The issue of portability is only being made more complex [235001]. Portability is a huge issue for game developers which can lead to a number of problems. For game developers portability is when you move from one engine to another “without having to redevelop or modify the game to suit that engine” [games.pdf]. There are lots of key issues with them trying to port to a different engine which I will look into. However there are benefits of being able to port games too from engine to engine but that comes with a cost.

Main Body:

Issues with Portability

There are many issues with portability especially with game developers. The first of these issues is if developers want to port a game from one engine to another there will be usually be a delay or an extra cost [6 3] which can lead to them wasting time / money and even losing out on money if it takes them long enough. This is bad as the aim of a company making software is generally to make money as well as make a game people enjoy. Although if done correctly it means they can get more sales as more people will be able to play their game [6 3] and research has shown that if implementations are done in multiple environments a program is likely to be more successful and easier to develop and maintain [6 3]. I believe this to be a good idea as game engines can become old or even discontinued [913] and if it’s on multiple systems it’d make it easier to continue building it without having to port it.

Another pressing issue with several game engines is that when they need to ported it’s a lot more difficult to do so because the logic has to “formatted in their proprietary format” [580][games.pdf]. This causes problems as the logic is the “core of the game” and where most of the game is stored [580]. Researchers have tried to separate it using “middleware between the logic and game engine” [580]. This approach has worked for smaller projects as when they compared a game they built through traditional methods and a game they built when the logic was separated and tested for thirty minutes the average fps drop was 11.69% [913]. If they continue trying to perfect this method and reduce the fps drop for larger games this would be an easy solution for porting games.