Final Synopsis in Software Studies

Object Orientation is a term that for the sake of simplicity points in two directions; inwards and outwards. Looking inwards we find a programming paradigm with roots in computer science and software development. Object-oriented programming shows us a certain way to approach the writing of code. The programmer will write the code in object-structured constellations, where the different lines of code can interface with one another, not through procedural representations, but instead through objects. Thus, an object becomes a passageway, of which different operations or procedures are forced to go through, in order to access the content that is encapsulated within an object. This benefits the programmer when developing a functional piece of software. However, once we look outwards we see the larger context in which object orientation positions itself. Object Orientation is also a methodology used for assimilating the complex structures of reality into neat ontological "objects". Though widely criticized as a purely ontological standpoint, elements of object orientation are still widely apparent in our world, and there are arguments for using OO as an epistemological tool. However, this focus still retains objects as a merely conceptual definition. Object orientation within software on the other hand realizes an underlying ontology of objects that places restrictions on the operations and processes in the development and usage of the program. Objects are no longer merely perceived abstractions, they are actualized as building blocks. When we then acknowledge how software is deeply intertwined with reality outside of computable processes, it becomes adamant to critically assess these structures and their resulting claims about reality.

This paints the landscape from which I take my own stance in terms of video games. Video games have a complicated ontology. Scholars and developers have approached video games from various angles such as narratology, ludology and software studies. This last field is very important to my paper, because I wish to position my own claims and conclusions in relation to the research that has been done in the field so far. With this paper, I seek to understand how object orientation in video games (here understood conceptually as the actualized structure of abstractive interfaces and not specifically the programming paradigm) can be further contextualized in light of established theoretical frameworks bridging the fields of Game Studies and Software Studies. Looking at my sources I believe these frameworks ultimately focus on how the computational processes are vital in contributing to the experience of a game, but also in developing games to be expressive (whether that be with ideological tenets or excellent authorship).

To provide any conclusive explanations for this topic, I have done a lot of reading, which has been very profitable. First of all, it is important that I understand the element of abstraction in software. In my paper this is especially concerned with objects, because the actualized objects in software opens up for comparisons to the way we perceive conceptual objects in our lives outside of computation. “Object Orientation” from the Software Studies Lexicon is my starting point, but it is essential that I both provide explanations of the inwards and outwards perspectives as I defined earlier, and why I can speak of object orientation without speaking of object-oriented programming. Secondly, once this is done I will attempt to outline the discrepancies in approaching the study of video games with a software-oriented theoretical framework. While my paper does not seek to specifically address Game Studies, it is important to recognize the different ontological standpoints in this paper. I come from the study of Software and I rely on related texts in the intersection of the two fields in order to propose my arguments.

Thirdly, with that in mind I come to the more undecided aspects of my paper. Having read a lot about the term *Operational Logics*, I will want to dedicate a lot of my paper to explaining this concept and its relation to object orientation. Operational logics describe a bridging of game studies and software studies. These logics explain abstract processes represented and communicated in a video game output with equivalent implemented computational processes located in the coding of the video game. Examples of operational logics are *collision detection* and *resource management*. There are more, and Wardrip-Fruin, Osborn and Mateas have been so kind as to define some of the most influential ones as it is very easy to mistake operational logic with game studies related terms such as mechanics or rules. Operational logics *necessitate* an underlying equivalent computational process. Collision detection is related to the graphical output of a game, but it does not speak of the narrative context whether that be bullets or tennis balls.

Implicitly implied here is also that I believe object orientation is one way to address the notion of operational logics, and I wish to do this – somehow. I think I will produce small segments of analysis based on different games. This will be intertwined with the established theoretical frameworks in my paper. Thus, I will switch between analytical observations and further theoretical comparisons. My combinatory views on object orientation and operational logics will also take on a speculative character using independent analysis to argue for my viewpoints. I am thinking of structuring my analytical observations in terms of some game design conventions. I can delve into the processes behind NPCs (non-personal characters) in computer games or the accessibility of player interaction on a computational level. I am not sure if I should try to find that *one* point of focus, and even if I attempt that I fear I may end up discarding important insights for my arguments by narrowing myself to one specific subject. I know that I find many critical aspects of OO in the processes behind AIs and player interaction and so they seem like good candidates. Perhaps I should just focus my analysis on one specific game in order to offer insight into the object-oriented approaches offered by that one game.

I have also read the book “Expressive Processing” by Noah Wardrip-Fruin, which has provided valuable insights into the bridging of software studies and video games by approaching understanding the processes and the data in the expressive output of a game rather than limiting oneself to the surface-layer of a game or any other digital fiction. There are also multiple great examples of authoritative expressiveness by using computational processes, and I can put OO into perspective by using this book and the notions of operational logics.

Furthermore, I will read some if not all of Ian Bogost’s book: “Unit Operations” and (so far) the first chapter of another of his books “Persuasive Games”. Bogost has contributed heavily to the discussion of operational logics, and in his books, he offers potentially valuable insights into the expressivity of *unit operations* – which are in some ways “just” operational logics made comparable to literature and poesy. I may abandon *Unit Operations* if it goes into another direction entirely, but the arguments brought forward in *Persuasive Games* are likely of high value for my paper. He argues that games express ideology, bias, assumption and a prevalent persuasiveness through the *procedural rhetoric* posed by computational processes. Regarding the critical side of Object Orientation as a methodology for understanding the world, it seems that arguments about procedural rhetoric is very relevant to bring to the table since games represent systems of rules and logics through object-oriented programming structures.