My senior project will explore how to best generate puzzle games that require path planning to solve. In particular, I am narrowing my focus to a subset of puzzle games that challenge the player to find a path from point A to point B. This path may not be readily available to the player and must be found by them activating a series of “switches.” These “switches” modify the game environment and, consequently, the paths available to navigate. The overarching goal of this project is to develop an algorithm which can automatically generate this type of puzzle game. To achieve this, I look to a specific path-puzzle game, called *Monument Valley*, for a basis to begin experimentation. Figure 1 provides a visual reference for *Monument Valley*. Note that in Figure 1 the goal is to find a path from the orange character in the bottom, point A, to the black square-shaped target in the top, point B.

Specifically, I will use Unity to build *Monument Valley* styled puzzle game levels and C# to program the “switches” and conduct the level generation.

*Monument Valley* provides a great foundation for this work because of its unique “switching” style. It portrays the game environment in an isometric perspective. The use of an isometric perspective lets the “switches” occur in unexpected ways. The player can unlock paths through Escher-like illusions in order to reach the level’s goal position. Because this project seeks to focus on path-based puzzle games that require planning to solve, *Monument Valley* is the ideal game style on which to base this work.
The research focus of this project is determining how to develop an abstract representation of the puzzle levels that can then be generated by an algorithm. The preliminary work in this endeavor will be in the re-creation of several levels from *Monument Valley*. With these constructed game environments I can compile an inventory of the different types of “switches” used in each level of the game. Once this inventory is produced, research into the types of generation algorithms will begin. There are several types of algorithms to consider for the purposes of level generation, namely, evolutionary search, grammars, test-and-reject, answer set programming, and wave-function collapse. Reading about and testing these algorithms will be the crux of the research component for this project. In addition to these, the final implementation and testing of the game will require the use of path planning algorithms such as Dijkstra’s algorithm and the A* variant as they are needed for actual game play.

The work of this project will be completed with the guidance of Professor Normoyle. We plan to compile our completed research into a “late-breaking work” submission to the Foundations in Digital Games Conference. For this paper our focus will center on the research and results we have gathered from our work with the level generating algorithms outlined above.

Over the course of completing this thesis research/project I hope to apply the skills I have gained, not only through completing my computer science course work, but also those that I honed within my visual studies courses. There is a large focus on learning new concepts, tools, and environments which enables me to expand my knowledge in an area of computer science that my coursework has not yet afforded me. In the process of such learning, I am also able to create a tangible, functional, aesthetically-minded final product.
Resource and Reading List


Wave Function Collapse algorithm:
- [https://github.com/mxgmn/WaveFunctionCollapse](https://github.com/mxgmn/WaveFunctionCollapse)


