Abstract: Balancing video game difficulty is a constant struggle to keeping the game challenging to maintain player interest but not overly challenging as to frustrate the player. While the player progresses through the game, their perception of too hard and too easy changes. Games often use a static difficulty curve to increase the difficulty as the player progresses through the game. However, dynamically adjusting the game difficulty creates a better fitting difficulty balancing, leading to a better player experience. To create such an experience, this work focuses on matching learning techniques decision trees and Fuzzy Inference Systems for the use of Dynamic Difficulty Adjustment. A decision tree learns how to evaluate a player’s performance based on in-game statistics and outputs a ranking. This ranking is then used by the Fuzzy Inference System, along with the player’s current health and damage output, to decide if the difficulty is increased, decreased, or left the same and to what degree it is changed. Therefore, a combination of decision trees and Fuzzy Inference Systems leads to a Dynamic Difficulty Adjustment.

Abstract: Facial modeling and animation play a crucial role in today’s society in the fields of simulation, communication, and digital entertainment. Realism and efficiency are the two factors holding us back from optimizing our models of the human face. Facial expressions play a key role in human society and are strongly linked to emotion. In this paper, we explore the different methods of modeling and animating human facial expressions, highlighting the problems and solutions that have been developed. We also investigate the biological and psychological properties of facial expressions to gain a better understanding of how to induce more realism into our models. Some current research and applications in the field are explored. We apply the theory by attempting to improve on Keith Waters’ 1987 facial model to improve realism, detailing the existing model and its drawbacks as well as our improvements.
Jacob Priest

Computer Science major

Robotic Navigation Using Image Analysis
(advised by Sofia Visa, Computer Science)

Abstract: The main focus of this independent study is robotic navigation within a maze. We look at Artificial Intelligence, robotic navigation, and image analysis to show how far robotics has come, especially in the last twenty years when hardware and software became both more capable and inexpensive. We use image recognition software, known as OpenCV (Open Computer Vision) and a Raspberry Pi mini-computer to provide the robot the capability of analyzing images and performing various functions to manipulate images to each experiment’s needs. This independent study covers five experiments testing the capabilities of both Open CV, Raspberry Pi and their various functions and practicalities.

Jacob London

Computer Science major

Show Me What I Want: Complexities in Creating an Effective User Interface
(advised by Denise Byrnes, Computer Science)

Abstract: User interfaces are everywhere, they dictate how humans are able to interact with machines and their effectiveness determines the effectiveness of the user. This project looks at the underlying principles associated with creating an effective user interface and applies these guideline to the creation of a user interface in a video game demonstration level. The interface is designed using Unreal Engine 4 and, along with Battlefield 4, a currently popular video game, is subject to a usability test with four testers of varying degrees of experience. This project also looks at the power software suites – video game engines, describes the design and implementation process of creating the user interface, as well as the usability tests, and discusses the results and implications of those tests. The reader will understand the principles that determine an effective user interface, the process for which usability testing is conducted, why Unreal Engine 4 was chosen for this project, and the difficulties associated with creating a user interface for different types of users.

Catherine Eckbold

Computer Science major

Are You Sure You’re in the Right Classroom? Research the Lack of Women in Computer Science
Workshop and Website Creation Designed to Encourage Women in Computer Science
(advised by Simon Gray, Computer Science)

Abstract: This I.S. project examines the fundamental disincentivizing reasons for the decline of women in the field of computer science. This thesis addresses this problem in three ways: (1) researching the reasons why women’s participation in the field of computer science is declining and presenting several ideas to address these reasons; (2) creating a workshop for middle school aged girls utilizing the research from (1), with an aim to spark and then maintain their interest in computer science; and (3) building a website to circulate the information gathered from (1) and (2) and to model and illustrate important data, facts and information pertinent to women in computer science.
Torger Miller

Computer Science major

On the Design of Decentralized Anonymity Networks
(advised by Rhys Price Jones, Computer Science)

Abstract: Tor, the popular onion-routing based anonymity network, has an outdated threat model that does not account for the increasing surveillance capabilities prevalent today. Although certain strategies, such as traffic mixing and traffic padding, are performance prohibitive when applied at scale to the entirety of an onion-routed anonymity network, the careful application of these techniques to specific types of traffic vulnerable to low-resource attack may allow dramatically improved security with a relatively small performance cost. The Tor rendezvous model, in particular, stands to gain substantially from targeted hardening against various low-resource traffic correlation attacks.

Sue Reon Kim

Computer Science major

Bringing Life to a Historical Drawing
(advised by Sofia Visa, Computer Science)

Abstract: This independent study investigates Maya’s capabilities, a computer graphics software. It discusses modeling, rigging, texturing, animation, rendering, and many other effects. It also explains how Maya’s script language (MEL) can be used to create customized computer generated graphics. In addition, it created a 3D animation based on a historic 2D Korean drawing, The Scenery of Dano Day, by Shin Yun-Bok. This involves several steps including: using Photoshop for base images; exporting these images to Maya for 3D modeling; generating a character; animating movement with Maya GUI and script language; rendering the scene; and video editing with Premiere Pro.

Tyler Catlin

Computer Science major

Predicting Stock Market Directionality Using Twitter and Neural Networks
(advised by Sofia Visa, Computer Science)

Abstract: In recent years, social media has become ubiquitous in social networking and content sharing. Yet the content that is generated from these websites remains largely untapped. This Independent Study investigates the usability of Twitter data for predicting trends of share price directionality in stock markets. In particular, it uses Tweets about Tesla and Boeing together with five learning algorithms (two of them being neural networks) to predict these trends. First, the data is pre-processed and analyzed for sentiment (i.e. each Tweet is labeled as positive or negative). Then, these labeled Tweets are used together with the actual stock prices in the five learning methods. The best accuracy is obtained using a cascading neural network with a windowing technique that incorporates the share prices and the sentiment data from the five days prior to the day being predicted. The researchers found a moderate correlation (62% correct prediction) between the model’s predictions of directionality and the true directionality of the stocks sampled. Better results could be achieved with better sentiment analysis and with larger datasets.
Sarah Williams

Computer Science major

From Fantasy to Virtual Reality: An Exploration of Modeling, Rigging and Animating Characters for Video Games

(advised by Denise Byrnes, Computer Science)

Abstract: In the last few decades, video games have quickly become one of the most popular forms of entertainment around the world. This can be linked to the improvement of computer systems and graphics which now allow for authentic and highly detailed computer generated characters. This project examines how these characters are modeled and developed. The examination of game characters entails a brief history of video games and their aesthetics. The foundations of character design are discussed and 3D modeling of a character is explored in detail. Finally, rigging or skeleton placement is investigated in order to animate the characters designed for this study. The result is two animated characters, which can be incorporated into several of the current and popular game engines. By the end to this paper, the reader should have a fundamental understanding of how a video game character is designed, modeled, rigged, and animated.

******************************************************************************************

Pratistha Bhandari

Computer Science major

3D Modeling and Scene Design for Games Using Unreal Engine 4

(advised by Denise Byrnes, Computer Science)

Abstract: This paper outlines the process of 3D scene design through the study of lighting, shading, and texture mapping. These factors are interlinked and can merge together in complex ways to create extraordinary video game scenes. The use of various illumination models, shading models, and mapping methods are discussed. For the software portion of this project, Blender and Unreal Engine 4 are used to create a semi-realistic fantasy forest scene. Capabilities and limitations of these tools are further examined.

******************************************************************************************

Dagmawi Zegeye

Computer Science major

An Exposition of Support Vector Machine Classification with a Purpose-built GUI for Exploring Some Commonly Used Kernels

(advised by Sofia Visa, Computer Science)

Abstract: We provide an exposition of the support vector machine classifier (SVMC) algorithm. We show that SVMCs produce a linear discriminant, specifically the maximal margin hyperplane and we detail how we can use the method of Lagrange multipliers and the Kuhn-Tucker conditions to find this maximal margin hyperplane. We also present some of the most popular kernels: the Gaussian radial basis function kernel, the inhomogeneous polynomial kernel, and the hyperbolic tangent kernel and discuss their advantages in the context of SVMCs. To provide an example of SVMCs in use, we perform a series of experiments on the Titanic data set, made available by Kaggle. Finally, we present SVMC Visualizer, a web-based graphical user interface we developed in the R programming language, that can be used for experimenting with SVMCs under various feature maps and kernels on user specified datasets. We developed SVMC Visualizer with the intention that it aids users in better understanding the support vector machine classification.
**Peter Nelson**  
*Computer Science major*  
**Needles and Haystack: A Computational Approach to Finding Subalgebras of the Symmetric Group**  
(advised by Sofia Visa, Computer Science)

Abstract: In abstract algebra, the symmetric group $S_n$ on a finite set $X$ (containing $n$ elements) is the set of all possible mappings from $X$ to itself, and whose group operation is function composition. In practice, $S_n = \text{all possible permutations of set } X$’s $n$ distinct elements; thus the magnitude of $S_n$ is $n!$. We can partition $S_n$ into one or more parts (also known as blocks or cells); that is to say, we can divide $S_n$ into a union of overlapping, non-empty subsets; these blocks form a basis for $S_n$. We can also say that they form a subalgebra of $S_n$ if any combination of these parts via multiplication produces a linear combination of the blocks formed by the initial partitioning. This paper seeks to find out, for any given $n$, how many of the potential Bell number of ways to partition $S_n$ actually yield a working subalgebra. Once all of the subalgebras for $n = 1, 2, \ldots$ are found, the intention is to try to discern if there are any mathematical patterns and/or categories that these partitions fall into. The number of possible ways to partition $S_n$ increases dramatically as $n$ increases, meaning that even for $n = 4$ we are already looking at a 15-digit number of possible ways to divide $S_n$. This means that checking all possible partitions of $S_n$ via brute force would be prohibitively inefficient. Instead, this paper will attempt to find all possible subalgebras via several different approaches, including Monte Carlo simulations, as well as examining subsets of prohibitively large search spaces.

******************************************************************************************

**William Rial**  
*Computer Science and Classical Studies double major*  
**Veni Vidi Didici: A Video Game for Learning Latin**  
(advised by Sofia Visa, Computer Science, and Monica Florence, Classical Studies)

Abstract: Learning a language is a difficult task and often requires many years of dedicated practice to master. Thus, many have attempted to find other ways to learn a language. One possible solution is through games. J.P. Gee writes that playing a video game actually teaches a variety of skills based on the challenges presented. Websites like Dueling reward the user for practicing a language and offer greater prizes for regular participation. Some other attempts emphasize fun but neglect the pedagogical aspect. This project attempts to create a video game that is both educational and engaging. The game, called *Veni Vidi Didici*, has two distinct parts: a translation game inspired by Dueling, and an interactive game where the user plays as a Roman legionary in an episode from Caesar’s *De Bello Gallico*. In part one, we explain why choosing Caesar and his narrative about the Gallic war benefits eager students. Then we examine Caesar’s life up to the Gallic War and his habits as a general before we consider his historiography and its implications. In part two, we discuss the Unity Game Engine and the process for creating the game.