

Computer Science at BGS

Mr Osborne, Subject Lead for Computer Science

It has been a busy few months for the Year 13 Computer Science cohort, busily putting the finishing touches to their Computer Science NEA (Non-exam assessment) projects. The students' work is impressive, with some of the submissions well beyond the standard required of the A-Level specification - often the product of substantial numbers of hours work planning, researching and coding to produce such polished projects. A showcase of some of the very best work is included below.

BGS PATHFINDING

Adhish Sunishkumar

A few snapshots of the work Adhish created on 'BGS PATHFINDING' an app to help students at BGS navigate their way around the newly implemented 'one-way system' and to formulate the shortest path to their next class using the A* pathfinding algorithm.

After researching the many different PFA, I've decided to implement the A* algorithm into my solution for many reasons:

Reason	Explanation
Efficiency	As A* utilises a heuristic estimate to end point, as well as the cost from start to current node (more on this later), the algorithm decides to explore nodes that are more likely to lead to the goal in a shorter distance. This outperforms exhaustive algorithms such as Dijkstra and DFS.
Optimality	A* guarantees to find the shortest path from start to end if heuristic function isn't overestimating the true cost.
Flexible/Customisable	You can tweak the heuristic function to prioritise speed over accuracy or vice versa.

Comparing A* vs other algorithms:

Name	Optimal Path?	Time Complexity? (best case) *	Space Complexity?	Can Handle Weights?	Negative Weights?
A*	Yes	O(V+E)	O(V)	Yes	No
Dijkstra	Yes	O((V+E) + log V)	O(V)	Yes	No
BFS	Yes	O(V+E)	O(V)	No	No
DFS	No	O(V+E)	O(V)	Yes	No

*V = Number of Vertices
E = Number of Edges

As you can see, Euclidean Distances are usually shorter than Manhattan, in fact, Manhattan distances are always greater than or equal to Euclidean Distances. Here's why...

Manhattan Distance is calculated like this:

$$d = |x_1 - x_2| + |y_1 - y_2|$$

Euclidean Distance is calculated like this:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

We are trying to prove:

$$|x_1 - x_2| + |y_1 - y_2| \geq \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Square both sides to get rid of square root:

$$(|x_1 + x_2| + |y_1 + y_2|)^2 \geq (x_2 - x_1)^2 + (y_2 - y_1)^2$$

Expand the brackets, collect and simplify like terms:

$$2|x_2 - x_1||y_2 - y_1| \geq 0$$

A* Pathfinding Algorithm

Controls:

- Left click to place Start & End points
- Right click to reset Start & End points
- Press SPACE to start pathfinding
- Black squares are obstacles
- Blue path is the shortest route

Current Floor: 7 End Floor: 7

Paul Tol's colour schemes have widely been recognised as the most colour blind friendly. Some of the key takeaways I have found from reading about his work are as follows:

- Avoid red (especially with green). Red-Green colour-blindness is the most popular
- Pursue contrast. Universally distinguishable colour pairs ensure ALL colour blindness is accounted for.
- The safest schemes are bright, contrast, vibrant, muted and black and white.

Figure 1: BGS PATHFINDING - Adhish Sunishkumar

Nature in Motion, a predator-prey simulation

Atiksh Agnihotri

Here Atiksh created a tool for biology teachers to help them teach the dynamics of the predator-prey interaction. This project explores predator-prey dynamics, which have been modelled using digital creatures on a grid and producing graphs using the Lotka-Volterra differential equations.

4/29/2025

Nature in Motion

A predator-prey simulation

ATIKSH AGNIHOTRI
BOURNE GRAMMAR SCHOOL

Abstract

Rationale
Biology students are required to learn animal population dynamics that take place over prolonged timeframes, on an enormous scale. They can learn about these systems theoretically, but lack the means to fully comprehend these complex mechanisms. There are limited simulation tools available to help biology teachers bring these concepts to life for their students.

The purpose of this project is to create a tool (customisable simulation/model) for biology teachers to help them teach the dynamics of the predator-prey interaction by using a 'visual' teaching aid to support learning.

Methodology
To develop the simulation, existing models were evaluated along with capturing user requirements through a questionnaire and end-user interviews. Prototypes were used to inform the development of the final solution. The design of the project provides an overview of how the different parts of the solution will interact. This includes algorithms, flowcharts, graphs and data tables.

Solution
A simulation was developed in Python, using digital creatures to recreate a natural environment and provide a graphical representation of the interaction between predators and prey. The simulation allows users to play around with different variables to model various scenarios that occur in the natural world.

This system uses the Lotka-Volterra differential equations to model the observed interaction between predators and prey in the simulation. A curve is produced, fitting the simulation data to the model's prediction for the population numbers of predators and prey.

End users (Teachers/tutors) can interact with the system by creating an account, where they can store previously generated figures and download the graphs generated, creating a physical repository for their students.

Key Words: Predator-Prey, Simulations, Algorithms

1.5. Modelling

1.5.1. Lotka-Volterra Model

Building on Mr P's prompt, this phenomenon of predator-prey systems was further researched to ascertain if it was possible to build this system. Predator-prey systems principally rely on mathematics to model the relationship, with many different equations being used to model them, such as the Rosenzweig-MacArthur model and Arditi-Ginzburg equations. Such systems use differential equations, equations which describe quantities over time, to model aspects of the predators and prey. The most famous of these is the Lotka-Volterra equations, which are a pair of equations that model the change in predator and prey populations over time according to several parameters.

$$\frac{dx}{dt} = \alpha x - \beta xy,$$

$$\frac{dy}{dt} = \delta xy - \gamma y,$$

Equation 1: Lotka-Volterra Differential Equations

Parameters.

dx/dt: This refers to the change in prey population over time. It represents the growth rate of the prey at the time

dy/dt: Similarly, this refers to the change in predator population over time. It represents the growth rate of predators at the time.

x: This is the current population level of the prey

y: This is the current population level of the predators

α: This is a constant parameter, which determines the growth rate of the prey

β: This is a constant parameter, describing the effect of the predators on the prey's death rate

γ: This is a constant parameter, which determines the death rate of the predators

δ: This is a constant parameter, which determines the effect of the prey on the predator's growth rate.

This model is one of the simplest that exists, but it is known to have good accuracy at predicting the general trend of the majority of actual ecosystems. It is also the model

1.5.4. FSM Models

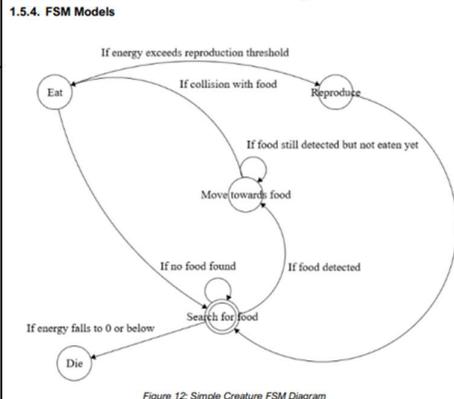


Figure 12: Simple Creature FSM Diagram

Simulation Screen

This is the simulation screen. It includes the simulation on the left of the screen, a timer at the top left corner and a graph, which is dynamically generated on the right. Creatures move on the grid, in discrete movements, which the user sees in real time.

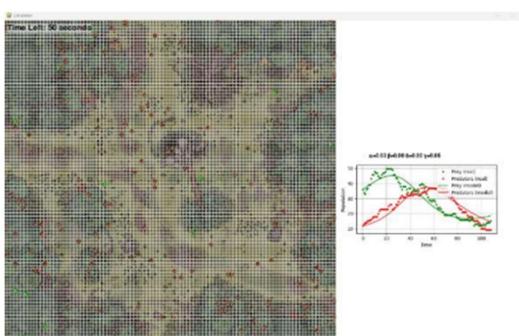


Figure 35: Simulation Screen

Figure 2: Nature in Motion, a predator-prey simulation - Atiksh Agnihotri

Census Data Presentation Program: Sexual Orientation

Courtney Pudney

For her project, Courtney decided upon some form of data visualisation software, opting to create a census dataset visualisation of sexual orientation within England and Wales. A difficult project that needed understanding of SQL, Pandas (a Python Data Analysis Library), TKinter and application of sorting algorithms such as 'merge sort'.

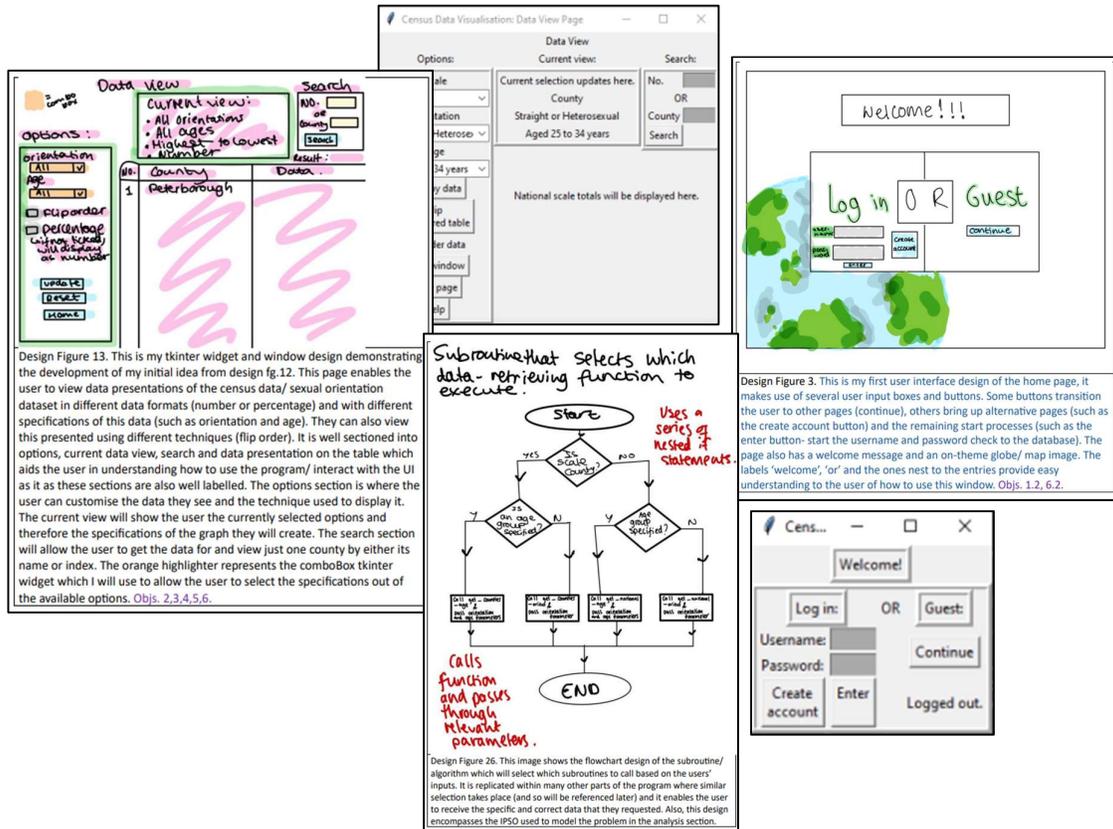


Figure 3: Census Data Presentation Program: Sexual Orientation - Courtney Pudney

I wish the entire group the very best of luck for their exams and sincerely hope that many of these students continue studying some form of Computer Science as they progress in their careers.