Numatic interview prep

**Who are Numatics competitors?**

According to ZoomInfo as of 24th April 2023, your top competitors are:

* Hako Machines Ltd
* Gtech
* Vax Ltd
* Kirby Corp
* AllBatteries
* Kaercher.com/us

With Kirby and Kaercher having a high revenue at $2.8B and $230.9M respectively in comparison to Numatics $196M

Source: <https://www.zoominfo.com/c/numatic-international-ltd/37455215>

**Who is the CEO?**

Chris Duncan

**When was Numatic created?**

Formally titled Numatic Engineering Limited was founded in 1969 as a British Manufacturer. The company is solely owned by the CEO, Chris Duncan who created the compact shape of the cleaner.

**What does Numatic do?**

As the UKs Number 1 Commercial Cleaning Equipment Manufacturer, you produce, -built-to-last- domestic, commercial, and industrial cleaning and maintenance equipment distributed world-wide.

One of the major producers of vacuum cleaners in the United Kingdom and produces the Henry domestic vacuum cleaner and its related models, which are canister designs having human names and smiling face appearance.

You offer professional tailored cleaning solutions for all sectors such as

* Healthcare
* Education
* Logistics and Handling
* Hospitality
* Construction
* Facilities Management

And you even go so far as to offer a 5-point Partnetship Pledge whereby as an approved supplier to the NHS, you ensure hospitals are provided with the best equipment to suit their cleaning requirements.

Partnership pledge:

* Reduced cost of Cleaning
* The sustainable solution
* NHS Approved Supplier
* Designed and Made in the UK
* Best-in-Class-Service

Tailored Cleaning solutions per sector - Healthcare

Furthermore, you offer professional equipment, configurable systems, and on-hand training for the advanced delivery of cleanliness, sanitation, and infection control (which I think is simply going above and beyond.).

**Cleaning Product Range**

Based on your product range I would say the only cleaning products you don’t yet have in your arsenal are products for space and underwater.

* Professional vacuum cleaners
  + Such as the Henry line including Hetty (which my mum has had one of for years)
* Cleaners with Advanced filtration and control of dust and hazardous substances
  + Such as Hazardous Vaccums (H-class equipment which is the legal requirement when working with high-risk non-explosive, hazardous dust and materials including asbestos)
  + L+M Class Vaccums (Which have advanced filtration and control of low to medium risk dusts and materials such as mixed dust, gypsum, silica, hard-wood and soft-wood)
  + Industrial Vaccums (specification equipment for a wide range of applications including boiler cleaning, floor cleaning, machine maintenance, high level cleaning and metalworking.)
* Cleaners with reliable and effective floorcare solutions
  + Floor machines (a variety of applications such as polishing, scrubbing, stripping or abrading floor surfaces)
  + Compact scrubber dryer (designed to address the question of size being exceptionally compact yet with performance equal to bigger machines)
  + Walk Behind Scrubber Dryers (The best of both worlds with compact design and a full width cleaning capability.)
  + Ride-On Cleaners (Designed to provide long, simple and reliable service day in, day out whatever the environment.)
  + NX1K (Scrubber dryer range that delivers leading cleaning results from your latest battery technology)

Lastly. Products that emphasise raising cleaning standards and productivity with versatility and quality.

* Mopping Range
* Carousel Range
* Multi-Matic Cleaning Trolley Range
* ECO-Matic Cleaning Trolley Range
* SERVO- Matic Cleaning Trolley Range
* PRO-Matic Cleaning Trolley Range
* NuKeeper Range and NuBag Range

**Nu-Design app**

You have an app that guides users through the product configuration process and enables the creation of a cleaning system that meets user needs precisely. The result being optimum cleaning productivity with the highest cleaning standards.

**Nu-Assist app**

An app that offers the benefit of online training, through a simple versatile and user-friendly solution, resulting in increased cleaning performance and productivity.

**Nu-AR app**

An app that bring cleaning solutions to life in any environment supporting Numatic customers to select the right tool for the job every time.

**Where are their factories?**

They have manufacturing facilities in Chard, Somerst, south-west England.

Four wholly owned international; distriibutors in France, Germany, the Netherlands and South Africa. Over 40% of the daily units (4,500 as of 2017) produced are exported.

SOURCE: <https://numatic.co.uk/products/versacare/>

DEMONSTRATE THE FOLLOWING:

Knowledge or Experience of:

SLAM / Path planning

* MSc using pose to get object location and waypoints to direct a Thorvald robot in a simulated grape field and enable autonomous navigation.
* Robot localisation (Odometry, Map based and GNSS)
  + Odometry – Dead reckoning (localisation based on the previous calculated pose and velocity estimation starting from 0, robot’s initial pose)
    - Odometry variants – Odometry can be visual, or laser-based . In visual odometry the velocity is estimates from the matching consecutive camera images (VPS on Tello drone which I have experience with).
    - All odometry solutions suffer from inherent drift problems. Statistical estimators like Kalman filters can reduce the drift but cannot completely eliminate it. Fusing information from multiple odometric sensors has a similar effect.
    - Kalman filter - a statistical estimator which filters the noisy measurement data, models measurement uncertainty and propagates that over time to exploit the knowledge about typical state changes (motion models.) It can fuse observations from multiple sources. It has a two step iterative procedure: predict where it propagates the state using a motion model and update: where it incorporates the measurements. The output is an averaged state together with its uncertainty.
    - Kalman Filter for robot localisation: The state corresponds to the robot’s pose. Observations are measurements from odometry, IMU, etc. Motion model is based on robot’s kinematics (e.g. differential drive)
  + Map based localisation – this uses a map of an environment as a references. The map can be an occupancy map (created with laser, 3D Cameras), visual landmarks or just geo-tagged images.

The sensor reading (laser, image) is then matched against the reference map: result is the most likely location which that reading originated from. An oppupancy map represents free and occupied parts of the environment and it is built from the 2D laser sensors, using SLAM (gmapping).

* Robot Navigation
* Can be split into Global planner, local planner and a recovery behaviour. A global planner is a type of high level navigation that finds the path from a start point to a goal using geometry. A local planner is a type of low level navigation that acts on the global plan and avoids obstacles using sensors(kinematics). A Recovery behaviour is only invoked if a robot gets stuck.
  + Global planner detailed (examples of global planner/navigation algorithms) - Computed before the robot starts moving Generates a series of waypoints for the local planner to follow.
  + Built-in types (BaseGlobalPlanner):
    - carrot\_planner - takes a goal point and attempts to move the robot as close to it as possible, even when that goal point is in an obstacle
    - navfn - uses a navigation function to compute a path for a robot
    - global\_planner - a fast, improved version of navfn
* Cost map: occupancy grid that represents safe places for a robot to be in. a global costmap is an amalgamation of a full static map + known obstacles.

Navigation algorithms

* A global planner, assuming a circular robot will operate on a grid costmap to find a minimum cost plan using a search algorithm. An example of a search algorithm would be Dijkstra which is utilised by thee navfn. The global planner is an improved version implementing A\* which incorporates heuristics.

Computer vision

* Image classification, pneumonia detection, skin lesion detection, face detection, object detection

Robotic related sensors and motion controllers

* Lidar and camera for computer vision on Thorvald and camera on tello drone, qibullet motion controller to program pepper bot to be able to point at the location of a card on a table based on a set of x,y coordinates from computer vision which are converted to grid format. Introduction to motion controllers in ROS

Software Development methodologies

* Agile – a software development methodology that is closely associated with object-oriented programming which utilises Languages such as Smalltalk, Lisp, Java and C#.
  + It is an iterative approach to project management and software development that helps teams deliver value to their customers faster. An agile team delivers work in small but consumable increments.
  + Scrum is an agile methodology that is adaptable. Fast, flexible and an effective agile framework designed to deliver value to the customer throughout the development of a project. The five principles of scrum are: Commitment, courage, focus, openness and respect.
  + Pair programming is an agile development technique originating from Extreme programming in which two developers team on one computer. The two people work together to design, code and test user stories. (A user story being an informal, general explanation of a software feature written from the perspective of the end user or customer. The purpose of a user story is to articulate how a piece of work will deliver a particular value back to the customer.) A common implementation of pair programming calls the programmer at the keyboard the driver, while the other is called the navigator. The navigator focuses on the overall direction of the programming. The collaboration between developers can be done in person or remotely.
* Spiral - a systems development lifecycle method used for risk management that combines the iterative development process model with elements of the Waterfall model. The spiral model is often used by software engineers and is favoured for large, expensive and complicated projects. Examples of spiral usage can be seen in Microsoft that used it to develop the early versions of Windows. Game development is another industry that uses spiral model to develop games.

Waterfall – a software development methodology that is a sequential development process that flows like a waterfall through all phases of a project. (Requirements, system design, implementation, testing, delivery/ development and Maintenance) with each phase completely wrapping up before the next phase begins.

**Unit testing** - Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation.

**Integration testing** - Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing is conducted to evaluate the compliance of a system or component with specified functional requirements.

**System testing** - System testing, also referred to as system-level testing or system integration testing, is the process in which a quality assurance (QA) team evaluates how the various components of an application interact together in the full, integrated system or application.

**Acceptance testing** - Acceptance testing is a quality assurance (QA) process that determines to what degree an application meets end users' approval. Depending on the organization, acceptance testing might take the form of beta testing, application testing, field testing or end-user testing

**Black box texting** - A method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied to virtually every level of software testing: unit, integration, system and acceptance.

**White box testing** - White box testing is a form of application testing that provides the tester with complete knowledge of the application being tested, including access to source code and design documents. This in-depth visibility makes it possible for white box testing to identify issues that are invisible to grey and black box testing.

**Field testing** – Field testing is a product development process for collecting usage data from potential customers before launch. This test type focuses on the unguided, natural, contextual usage of a product.

**End user testing/ User acceptance testing (UAT)** - User acceptance testing (UAT), also called application testing or end-user testing, is a phase of software development in which the software is tested in the real world by its intended audience.

**Object oriented vs Procedural Programming vs Functional**

**Procedural programming** is the use of code in a stepwise procedure to develop applications.

Text

Description automatically generated

For example, to develop a simple Bank Account App procedurally:

* Creating an account for an individual (account)
* Getting an account to deposit or withdraw funds (getAccount, deposit, withdraw)
* Transferring funds between two different accounts (transfer)
* Recording all changes that occur with all accounts (accounts)

**Object oriented programming** is the use of self-contained code objects to develop applications. In JavaScript, this can be achieved by creating a blueprint (class) for manufacturing objects.

Text

Description automatically generated

A **class** (BankAccount) encapsulates a set of properties (**constructor function**) and behavior (**class functions**deposit**,**withdraw**, and**transfer) that can be used to instantiate an **object**of specific values (i.e: let john = new BankAccount(“John”)). This is typically used to model real-world objects.

**Functional Programming** is the use of **pure** high-order functions to develop applications. This involves a focus on creating code that avoids changing state and mutating data. JavaScript methods such as map, filter, find, and reduce are built such that any data received by a program, functionally coded, will not mutate the original data. Also, the use of **closures**&**currying** makes functional programs simpler to implement and easier to read.

Text

Description automatically generated

Here, the data (origin) is not mutated during the creation of new accounts. Rather, a copy of it is created (slice) and merged with a new account (**spread operator**) into a new array (accounts). Accessing an account is also done without mutation (filter). Performance of transactions are done by creating an updated copy (map of accounts).

Unlike OOP, in functional programming there are no objects whose states are continually being updated (i.e.: no John or Joe objects). Rather, copies of an account or the list of accounts are copied, updated, and returned.