



Avant Zero

Random data-generating algorithm for (experimental)
Quake III Machinima (post)-production

FUNCTIONAL DESIGN REPORT

Table of Contents

| | |
|------------------------------------|-------|
| Chapter 1: Background | p. 6 |
| Chapter 2: Goals and Objectives | p. 10 |
| Chapter 3: User Roles | p. 13 |
| Chapter 4: Functional Requirements | p. 16 |
| Chapter 5: Limitations | p. 18 |
| References | p. 19 |
| Appendix A: Development Roadmap | p. 20 |

List of Terms and Abbreviations

Python

A versatile, high-level, general-purpose programming language; well known for its readability and expansive toolset for machine learning, scripting, algorithms, artificial intelligence, web development, stand-alone web applications and data analysis.

CLI

Abbreviation standing for '*Command Line Interface*'; a piece of software that requires text commands for command/software initialization and execution instead of a graphical interface (containing menu's, buttons and other graphical representations)

Algorithm

Software containing a specific and custom-made set of rules to get data such as media, text and the like from point A (input) towards a desired point B (output).

Machinima

Combination of the words 'machine' and 'cinema'; a popular artform for participatory media creation since the late 1990's/early 2000's. Literal meaning: the creation of digital films inside videogame engines (often by utilizing videogame assets, code adjustments, code execution and game modification programs)¹

¹ Oxford Dictionary (n.d.). *Machinima*. In *Oxford English Dictionary*. Retrieved July 15th, 2025, from https://www.oed.com/dictionary/machinima_n?tl=true

Avant Garde

Popular artform that develops new experimental concepts or techniques; often associated with the arts²

Dominion Algorithm

A random data generator for experimental Quake III machinima with limited functionality. Predecessor of the Avant algorithm. Initially created in late 2022 – early 2023 with the JavaScript language. Now considered legacy and defunct.

Proof of Concept

A version of specific software with limited functionality and scope to prove the feasibility of a concept or idea. AvantZero is considered a ‘proof of concept’ and only provides limited functionality compared to Avant – which is the full version of AvantZero - and currently, as of July 2025, still in active development.

Quake III Arena

Videogame released in December 1999 by game-developer *ID Software*³. Quake III Arena is a multiplayer-based first-person shooter, in which the player is able to compete (often during fast-paced gameplay) against computer-controlled enemies (bots) or other online players in arena-based levels (better known as ‘maps’) throughout a multitude of different game modes, such as, but not limited to: team deathmatch, 1 on 1 duels and Capture the Flag.

NLE

Abbreviation of: Non-Linear Editor; Software suite used for video-editing.

² Merriam-Webster. (n.d.). *Avant-garde*. In *Merriam-Webster.com dictionary*. Retrieved July 9, 2025, from <https://www.merriam-webster.com/dictionary/avant-garde>

³ Altar of Gaming (n.d.). *Quake III Arena*. Altar of Gaming. Retrieved July 10, 2025, from <https://altarofgaming.com/game/quake-iii-arena/>

GUI

Abbreviation of '*Graphical User Interface*'; a piece of software that contains a graphics-based interface with buttons, (colorful) text, navigable menu's and other forms of functionality for easy reference and execution of tasks.

Q3MME

Abbreviation of '*Quake III Movie Maker's Edition*'. A modification made for the game Quake III Arena by the users HMage, ent, auri and CaNaBiS. Q3MME allows the users to capture footage within the ID Tech 3 engine from recorded gameplay and turn them into video files or image sequences for use in machinima or game art projects. The mod contains several advanced functions such as Multi-View support (allowing to see multiple point of views at once), Depth of Field generation, custom export parameters and the implementation of motion blur.

Chapter 1 – Background

AvantZero is a Python-based tool designed to simplify and streamline the creation of randomized experimental machinima films, with an emphasis placed on Quake III machinima. Initially conceived as an experimental edit-decision randomizer with very basic functionality as the Dominion algorithm⁴ in 2023, it was written with Node.js and JavaScript over the course of a few weeks and initially released on NPM. Dominion was quickly considered obsolete and more of a pre-prototype for new, upcoming algorithms due to its severe lack of automation capabilities. It merely outputted randomized edit decision data, to which the editor should adhere to during their post-production process. At that point in time, the algorithm failed to automate anything for the user.

The Dominion algorithm was originally developed as a tool for my machinima label ‘A Pixelated Point of View’ and its main function was to randomly decide how several experimental and randomized compositions should be edited. In a sense, it should have been the missing link inside a collaborative and artistic process between human and machine. The main line of thought behind Dominion was that the algorithm would make final decisions regarding the visual aspect of the film, while the user was solely responsible for the auditory aspect of the film.

At its earliest stage of usability, the Dominion algorithm painfully pointed out that too much work still had to be conducted by hand. It did provide the user with data to adhere to during the post-production stage, yet the data wasn’t always clear enough and simply too extensive. Furthermore, each of the steps that had to be followed afterwards still had to be conducted manually; thus, sparking the idea for creating a better and polished version of the Dominion algorithm by placing emphasis on functionality, broader usability and automation while still embracing the core idea of randomized edit data.

It was at that moment in early 2024 that Avant was born: the successor to the Dominion algorithm. Avant’s main function is to generate random, experimental Quake III Arena compositions while automating much of the production process. It

⁴ Veenstra, J.P. (2024). Dominion Algorithm. *Bitbucket*.
<https://bitbucket.org/appov/dominion/src/main/>

can create all the required project files for the Quake III Movie Maker's Edition mod, a simple executable bat file to start the entire capturing process and finally generate XML/EDL files required for the NLE to properly read and export the randomized compositions to movie files.

Technically speaking, Avant allows for the generation of practically unlimited variations of potential experimental Quake III machinima films with a few mouse clicks and inputs. While the algorithm's core functionality revolves around Quake III machinima, the algorithm contains additional functionality to simply randomize existing footage captured from an unlimited number of games and/or video sources. Even footage from video cameras and phones; no matter what the user wishes to randomize at that point. If it contains a codec that is compatible with the algorithm, Avant will be able to randomize it and output the required files for use in NLE software.

Avant was named after the highly popular art movement '*avant-garde*', as it outputs raw, randomized and basically unchecked film data for the primary purpose of creating innovative, experimental and unconventional Quake III machinima films. It is not until the actual post-production process that the user is able to see how the generated data is visualized and captured. The films generated with avant focus on design, graphical representation, artistic expression and the idea of challenging traditional norms and practices found within Quake III movie production, and not necessarily on their contents nor story; thus staying in true '*avant-garde*' fashion.

An extensive list of additional forms of functionality has been planned for the future, such as an API containing extensive information regarding Quake III maps, such as thumbnails, spawn points and locations, map boundaries and an auto downloader function to automatically download and extract the required levels; or the implementation of machine learning: allowing the algorithm to embrace different editing styles based on the users' input, learning from its choices and allowing it to step away from the pseudo-randomized nature of its generated compositions if desired.

This is where AvantZero comes in. Considering the Avant algorithm will most likely take a few more years to complete, it was decided to create AvantZero: a proof of concept for the Avant algorithm. Avant remains in active development to this day and finds itself in early stages of development (as of July 2025: version 0.2.9), however it

must be noted that all resources will currently be allocated towards the completion of AvantZero before development on Avant will be resumed. The full Avant algorithm, in its current state, contains a few forms of functionality such as a fully tested and working main menu, a work in progress documentation website, a fully working project manager with CRUD (Create Read Update Delete) functionality, several functions that ensure proper initialization of the algorithm, a number of general utilities, and a few helper functions for the extraction of map-related data and .bsp files from a .pk3 archive.

AvantZero should be seen as the little brother of the Avant algorithm as it serves as a proof of concept: proof that the functionality that is envisioned for Avant really is able to work as intended. While AvantZero receives less functionality, it contains most of the core pieces of functionality that Avant should contain such as randomized compositions and depth of field map generation. Furthermore, it can output and visualize all generated data in PDF and/or CSV format for data analysts or enthusiasts.

Once development for AvantZero has been completed, its proof of concept serves a few purposes. First, AvantZero serves as a final submission for CS50X: Harvard's Introduction to Computer Science and Programming course; thus marking the completion of said course. Secondly, AvantZero will be used for the generation of a number of early 'avant-machinima', which serve for both artistic exploration and submission to festivals, galleries, museums and other exhibitions. Considering my personal background as a machinima artist and software developer, AvantZero allows for a highly interesting connection between cinema, code, art and machinima. Lastly, all functionality that has been successfully implemented within AvantZero could easily be added in the full version of Avant: possibly saving time during the development process.

This deep-layered approach (Dominion – AvantZero – Avant) illustrates a trajectory of design, technological refinement and deepening of concepts. Each stage serves a purpose in the vision to merge algorithmic processes with create co-authorship in the context of machinima. While limited, Dominion exposed the core functionality but also the core bottlenecks of early automation attempts. It laid the foundation for a more extensible system. Avant, while still in early development, reflects an ambitious take at creating a modular, intelligent and fully integrated piece of software that

aligns with efficiency and artistic unpredictability. AvantZero, in this larger context, operates as a technical manifest. It allows for entry into assisted forms of machinima filmmaking and ‘avant-machinima’, opening possibilities for students, researchers and artists.

The project, while rooted in Quake III Arena, opens a broader conversation around how experimental filmmaking can evolve through co-creation, partial automation, machine-based assistance and randomness. Avant and AvantZero position themselves as critical bridges between co-creativity and co-authorship. This shift not only challenges traditional notions of authorship but also invites a re-examination of the filmmaker’s role inside an automated creative process. Instead of controlling every frame, the filmmaker becomes a curator of unpredictability. They’re selecting, interpreting, and responding to algorithmically generated outputs. In this model, the algorithm is not merely a tool but a collaborator with its own logic and aesthetic tendencies as the algorithm outputs decisions concerning the visual aspect of the film, while the filmmaker will remain in control of the auditory scope. The line between author and assistant becomes blurred, resulting in a new hybrid form of authorship where intentionality and spontaneity coexist.

Chapter 2 – Goals and Objectives

With the algorithm's artistic and technical origins in mind, AvantZero aims to fully automate various tasks in Quake III machinima (post)production; tasks that would otherwise need to be done manually, while in tandem achieving specific creative and technical objectives:

Generation of unique, pseudo-randomized and experimental compositions

AvantZero enables users to create pseudo-randomized experimental compositions by reorganizing pre-existing footage in an algorithmic way that defies repetition. Each composition generated by the algorithm is unique (to most extent), even when using the exact same source material. This process of automatically generating new content reflects the algorithm's underlying commitment to avant-garde principles, such as: spontaneity, variation and non-linearity over traditional and narrative-based ways of filmmaking.

Rather than following predetermined editing patterns, AvantZero outputs randomized edit decision data (in formats such as XML and EDL) that can be imported directly into a non-linear editing (NLE) software suite. It acts as a generative and creative partner, proposing unexpected visuals and rhythms that challenge conventional filmmaking logic.

It aligns with experimental filmmaking traditions that focus on abstraction and viewer interpretation. By automating the visual structure and leaving the auditory domain to the user (such as sound design, audioscapes, music and/or dialogue), the algorithm introduces an highly interesting form of collaboration between human and code. Avant-based machinima is a continuously evolving body of work where no two outputs are identical.

Automating the assembly process of structurally diverse montages

In virtually every filmmaking workflow, the act of assembling and trimming raw footage remains a foundational (yet undoubtedly repetitive) process. This is no different in machinima. Creators are often tasked with selecting, sequencing and refining a vast array of gameplay clips and sit through large quantities of videoframes. It must be noted that AvantZero does not seek to replace the manual editing process; neither will it propose automation as a substitute for artistic expression. On the contrary, the manual curation of footage should be valued and preserved as rhythms, contrast and visual leitmotifs appear organically throughout the process. It is within this very hands-on manipulation that a film's unique style takes shape.

Rather than supplanting, AvantZero serves as a complementary pillar. It is a system of visual experimentation designed to challenge traditional editing conventions by primarily emphasizing focus on experimentation and randomized footage selection alongside the idea of embracing uncertainty and unpredictability. It avoids the familiar pacing found in traditional cinematic convention. It introduces a new audiovisual syntax that privileges experimentation over story, and variation over repetition.

AvantZero supports an exploratory mindset, where the algorithm is a collaborative tool in the editing suite. The user is able to adjust the proposed edits to their liking at any given time, or, if desired, immediately accept the given composition. The algorithm primarily serves as a provocative starting point for further refinement, disruption, reinterpretation and cinematic expression. In a sense, AvantZero both embraces and restructures the creative loop between intuition, tradition and experimentation.

Broadening the roles of the User as Post-Editor and Interpreter

By default during most film post-production processes, the user acts as the author. Within the scope of the AvantZero algorithm, the user gains an additional role: that of interpreter. The editorial agency is shifted from the user towards the algorithm, while preserving the user's ability to intervene, reinterpret and/or reject. By shifting the

roles, the process of editing transforms into a form of dialogue between human and machine; between suggestion and creative desire.

Multi-Format File Compatibility and Integrated Depthmap Generation

AvantZero was designed with flexibility in mind. Thanks to a multitude of external software packages implemented within the algorithm, it accepts both image sequences and traditional video formats, such as, but not limited to .jpg, .png, .tiff, .dpx (digital negatives), and H264, Apple ProRes, MXF, HEVC and AVI. Its' wide range of compatibility enables artists to work across a broader spectrum of visual material. While the codecs above exemplify a number of codecs that will enjoy support from the first initial version, it will certainly be possible that support for additional codecs will be added during a later stage (as minor updates considering AvantZero serves as a proof of concept).

AvantZero's multi-format file compatibility allows for deeper integration into diverse post-production pipelines and, in tandem with its xml/edl generation capabilities, for direct compatibility with a large number of NLE's as well.

In addition to its compositional functions, AvantZero contains the capability to generate frame-by-frame depth of field image-streams alongside (or standalone) its randomized video output thanks to the DepthAnything framework. These depth maps provide valuable metadata that can be used in a variety of technical workflows (3D compositing and generating depth of field masks) or for further experimental and artistic intent⁵.

⁵ Veenstra, J.P. (2020). DEPTHPMAP NEXT [Video]. YouTube.
https://www.youtube.com/watch?v=c2FT_a34vw4

Chapter 3 – User Roles

While AvantZero is considered a niche algorithm by trade and primarily serves as a proof of concept for a broader and updated version of the algorithm (named Avant) with much more added functionality; AvantZero, as a prototype, automates several tasks and generates a multitude of data streams that may be of interest for a number of user-bases, such as but not limited to:

- Machinima Artists willing to conduct experiments with Depth of Field maps within their workflow or create machinima solely based on Depth of Field maps;
- Machinima Artists wanting to generate randomized and experimental compositions of their project for either final use or demonstrational purposes;
- Video-Editors wanting to generate randomized compositions of existing footage (of any kind) for either final use or demonstrational purposes;
- Machinima Artists wanting to combine original footage and depth of field streams altogether in an experimental manner to create highly artistic point of views;
- Video-Editors and Machinima Artists in need of generating storyboards or timeline reports in PDF format of their projects;
- Machinima Artists, Programmers and/or Data Analysts who would like to view, analyze and visualize the generated randomized data (in CSV format).

The broad range of potential users for the AvantZero algorithm reflects the hybrid nature of the algorithm itself. It's embedded equally primarily focuses between artistic exploration and technical automation, while offering utility among a multitude of disciplines. It's core functionality lies in experimental machinima production, yet it offers additional capacity in the form of randomizing compositions, depth map generation and handling structured data. Because of its wide applicability, AvantZero is perfectly usable for both creative and analytical workflows.

Machinima artists now can utilize Depth of Field streams throughout several ways, and more importantly: throughout an setup-based automated process. For instance, the artist can export 'depth maps' alongside their footage for later use within their post-production pipeline, such as for the addition of manual depth of field, color grading effects or focus to a shot as demonstrated in earlier video demonstrations.⁶ Secondly, one could use the 'depth map stream' as primary footage to create exciting and experimental point of views. Lastly, the depth of field streams could provide valuable data for users interested in analyzing or visualizing the random generated data, or simply provide a low threshold to dive into experimental depth of field map use for the very first time.

The inclusion of automated PDF- and CSV report further broadens its appeal to users seeking documentation, visualization or post-production analysis of their projects and allows for randomized brainstorming or rapid prototyping. Though AvantZero is not intended to function as a final product, the algorithm has emerged as a versatile and exploratory production tool; one that merges creativity with automation and aesthetic control.

Looking forward, the upcoming Avant algorithm (the fully completed version of AvantZero) will significantly expand upon this foundation. Unlike AvantZero, Avant is built with full-scale film-data integration in mind, allowing it to actively analyze patterns through machine learning and artificial intelligence, generate corresponding depth maps and even autonomously capture machinima footage.

It includes native support for the game Quake III Arena through the automatic generation of full Q3MME (Quake III Movie Maker's Edition – a mod to capture

⁶ Veenstra, J.P. (2022). Quake III Arena: Experimental Depth Map use [Video]. YouTube. <https://www.youtube.com/watch?v=7cu76qMIWIQ>

video or images within the id Tech 3 engine) project files, including scripts, executables and project files; further bridging the gap between procedural data and hands-on filmmaking. AvantZero not only serves as a technical prototype, but also as a conceptual framework and steppingstone towards a more intelligent and deeply integrated machinima production platform.

Chapter 4 – Functional Requirements

In a nutshell, AvantZero should offer the following functionality upon its initial (version 1.0) release:

- Provide the user with an easy-to-navigate main menu;
- Embrace a project-based workflow. Users can create, edit and delete projects. The user can generate random compositions, depth of field streams or generated data on a project-basis and is able to choose whether they would like to execute a single area of functionality or combine freely between multiple functions at will;
- Contain a clear overview of potential errors and provide simple, jargon-free potential fixes for the user if an error were to be thrown by the algorithm;
- Provide a consistent, qualitative, easy-to-read and quick-to-start documentation website for both first-timers and experienced users with Python when they decide to use the AvantZero algorithm;
- Allow the user to create a single or multiple experimental edits of footage of any kind, with an emphasis on machinima. The user should be able to utilize both image sequences and video files alike and alongside one another;
- Allow the user to create depth of field map streams:
 - Create depth of field streams of the users' entire footage bin;
 - Create depth of field streams of the randomized sequences only;
- Provide the user with both XML and EDL files for use in popular NLE's, such as Adobe Premiere Pro, Blackmagic DaVinci Resolve or Avid Media Composer – to name a few;

- Generate a PDF containing table-based data on all randomized choices made by the AvantZero algorithm, including timecodes, filenames and thumbnails (somewhat in comparison of a watered-down dailies report);
- Generate a CSV file containing data on all randomized choices made by the AvantZero algorithm, ready for use with data analysis frameworks such as pandas, vanilla Python or Jupyter Notebook;
- Add visualized charts of said data into the table-based data PDF upon request;
- Offer a submenu that allows the user to select their function of choice after selecting a project;
- Offer a submenu with all listed dependencies, a quick how-to, links to the documentation website and further copyright/license information;
- Multiple ways to run the application, such as: with Python commands, compiled executables or as a package on the Python Package Index ‘PyPi’;
- And most importantly: a way for the user to gracefully terminate and exit out of the application correctly;

Chapter 5 – Limitations

Within its specified scope of functionality, AvantZero currently knows the following limitations:

- AvantZero can only create randomized compositions from existing footage to present proof of concept of the current ‘RNG’-based workflow, which encompasses the heart of both the Avant and AvantZero algorithm. AvantZero cannot capture randomized footage; functionality that Avant will have with the implementation of the Quake III Movie Maker’s Edition mod.
- The Avant algorithm will eventually have a deeper, more sophisticated menu structure than the AvantZero algorithm.
- AvantZero will most likely not receive any further updates after its initial version 1.0 release, except for small numbers of patches to ensure full functionality. Avant, on the other hand, will receive long-term minor and major updates.
- Avant will contain broader means of functionality, such as, but not limited to: artificial intelligence, machine learning, an full-fledged external API (current codename: TRIII – which is pronounced as ‘tree’) containing extensive information of each existing Quake III map (relieving the algorithm from these repetitive duties and thus speeding up the overall production process) such as spawn points, screenshots, general information and an auto map-downloader function.
- AvantZero should be seen as a ‘console-first’ application as it is developed first with the terminal in mind. A minimalistic *Graphical User Interface* will be implemented during later stages of development for easy reference and initialization. The full version of AvantZero – the Avant algorithm – will contain two separate and dedicated modes for the user to choose between either CLI or GUI-mode upon boot; allowing them to swap modes on the fly.

References

- Altar of Gaming (n.d.). *Quake III Arena*. Altar of Gaming. Retrieved July 10, 2025, from <https://altarofgaming.com/game/quake-iii-arena/>
- Merriam-Webster. (n.d.). *Avant-garde*. In *Merriam-Webster.com dictionary*. Retrieved July 9, 2025, from <https://www.merriam-webster.com/dictionary/avant-garde>
- Oxford Dictionary (n.d.). *Machinima*. In *Oxford English Dictionary*. Retrieved July 15, 2025, from https://www.oed.com/dictionary/machinima_n?tl=true
- Veenstra, J.P. (2020). DEPTHMAP NEXT [Video]. YouTube. https://www.youtube.com/watch?v=c2FT_a34vw4
- Veenstra, J.P. (2022). Quake III Arena: Experimental Depth Map use [Video]. YouTube. <https://www.youtube.com/watch?v=7cu76qMIWIQ>
- Veenstra, J.P. (2024). Dominion Algorithm. Bitbucket. Retrieved July 16, 2025, from <https://bitbucket.org/appov/dominion/src/main/>

Appendix A: Development Roadmap

The table below outlines the development roadmap for the AvantZero algorithm in a concise and summarized manner. Please refer to the *Technical Design Report* for a more detailed description.

| Version | Functionality | Summary |
|---------|---------------|---|
| 0.0.0 | Documentation | <ul style="list-style-type: none">• Functional Design• Technical Design and software diagrams• Design mock-up for AvantZero docs• Development Roadmap |
| 0.1.0 | Preparation | <ul style="list-style-type: none">• File and folder structures ready• Documentation Website Loading Screen and Homepage layout ready (accessible and responsive)• Implementation of 'Internationalization' on the documentation website – support for both Dutch and English language• Outline of algorithm functions and helper functions defined in software |

| | | |
|-------|---------------------------------|---|
| 0.2.0 | Splash Screen / Main Menu | <ul style="list-style-type: none"> • Completed Loading Screen (with helper functions to ensure all required files and packages are installed) • Completed Main Menu, all possible selections are working and lead to their respective functions • All functions can return to the main menu. • The user is able to gracefully exit out of the algorithm • If applicable, errors are properly documented and jargon-free fixes are described for the user • The homepage of the Documentation website has clickable buttons leading to (for now) empty pages, is responsive, accessible and can be accessed in two languages • The documentation website has a fully functioning loading screen |
| 0.2.1 | Splash Screen / Main Menu (GUI) | <ul style="list-style-type: none"> • A GUI is created and designed for use with the loading screen and the main menu. • All functions and helper functions make proper use of the GUI |

| | | |
|-------|---|---|
| 0.2.2 | Splash Screen / Main Menu (UnitTest) | <ul style="list-style-type: none"> • Manual and automated unit tests of the main menu, loading screen and (helper)functions • Error Database |
| 0.3.0 | Documentation: Background, Requirements, Installation | <ul style="list-style-type: none"> • Completed the Background, Requirements and Installation chapters of the Documentation website • Content of said chapters is clear, free of errors. • Pages are accessible, responsive and are accessible in Dutch and English. • User can switch languages at will |
| 0.4.0 | Project based workflow | <ul style="list-style-type: none"> • Users can see, create, edit, select and delete projects • AvantZero automatically creates the right folder structure upon creation of a project |
| 0.4.1 | Projects (GUI) | <ul style="list-style-type: none"> • GUI implementation of projects menu |
| 0.4.2 | Projects (UnitTest) | <ul style="list-style-type: none"> • Automated and manual tests of the projects menu • Update of Error Database |

| | | |
|-------|-------------------------------------|--|
| 0.5.0 | Basic Data Generation Functionality | <ul style="list-style-type: none"> • Proper implementation of data generation functions and helper functions • User should be able input video and receive a randomized .edl and .xml file |
| 0.5.1 | Data Generation (Unit Test) | <ul style="list-style-type: none"> • Automated and manual tests of Data Generation functions • Update of Error Database |
| 0.5.2 | Data Generation (GUI) | <ul style="list-style-type: none"> • GUI implementation of data generation functions with visual feedback |
| 0.6.0 | Image Sequence Support | <ul style="list-style-type: none"> • Support for a wide array of image sequences inside the generator functions |
| 0.6.1 | Image Sequence (Unit Test) | <ul style="list-style-type: none"> • Automated and manual tests ensuring compatibility with image sequences • Update of Error Database |
| 0.6.2 | Image Sequence (GUI) | <ul style="list-style-type: none"> • Full GUI support for both video- and image-based data generation |

| | | |
|-------|-------------------------|--|
| 0.7.0 | PDF Support | <ul style="list-style-type: none"> The application is simultaneously able, at a base level, to generate randomized data and output it to a PDF file |
| 0.7.1 | PDF Support (Unit Test) | <ul style="list-style-type: none"> Automated and manual tests of exporting project data to a PDF file Update of Error Database |
| 0.7.2 | PDF Support (GUI) | <ul style="list-style-type: none"> Full GUI support for PDF generation |
| 0.8.0 | CSV Support | <ul style="list-style-type: none"> The application is simultaneously able, at a base level, to generate randomized data and output it to PDF and/or CSV files through separate, dedicated functions |
| 0.8.1 | CSV Support (Unit Test) | <ul style="list-style-type: none"> Automated and manual tests of CSV related functions Update of Error database |

| | | |
|--------|--------------------------------|---|
| 0.8.2 | CSV Support (GUI) | <ul style="list-style-type: none"> • GUI implementation of CSV related functionality |
| 0.9.0 | Data Visualization | <ul style="list-style-type: none"> • Implementation of generating plots and saving these as files within the project folder |
| 0.9.1 | Data Visualization (Unit Test) | <ul style="list-style-type: none"> • Automated and manual testing of Data Visualization function • Update of Error Database |
| 0.9.2 | Data Visualization (GUI) | <ul style="list-style-type: none"> • Implementation of Data Visualization inside of the GUI |
| 0.10.0 | Depth of Field Base Support | <ul style="list-style-type: none"> • Basic support for generating Depth of Field maps |

| | | |
|--------|--|---|
| 0.10.1 | Depth of Field Base Support (Unit Test) | <ul style="list-style-type: none"> Automated and manual tests for Depth of Field generation |
| 0.10.2 | Depth of Field Base Support (GUI) | <ul style="list-style-type: none"> GUI implementation of Depth of Field Support |
| 0.11.0 | Advanced Depth of Field Support | <ul style="list-style-type: none"> Base support has been implemented. At this stage, the user should be able to generate the depth of field maps from the same, randomly generated pieces of footage, or allow the algorithm to generate depth of field maps for the entire set of footage |
| 0.11.1 | Advanced Depth of Field Support (Unit Test) | <ul style="list-style-type: none"> Automated and Manual tests Update Error Database |

| | | |
|--------|---------------------------------------|---|
| 0.11.2 | Advanced Depth of Field Support (GUI) | <ul style="list-style-type: none"> • GUI implementation of all Depth of Field related functionality |
| 0.12.0 | Error Database | <ul style="list-style-type: none"> • The Error Database has been frequently updated and contains extensive and easy-to-execute information on how to solve common problems • The Error Database is implemented on the Documentation website |
| 0.13.0 | License, Credits Pages | <ul style="list-style-type: none"> • The License page on the Documentation website has been fully completed • The Credits page on the Documentation website has been fully completed |

| | | |
|--------|-----------------|--|
| 0.14.0 | Quick Start | <ul style="list-style-type: none"> • The Quick Start page contains concise and clear information on how to easily setup a project and start generating data • The Quick Start page is accessible, responsive and readable in two different languages |
| 0.15.0 | Project Manager | <ul style="list-style-type: none"> • The project manager page contains extensive information regarding the project manager and all its functionality • The project manager page is accessible, responsive and readable in two different languages |
| 0.16.0 | Data Generation | <ul style="list-style-type: none"> • The data generation page contains extensive information regarding data generation and all its functionality • The data generation page is accessible, responsive and readable in two different languages |

| | | |
|---------|--------------------|--|
| 0.17.0 | Data Visualization | <ul style="list-style-type: none"> • The data visualization page contains extensive information regarding data visualization and all its functionality • The data visualization page is accessible, responsive and readable in two different languages |
| 0.18.0 | Depth of Field | <ul style="list-style-type: none"> • The depth of field pages contain extensive information regarding depth of field and all its functionality • The page is accessible, responsive and readable in two different languages |
| 0.19.0a | Alpha | <ul style="list-style-type: none"> • Checks and final adjustments of documentation website • Final tests and final adjustments of algorithm code |

| | | |
|--------------|------------------------|---|
| 0.20.0b | Beta | <ul style="list-style-type: none"> • Creation of TestPyPi package and installation • QA testing by a body of external individuals • Final code and documentation adjustments |
| 1.0.0 | Initial Release | <ul style="list-style-type: none"> • Completed code and documentation website • Updated GitHub repository • Uploaded all required CS50X deliverables, such as code, readme, presentation and form • Not unimportant: received CS50x certificate • Created Windows and MacOS versions of AvantZero • Released AvantZero on PyPi • Project completed |

AvantZero might receive a few *patches* and *minor updates* after its initial release to ensure full functionality and compatibility for its intended purpose. However, additional functionality will not be guaranteed as development will eventually shift towards the completion of the Avant algorithm. The repository will be publicly archived on GitHub once development on AvantZero has been considered complete.