INTERNATIONALE HOCHSCHULE

Software Engineering

Paper Plane Rotations

Ivo Prahm (IU14089444)

Computer Science (M.Sc.)

DLMCSPSE

4th February 2025

Contents

1 List of Acronyms

M2FR Moved to a Future Release

TDD Test Driven Development

CORS Cross-Origin Resource Sharing

2 Conception Phase

2.1 Abstract

Objective	Provide mainly software testers together with students with an intuitive as well as interactive way to explore the interplay of transformations and their effects on fixed points in space to gain a better understanding of their individual problem at hand.
Main tagret group	Software testers
Secondary tagret group	Students
Target release	**2 Feb 2025**
Key Perf. Measures	Customer feedback, Employee satisfaction, Session duration
Status	**Done**
Designer	Ivo Prahm
Web Application	ivolino.de/plane
Git Repo.	GitHub - Den-Prahms-ihr-Ivo/DLMCSPSE

2.2 Problem Statement

Visual-spatial intelligence is challenging for the human brain and often times sheer impossible the more complex the layot becomes. Writers of test cases for software involved with more complex transformations and their interplay between objects frequently run into this issue. As well as students trying to gain an understanding of the math behind these transformations and what their calculated transformation matrix actually does.

Since all these transformations are immensely abstract, a visual representation of the problem at hand would help to write better test cases, gain a better understanding of specific szenarios and therefore to facilitate the identification of possible edge cases. Hence, improving the test-suite even further. Which in turn will improve the reliability, robustness and maintanance of the delivered product.

It is for that reason that I am proposing a tailor made tool to assist the design process of aforementioned test cases. Testers of software that has to do with aviation or obstacle detection in a broader sense are now able

to easier visualise scenarios, get a first idea of expected values and may see potentially neglected test cases by rotating and moving the little plane in space and observing the impact on other points in space.

2.3 Assumptions

Considering the target audience, a basic understanding of trigonometry is to be expected. Also it is presumed that this application is likely not going to be used on mobile devices, as its intention lies more in a univerity milieu.

Since the content of this web-app is aimed towards a more tech-savvy audience, backwards compatibility with older browsers is not considred as of now.

2.4 Risks and Measures for Risk Minimization

The goal could be too lofty and not possible to be built in the designated time frame by only one part-time developer. → Cut down on **NICE TO HAVE** -Requirements and Requirements that are not essential for the functionality of the project.

Implementing a full frontend test-suite is out of scope for this project and may lead to problems due to insufficient testing. → Design a comprehensive checklist that is to be passed before every release.

2.5 Requirements

A very rough sketch of future requirements. Reminiscent of the first meeting with stakeholders in which a broad idea is sketched out and refined in later sprints.

2.5.1 Requirements - Non-Functional

Requirement	Scope
An instance of the project shall be directly accessible via the internet.	**MUST HAVE**
The Application shall run in modern browsers (focussing on FireFox, Chrome and Opera).	**MUST HAVE**
The application shall run as a dockerized container.	**NICE TO HAVE**

2.5.2 Requirements - Functional

User Story	Scope
As a user I want to be able to rotate the plane by providing euler angles.	MUST HAVE
As a user I want to be able to see the current angle and distance between a selected threat and the plane.	MUST HAVE
As a user I want to be abler to freely translate the plane in space by providing a translation vector in cartesian coordinates.	MUST HAVE
As a user I want to freely add and remove individual threats to the scene.	MUST HAVE
As a user I want to see a 3D representation of the plane and threats.	MUST HAVE
As a user I would like to see a 2D projection next to the 3D scene. Akin to the stereotypical image of a submarine radar.	MUST HAVE
As a user I want to move threats in space by providing a translation vector.	NICE TO HAVE
As a user I would like to enter a full transformation matrix by hand and see its effect on the plane and/or bird.	NICE TO HAVE
As a user I would like to move the calculation center arbitrarily.	NICE TO HAVE
As a user I would like to visually differentiate my added threats.	NICE TO HAVE
As a user I would like to see an animation of the entered movement.	NOT IN SCOPE
As a user I would like to use this tool on all my devices.	NOT IN SCOPE

2.6 Software Development Methodology

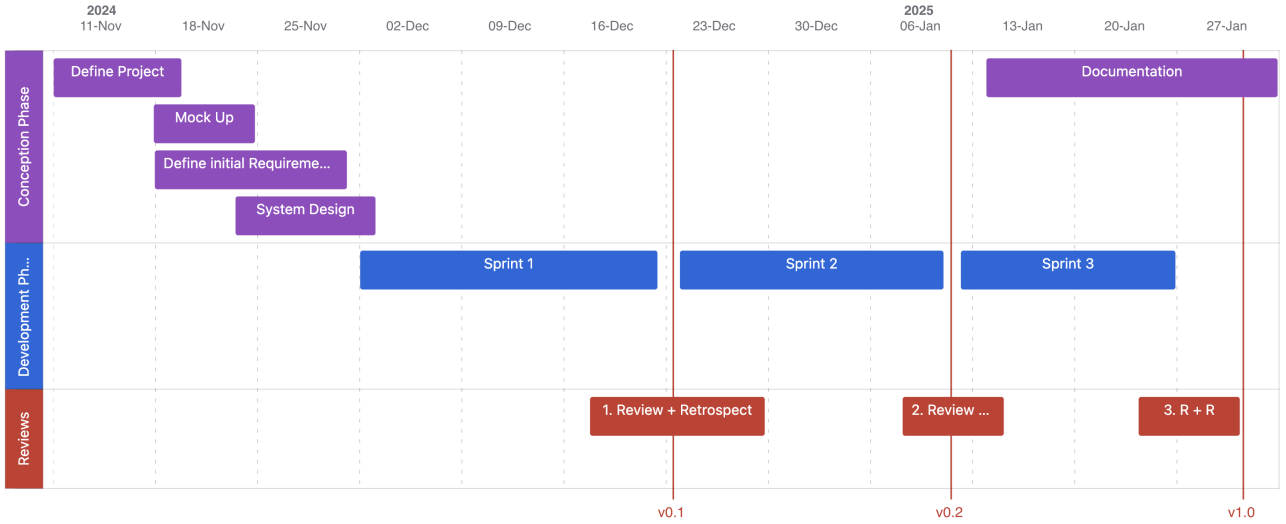
Since Scrum excels at projects with high levels of uncertainty that require an adaptive approach, and the idea for this app was only a rough sketch at the beginning of the project, Scrum is the perfect pick.

Because this is not my fulltime occupation, I decided on three sprint intervals of roughly three weeks each to have enough implementation time for each sprint. After each of the following review phases the requirements get updated accordning to newly discovered requirements and ideas from the customer/me.

As this project is developed only by me, the obligatory Scrum meetings are replaced with a relaxing run outside to contemplate ideas or do my sprint retrospective.

2.7 Timeline

The Timeline of the project plan is built according to the standard scrum sprint procedure.



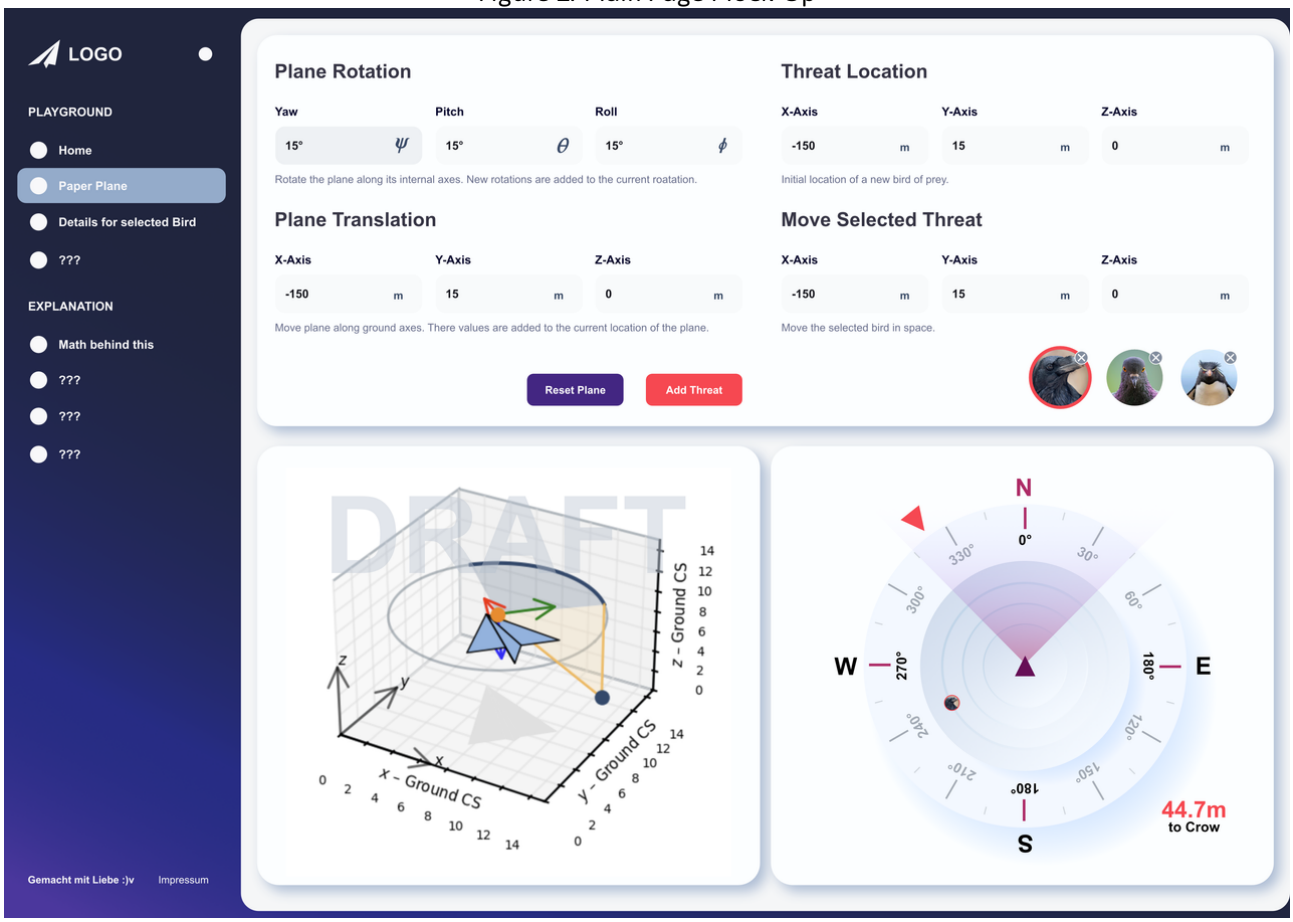
2.8 Milestones and Deadlines

Milestones and Deadlines	Version	Deadline	Status
Basic Layout + Diagram-Skeleton	v0.1	**29 Dec 2024**	**DELIVERED**
Functionality	v0.2	**19 Jan. 2025**	**DELIVERED**
Finalisation/Polish + (inevitable) Bug Fixes	v1.0	**2 Feb. 2025**	**DELIVERED**

2.9 User Interaction and Design

Mock-up of the main intention and vital component of this project. All other minor aspects were designed on the fly according to the respective user stories.

Figure 1: Main Page Mock-Up



2.10 System Design

Since the app is a self-contained single-page application and does not require a backend, isn't part of any larger system that would justify a container diagram (yet), and has no need for any from of authorization at this point, the system design is omitted.

As for 3rd-pary tools:

- The App shall be developed in TypeScript using the vite framework (V. Inc., 2019a).
- For 3D Graphics the open source library plotlyjs shall be used (Plotly, 2025).
- Testing is done using the vitest framework (V. Inc., 2019b).
- The App shall be deployed as a dockerized container on my personal server (D. Inc., 2025).

3 Implementation Phase

3.1 Objective

The purpose of this application is to provide an intuitive as well as interactive tool to visualise transformations and their respective effect on the relation to fixed points in three-dimensional space.

This tool is supposed to support the development of better test cases as well as aid students in the understanding of trigonometry. This tool is mainly aimed at software testers, as it helps them to get a more tangible vision of possible szenarios and now imaginable edge-cases. A secondary target group are (engineering-)students struggeling to get their head around trigonometry.

Since rotations in space become far more difficult to envision if we are concerned about an arbitrary point on the plane, and its relation to fixed points in space, instead of its center, this issue is being addressed by this application.

Although there are many other rotation visualisation tools out there, none of them visualize the relation to fixed points in space, let you move said points or the plane in space and don't address afroementioned issues.

3.2 Requirements Refinement

Previous rough user stories now get a clear testable statement assigned to them.

Requirements that are tagged with **M2FR**, could not be implemented in time for the planned release and had to be moved to a future release.
Since these requirements are not essential for the core functionality and a timely release is ranked higher than the fulfillment of optional requirements, this was considered justified.

3.2.1 Functional

Requirement	User Story	Importance	Jira Issue	Status
1. A maximum of 5 bird Avatar-elements shall be displayed in the main page. The most recent bird shall be at the front of the list. **2.** An Indicator shall signal if more than 5 birds were added.	As a user I want to see an overview of my added birds in the main page	**HIGH**	DLMCSPSE-224	**DONE**

Continued on next page

Table 5 – continued from previous page

Requirement	User Story	Importance	Jira Issue	Status
1. Birds shall be addable by providing the initial XYZ-coordinates. If no coordinate or a NaN-Value is provided, it shall be interpreted as 0. **2.** A bird shall not be addable if its distance is greater than 100m from the plane. If such values are provided, the user shall be informed via a Toast-Message. **3.** A bird shall be removable via a simple click on an X-icon, coloured in the birds highlight colour, on/adjacent to the bird's avatar.	As a user I want to easily add and remove nearby threats without leaving the main page	HIGH	DLMCSPSE-225	DONE
1. A click on the birds avatar shall select the corresponding bird. **2.** The fact that a bird is selected shall be visible in the UI. **3.** Only one bird shall be selectable at a time. **4.** A second click on a selected bird shall deselect said bird.	As a user I want to be able to highlight a specific bird to make this specific bird stand out in the UI.	HIGH	DLMCSPSE-226	DONE

Continued on next page

Table 5 – continued from previous page

Requirement	User Story	Importance	Jira Issue	Status
1. The user shall be able to translate the plane in space by adding values to the current position of the planes XYZ-axis. **2.** If no coordinate or a NaN-Value is provided, it shall be interpreted as 0. **3.** The plane shall not be able to be translated below ground. **a.** If this would be the case, the plane shall not be moved and an error shall be displayed.	As a user I want to move the plane in space.	HIGH	DLMCSPSE-228	DONE
1. The plane shall not be distorted → the viewport shall remain a cube. **2.** The view shall always be centred around the plane. **3.** The side length of aforementioned cube shall be the distance to the furthest threat and at least 10.	As a user I always want to see the plane centred and as close as possible.	HIGH	DLMCSPSE-229	DONE
All birds shall be displayed as a point in space in the colour of the corresponding bird.	As a user I want to see my added birds in the 3D-Diagram in the main page	HIGH	DLMCSPSE-231	DONE

Continued on next page

Table 5 – continued from previous page

Requirement	User Story	Importance	Jira Issue	Status
1. The user shall be able to translate a bird in space. **2.** A Bird cannot be below ground.	As a user I want to be able to move a threat in space	HIGH	DLMCSPSE-232	DONE
The user shall be able to rotate the plane by providing Euler Angles in ZYX order. It shall be possible to enter angles greater than 360° If no angle or a NaN-Value is provided, it shall be interpreted as 0.	As a user I want to be able to rotate the plane.	HIGH	DLMCSPSE-233	DONE

Continued on next page

Table 5 – continued from previous page

Requirement	User Story	Importance	Jira Issue	Status
1. The user shall be able to navigate a page displaying every bird as a card component detailing its relation to the plane. Listed shall be the following information **a)** name **b)** distance to plane **c)** Horizontal Distance to plane **d)** angle from plane 2 bird (azimuth and elevation) **e)** altitude **f)** an indication of the direction and horizontal distance as an icon from the viewpoint of the user. **2.** The information on the main page shall not be lost when navigation to and back from this page.	As a user I want to see a detailed view of all the birds and their relation to the plane.	HIGH	DLMCSPSE-240	DONE
The planes X-Axis shall represent the head of the plane and the ground X-Axis shall be treated as North (positive) and South (negative) in the compass component.	As a user I want to see a 2D "Compass-like" Representation of the planes rotation.	HIGH	DLMCSPSE-244	DONE

Continued on next page

Table 5 – continued from previous page

Requirement	User Story	Importance	Jira Issue	Status
The compass component shall indicate the birds' current, relative to the plane, position. **Horizontal Distance <= 14:** A Bird Icon shall indicate the position. **Horizontal Distance >14:** A Triangle shall indicate the direction of the bird	As a user I want to see the birds displayed in their relative distance to the plane in the compass component.	HIGH	DLMCSPSE-245	DONE
The home page shall contain an article explaining the idea and concept behind this app	As a user I want to read an explanation on what I am seeing here	HIGH	DLMCSPSE-249	DONE
The birds name shall be displayed in a Tooltip with the birds color as the background color over the birds avatar.	As a user I want to see the birds name when I hover over it, to later identify it easier in the Card View	LOW	DLMCSPSE-227	DONE
The footer shall include a link to a page displaying the legal information. All transformations that have been made, up to the point of navigating away, shall not be lost when navigating back to the main page.	An Impressum is mandated by law to be included on every published medium in German speaking countries.	LOW	DLMCSPSE-236	DONE

Continued on next page

Table 5 – continued from previous page

Requirement	User Story	Importance	Jira Issue	Status
-	As a user I would like to see the selected birds name in the "Move Bird"-Component. So it is absolutely clear which bird I'm translating.	LOW	DLMCSPSE-242	DONE
-	As a user I would like to set the center of my angle and distance calculation to the center of the plane	LOW	DLMCSPSE-247	DONE
Show two Arrows at ground Level indicating N and E	As a user I would like to easier see North and East in the Diagram	LOW	DLMCSPSE-248	DONE
1. The name shall be displayed in the birds colour. **1.** The layout shall be '*<Name>*\n$(X, Y, Z)$'	As a user I would like to see the name of the bird as I'm hovering over it in the diagram.	MEDIUM	DLMCSPSE-230	DONE
A page displaying the math behind these transformations shall be accessibe via the main nav bar.	As a user I want to understand the math behind these transformations	MEDIUM	DLMCSPSE-234	DONE
The initial values of the plane shall be restoreable with the click of a button.	As a user I want to reset the plane to its default position without having to reload the page.	MEDIUM	DLMCSPSE-235	DONE

Continued on next page

Table 5 – continued from previous page

Requirement	User Story	Importance	Jira Issue	Status
The rendering of the coordinate system shall be conditional. But the marble shall stay and not be rendered conditionally. The user shall toggle this condition via an switch component in the main page.	As a user I want to be able to hide and show the "Pilot" and plane coordinate system at will.	MEDIUM	DLMCSPSE-239	DONE
Display a shadow of the plane on the ground. The sun shall be assumed to be perpendicular to the ground.	As a user I would like to a shadow of the plane to get a better sense of perspective.	MEDIUM	DLMCSPSE-241	DONE
These values shall only be rendered, iff a bird is selected.	As a user it would be nice to see the angle and distance to the selected bird in the compass component.	MEDIUM	DLMCSPSE-246	DONE
The NavBar shall contain a link to a page showing a bigger view of the plane scene.	As a user I would like to be able to see a bigger image of the plane scene.	MEDIUM	DLMCSPSE-250	DONE
-	As a user I would like to see visual pleasing 404 Page if i entered the wrong url.	LOW	DLMCSPSE-243	M2FR
The math explanation page shall contain at least one diagram explaining the azimuth and elevation angles along with imporant sides aiding in calculation	As a user I want to see an explanation of the mentioned azimuth and elevation angles	MEDIUM	DLMCSPSE-237	M2FR

Continued on next page

Table 5 – continued from previous page

Requirement	User Story	Importance	Jira Issue	Status
The math explanation page shall contain at least one diagram visualising the planes axes.	As a user I want to see an explanation of the planes coordinate system	MEDIUM	DLMCSPSE-238	M2FR

3.2.2 Non-Functional

Requirement	Importance	Status
The Application shall run smoothly in the following browsers: Chrome (Version $\geq$ 109) FireFox (Version $\geq$ 98) Opera (Version $\geq$ 100)	HIGH	DONE
The application shall run as a dockersied container.	HIGH	DONE
An instance of the project shall be directly accessible via the internet.	HIGH	DONE
In Terms of usability, the user shall navigate the page without losing entered information.	HIGH	DONE
The The loading time shall not exceed 3 seconds to not lose customers (Ltd., 2025).	HIGH	DONE
Images shall either have the format .svg or .webp.	MEDIUM	DONE
New birds/threats shall be addbale by a single click.	MEDIUM	DONE
The The loading time shall not exceed 1 second to increase customer satisfaction (Ltd., 2025).	MEDIUM	DONE

Continued on next page

Table 6 – continued from previous page

Requirement	Importance	Status
For further usability it shall not be necessary to have to fill out every input field to be able to add new birds or translations.	MEDIUM	DONE
A faulty input shall not delete all progress that has been made so far, to not frustrate customers.	MEDIUM	DONE

3.3 Out of Scope

- **Possibility for the user to enter a full transformation matrix by hand.**
- **Adding a Trajectories** - via speed vectors in x,y and z direction. And calculating possible intersections/collisions.
- **Mobile responsiveness** - mobile users aren't the target audience. Therefore the extra time can not be justified.
- **A complete automated Frontend Test Suite** - currently I confined myself to manual tests after every major update to ensure correct functionality. This is feasible since the scope/functionality of this project is comparably small and the extra time can not be justified.
- **Backwards compatibility with older browser**
- **Assure correct rendering in more niche browsers**
- **Undo/Redo functionality**
- **Add a 2D ('Ship'-)Mode** - Still render everything in 3D but give the option to be limited to the ground and render the plane as a ship. In this case threats could come from above and below.
- **Move and rotate Marble arbitrarily** - The point on the plane from which all measurements are calculated shall be freely positionable in relation to the plane.

3.4 Requirements that were moved to the next release.

These requirements were not fulfillable in the set time frame and had to be moved to the next release since they are not vital for the intended functionality of the app.

- `DLMCSPSE-238` (Math Explanation Page)
- `DLMCSPSE-237` (Math Explanation Page)
- `DLMCSPSE-243` (404 Page)

3.5 Documentation of Software & System Architecture

The basic architecture of this web-application follows the schematic sketched in Figure 2. The respective reducer, which is basically just a function, defines actions that atomic components can dispatch, for example ROTATE-PLANE or TRANSLATEPLANE. This way, the reducer is the single place where updates for the, for instance, plane rotation logic have to be performed or new logic is to be added. The reducer returns the updated instance (either a transformed plane or an updated array of birds) which is then used to update the overarching context and further propagated to the corresponding components by react.

Atomic components are the parts of the web application that the user interacts with. They are either used to display information to the user, like in the BirdCardComponent, or to enable the user to input information into the system, like the MathInputComponent.

Pages (rectangles) and Components (circles) that are involved in updating and consuming the state context are presented in Figure 3.

Figure 2: Basic structure of the web application

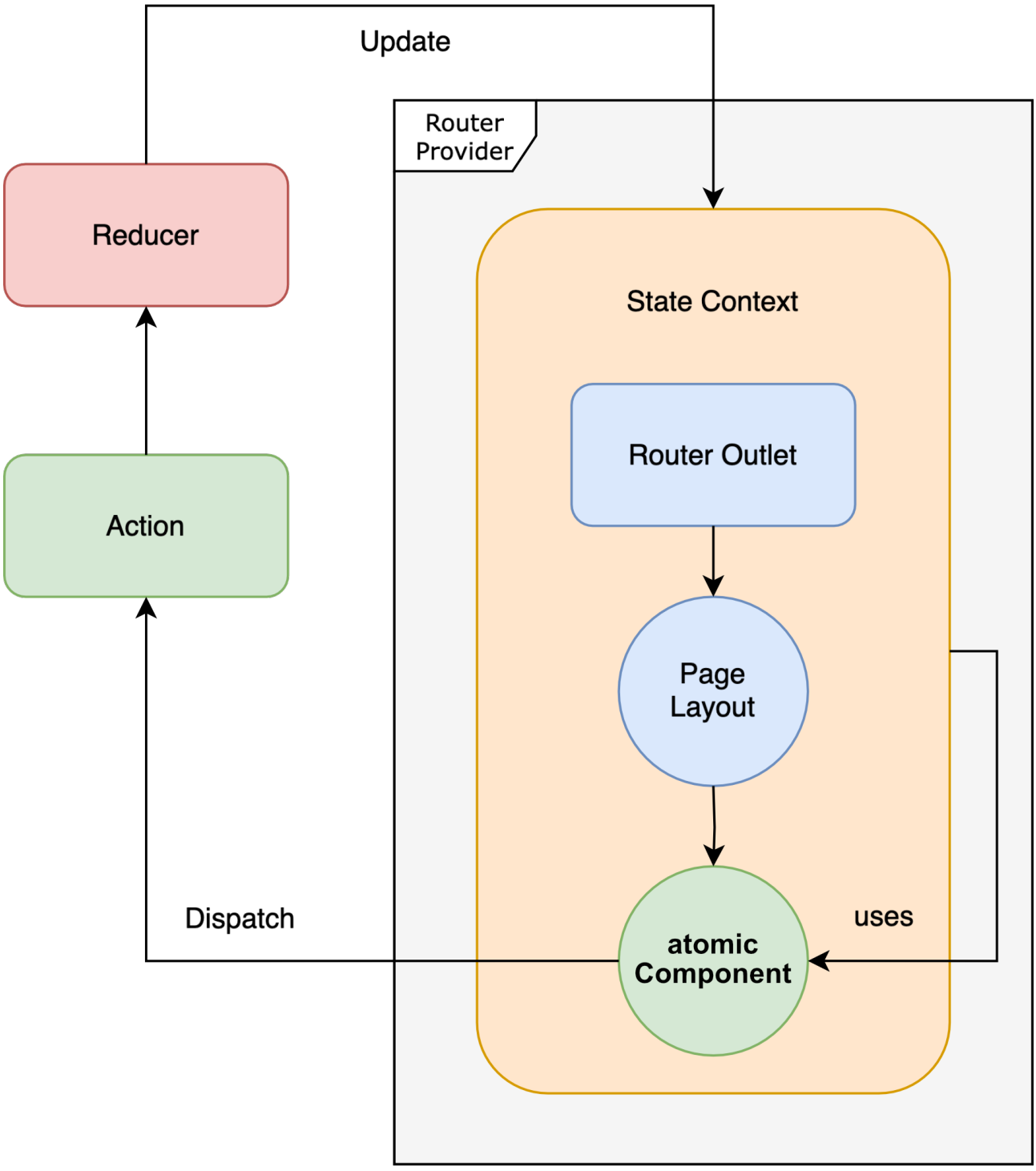
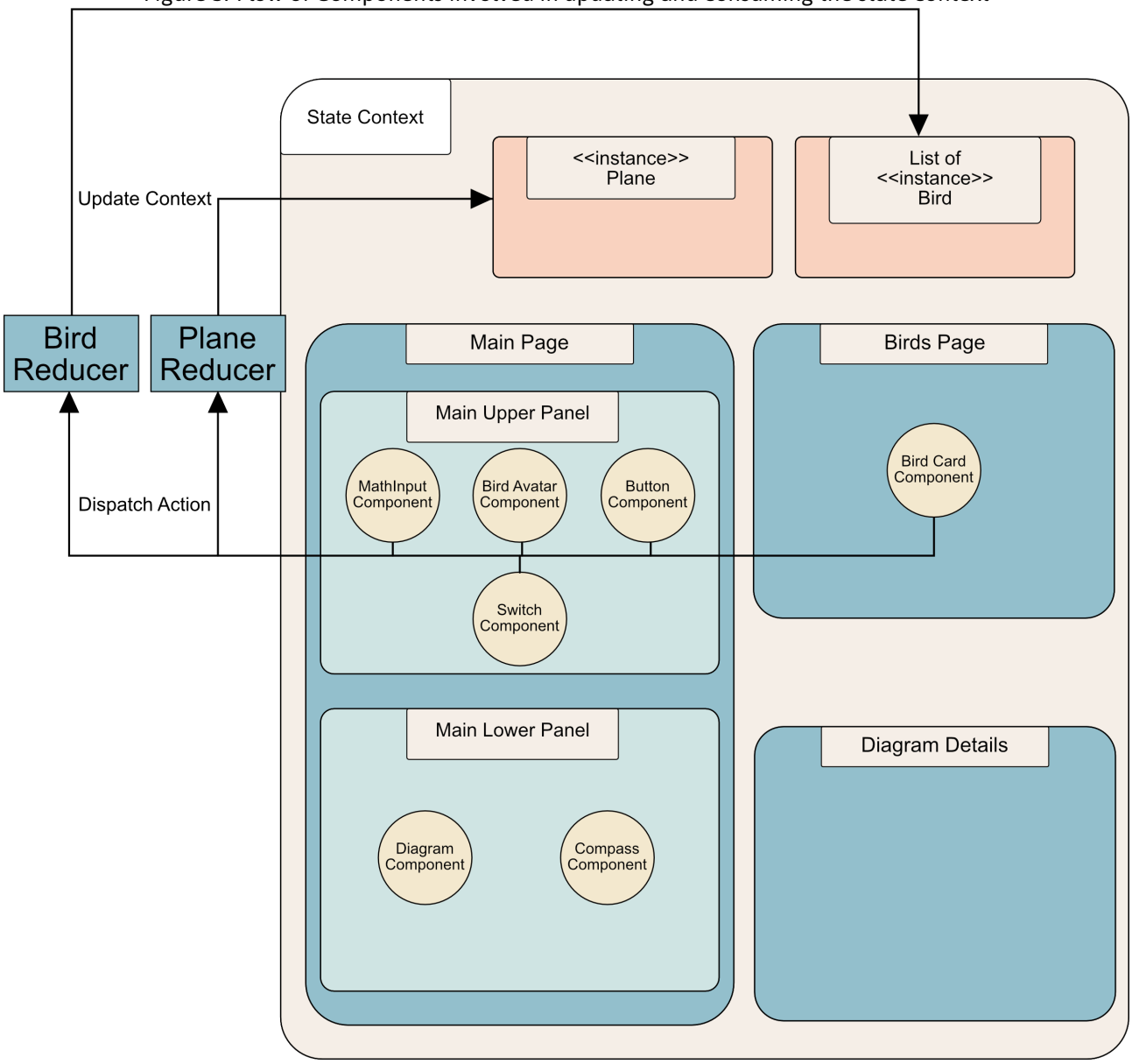


Figure 3: Flow of Components involved in updating and consuming the state context



3.6 Design and Implementation Procedure

The design process was mostly defined by drawing and redefining new mock-ups that were refined after each sprint. The implementation of mathematical functions followed a Test Driven Development (TDD) procedure.

3.7 Used 3rd Party Libraries

The following 3rd party libraries were finally used in the making of this application.

- React (Meta Platforms, 2024), because of its simple design and vast ecosystem and the ”ease” of state updates.
- React Router (Shopify, 2024) as a well established way to navigate a website.
- React Icons (kamijin fanta, 2018) as a nice and easy way to display common FontAwesome (Fonticons, 2024) Icons.
- Chakra-UI (Systems, 2025), again for its ease of use and customizability of existing designs.
- PlotlyJS (Plotly, 2025), because it is the only graphing library I found that I could misuse for drawing shapes rather than the intended purpose of presenting data graphically. On top of it all it is well documented.
- Vite (V. Inc., 2019a) as a modern building tool, to get the app ready for production.
- Vitest (V. Inc., 2019b), since I’m already using vite, the corresponding testing framework was the only logical choice.
- Docker (D. Inc., 2025), which was not really necessary for my application in the end. But since the task asked for it, the webserver still runs as a dockerized container, running an NGINX F5, 2025 on the backend, serving the distribution version of this web application. Thanks to problems with CORS, docker can also be used to run the application locally on your machine. See listing 3.7 for more details.

```
docker build -t paper-plane .
docker run -p 3137:80 paper-plane

# And then access the page via:
http://localhost
```

4 Test Document

4.1 Test Plan

Test Coverage for all functions that are involved in mathematical calculation shall have 100% code coverage and shall cover edge cases. Functions and methods involved in mathematical calculations are listed under 'Mathematical Functions'.

The user interface shall be tested according to the checklist listed under 'Manual Tests'.

4.2 Manual Tests

A page reload shall be performed before each test case.

Test Case	Expected Outcome	Satisfies	Status
On the 'Paper Plane'-Page, click on 'Add Threat' 6 times.	Only 5 bird avatars shall be displayed and a grey circle showing `'+1'`. Pay attention that the first bird is the one that disappears after the 5th bird. All birds shall be displayed at `(0,0,0)`.	DLMCSPSE-224 DLMCSPSE-225-1	PASSED
Click 'Add bird' without providing any number. Click on the 'X'-Icon next to the birds Avatar. Add a bird at `(101,0,0)`.	A bird shall be added at `(0,0,0)`. Now no bird shall be displayed. No bird shall be added. A toast shall be displayed.	DLMCSPSE-225	PASSED
Enter the coordinates `(1,2,3)` and click 'Add Threat'.	A new bird shall be displayed at the specified coordinates.	DLMCSPSE-225-1	PASSED
Enter the coordinates `(101,2,3)` and click 'Add Threat'.	No new bird is added and a warning pop-up is displayed.	DLMCSPSE-225-2	PASSED

Continued on next page

Table 7 – continued from previous page

Test Case	Expected Outcome	Satisfies	Status
Click 'Add Threat'. Enter the coordinates `(1,2,3)` and click 'Add Threat' again. Click on the little x of the first bird.	Only one bird avatar is displayed at `(1,2,3)` on each page and in the 3D-Scene.	DLMCSPSE-225-3	PASSED
Add two birds. Click on a bird. Click on another bird. Click on the same bird again.	First no bird shall be selected. This bird shall now have a ring around it. And the compass component shall display 0m to <BirdName> and <BirdName> at $0°$. Now the other bird shall have a ring around it and the ring of the first bird shall disappear. Now no bird shall have a ring. The extra values in the compass component shall now dissapear.	DLMCSPSE-226 DLMCSPSE-246	PASSED
Move the plane -3 on the x-axis.	The plane shall not move and a toast shall be displayed.	DLMCSPSE-228-3	PASSED
Add a bird at `(50,2,2)` Move the plane -10 on the x-Axis.	Before doing anything. The scale of the scene shall be `(-6,6)` for x/y and `(0,12)` for the z-Axis. After the addition of the bird. The scale shall be: `(-50,50)` for x/y and `(0,100)` for the z-Axis. After the translation the scale shall be `(-60,60)` for x/y and `(0,120)` for the z-Axis.	DLMCSPSE-229 DLMCSPSE-228-1/2	PASSED

Continued on next page

Table 7 – continued from previous page

Test Case	Expected Outcome	Satisfies	Status
Add a bird at `(2,2,-3)`. Add a bird at `(2,2,2)`. And then try to move the bird z -3. And then move the bird x 2.	No bird shall be added and a toast ist expected. A new dot in the colour of the bird is expected. The bird shall not move and a toast shall pop up. After the second translation the bird shall be at `(4,2,2)`.	DLMCSPSE-232 DLMCSPSE-231	PASSED
Rotate the plane yaw $405°$.	The plane shall point NE 45°.	DLMCSPSE-233-2	PASSED
Randomly rotate the plane and click on 'reset plane'	the plane shall now be at `0,0,2`.	DLMCSPSE-235	PASSED
Click 'Hide Coordinate System' twice.	After the first click the coordinate system in the scene shall disappear and reappear after the second click.	DLMCSPSE-239	PASSED
Add a bird at `(2,2,2)`. And navigate to the 'Bird' Page. Navigate back.	A bird card shall be visible displaying the following information: 2.83 Meter 2.83 Meter -0.73 Degree 3.13 Meter After navigating back the information shall not be lost.	DLMCSPSE-240	PASSED

Continued on next page

Table 7 – continued from previous page

Test Case	Expected Outcome	Satisfies	Status
Add a bird at `(3.3,0,0)`. move to the 'Birds' component and check. move back and translate the bird `(0,0,3.3)` and move back to the Bird Cards.	The point `(0,0,3.3)` is the planes marble. So an elevation angle of $-45°$ is to be expected. After translation an angle of $0°$ is expected.	DLMCSPSE-240-f	PASSED
Rotate the plane yaw $90°$. Add another rotation yaw $225°$.	Before doing anything, the plane in the compass component shall point north. After the rotation the plane shall point East. After the second rotation the plane shall point NW ($-45°$)	DLMCSPSE-244	PASSED
Add a Threat at (2,2,2).	A Bird Icon shall be displayed at roughly $315°/-45°$ in the compass component.	DLMCSPSE-245-1	PASSED
Add a Threat at `(14,0,0)` and a Threat at `(13,0,0)`.	The first one shall be indicated by an arrow and the second one shall be indicated by an icon at $0°$	DLMCSPSE-245-2	PASSED

4.3 Mathematical Functions

List of functions and methods that are involved in mathematical calculations and are therefore essential to test properly. Coverage is calculated by the `vitest` testing framework.

ID	Function/Method	Description	Coverage
F1	`azimuthAngle(A: Point, center?: Point): number`	Calculates the angle measure (in radians, with $-\pi < \theta \leq \pi$ between the positive x-axis and the ray from the origin to the point `(x,y)` in the Cartesian plane. If a center point is provided, the angle is calculated from the new center position.	**100%**
F02	`yawPitchRoll2Matrix( yaw: number, pitch: number, roll: number ): number[][]`	Returns a $3x3$ rotations matrix that represents a rotation in ZYX order in degree. Yaw, Pitch and Roll are in degrees.	**100%**
F03	`cosineDegrees(degrees: number): number`	Calculates the cosine of a number in degrees.	**100%**
F04	`sineDegrees(degrees: number): number`	Calculates the sine of a number in degrees.	**100%**
F05	`rad2Degrees(rad: number): number`	Converts rad to degrees.	**100%**
F06	`crossProduct(pointA: Point, pointB: Point): Point`	Calculates the cross product between two vectors	**100%**
F07	`vectorLength(p: Point): number`	Calculates the length of a vector	**100%**

Continued on next page

Table 8 – continued from previous page

ID	Function/Method	Description	Coverage
F08	`dotProduct(pointA: Point, pointB: Point): number`	Calculates the dot product of to vectors	**100%**
F09	`angleBetween2Points(A: Point, B: Point): number`	The final angle between is calculated according to the dihedral angle: The dot product of the two normal vectors divided by the product of their lengths. $cos(phi) = Norm(n_A * n_B)/(Norm(n_A) * Norm(n_B))$	**100%**
F10	`subtract2Point(pointA: Point, pointB: Point): Point`	Subtracts two vectors.	**100%**
F11	`multiply(a: number[][], b: number[][]): number[][]`	Multiplies two matrices.	**100%**
F12	`buildTransformationMatrix( rotationMatrix: number[][], translationVector: number[] ): number[][]`	Builds a transformation matrix from the rotation and translation matrix.	**100%**
F13	`matrixTransformation( transformationMatrix: number[][], R: number[][] ): number[][]`	Multiplies a point with a transformationsmatrix and thereby transforms it in space.	**100%**
M01	`matrixTransform: void`	Applies all subsequent transformations to the initial plane, marble and plane coordinate system.	**100%**
M02	`resetPlane(): Plane`	resets plane to its initial position.	**100%**

Continued on next page

Table 8 – continued from previous page

ID	Function/Method	Description	Coverage
M03	`rotatePlane(yaw: number, pitch: number, roll: number) : Plane`	Updates the rotation matrix via Euler Angles in ZYX order, rotates the plane and returns the freshly transformed plane.	**100%**
M04	`translatePlane(x: number, y: number, z: number): Plane`	Updates the planes translation matrix, translates the plane and returns the freshly transformed plane.	**100%**
M05	`getAngle2North(): number`	Returns the angle α between the planes x-Axis and the earth referential x-Axis in degrees. $\alpha \in [0..359]$	**100%**
M06	`getVerticalDistance2Plane(p: Point): number`	Returns the vertical hight difference between the plane and a point in space.	**100%**
M07	`getDistance2Plane(p: Point): number`	Returns the length of the vector $\vec{MP}$ from the Marble M to an arbitrary Point P.	**100%**
M08	`getElevationAngle2Threat(p: Point): number`	Returns the vertical angle α between the Marble and an arbitrary Point. $\alpha \in [-179..180]$	**100%**
M09	`getAngle2Plane(location: Point): number`	Returns the horizontal/azimuth angle α between a point in space and the marble $\alpha \in [0..359]$	**100%**
M10	`getDistanceFromGround(): number`	Returns the altitude of the plane.	**100%**

5 Finalisation Phase

5.1 Sprint Retrospective - Main Take-Aways

Especially when working alone, complacency of procedure becomes a major issue and needs to get extra attention during future projects.

It is vital for future project plans and/or deadlines to be more realistic in terms of what is achievable in the given time/cost-frame.

5.2 Methodologies and Tool

You should especially reflect on the applied tools, methodologies, and techniques and on your handling of the resources

In future it maybe more sensible to switch to a backend centred application that utilises python and its vast ecosystem of well-tested mathematical function and not having to reimplement basic functionalities like matrix multiplication and therefore introducing unnecessary risk for error. Especially if the application is expected to grow and the calculations will get more complicated.

6 Glossary

Term	Definition
Bird	A more friendly and on theme alias for *Threat*.
Marble	A coordinate system that is offset from the center of the plane and is rotated along with the plane. It represents a point on the plane that is affected by its transformation. This system is the heart of all calculations made in this Tool.
Pilot	A more friendly and on theme alias for *Marble*. Since the *Marble* is currently not moveable, it serves as a stand-in for the head of a pilot.
Plane	A visual pleasing and more tangible object to visualise the effects of transformations.
Threat	A point in space to which the Azimuth & Elevation as well as the distance from the *Marble* is calculated.

References

F5, I. (2025). *NGINX*. Retrieved February 4, 2025, from https://www.f5.com/go/product/welcome-to-nginx

Fonticons, I. (2024). *Font Awesome v6*. Retrieved February 4, 2025, from https://fontawesome.com/v6/icons

Inc., D. (2025). *Docker*. Retrieved January 23, 2025, from https://www.docker.com/

Inc., V. (2019a). *Vite*. Retrieved January 23, 2025, from https://vite.dev/

Inc., V. (2019b). *Vitest*. Retrieved January 23, 2025, from https://vitest.dev/

kamijin fanta. (2018). *React Icons*. Retrieved January 30, 2025, from https://react-icons.github.io/react-icons/

Ltd., B. P. (2025). *What is page load time and why is it important?* Retrieved February 4, 2025, from https://www.bigcommerce.com/glossary/page-load/

Meta Platforms, I. (2024). *React*. Retrieved January 30, 2025, from https://react.dev/

Plotly. (2025). *PlotlyJS*. Retrieved January 23, 2025, from https://plotly.com/javascript/

Shopify, I. (2024). *React Router DOM*. Retrieved January 30, 2025, from https://reactrouter.com/

Systems, C. (2025). *Chakra UI*. Retrieved January 30, 2025, from https://www.chakra-ui.com/