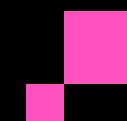


TOTAL WAR:
ROME II
CALCULATOR

Calculator mod

Team D:
Alexander Redinger
Thol Ucca Kool
Oziel Martinez
Jong SovanDara



Introduction & Project Motivation

Introduction

The Calculator Mod is a web-based application designed to simplify and enhance the process of creating custom factions for Total War: Rome II. The project integrates a user-friendly online faction builder, powered by JavaScript and hosted through W3Schools Spaces, allowing players to generate and balance their own factions without needing advanced modding experience.

The system automates the calculation of unit costs, upkeep, and other balance factors, exporting completed factions directly into a shared master mod via backend scripts developed in JavaScript and integrated with the Rome II Assembly Kit. The result is an accessible tool that bridges the gap between community creativity and game development, offering an organized, data-driven approach to mod creation.



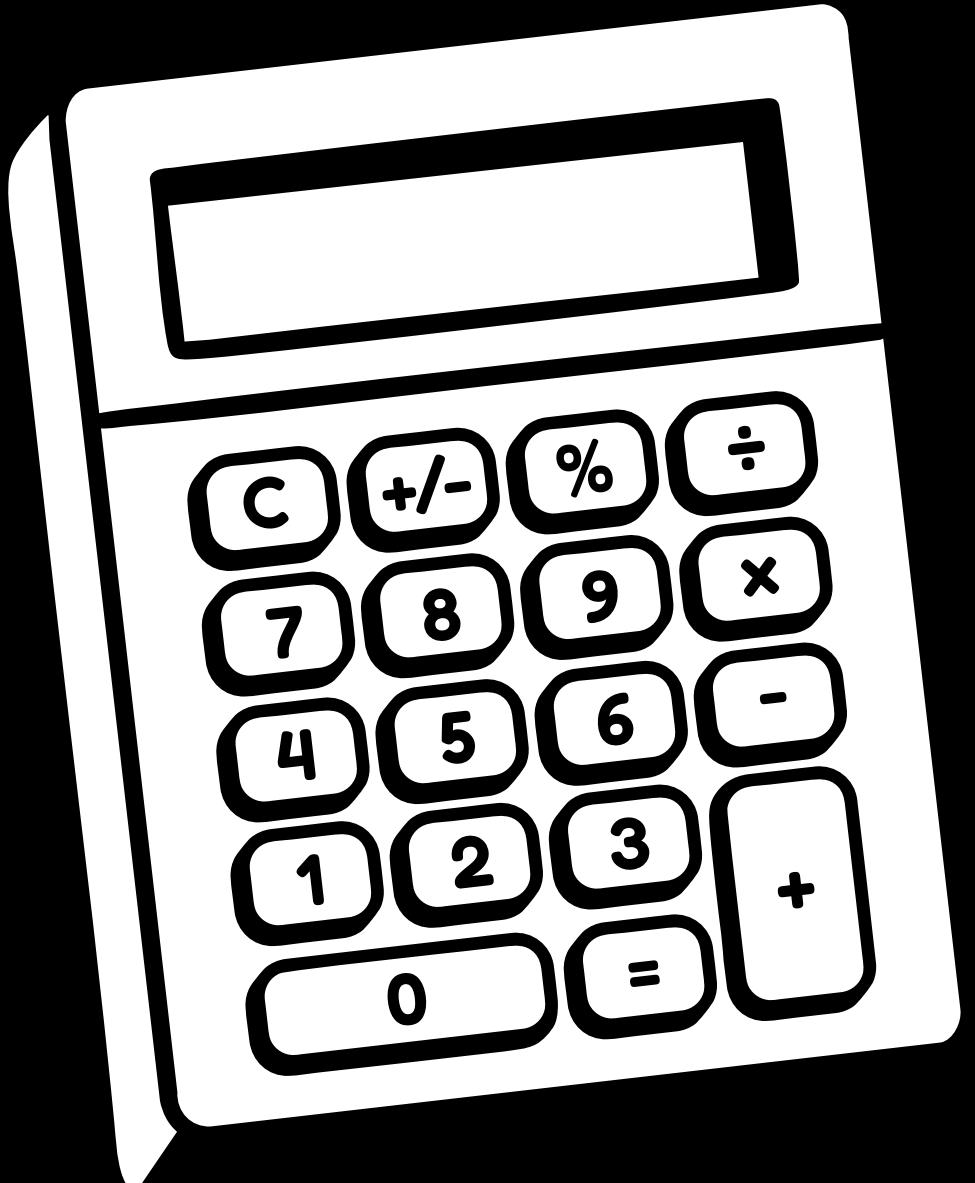
Project Motivation ➤

Modding in Total War: Rome II is often complex, requiring multiple tools, spreadsheets, and deep technical knowledge of the game's internal data structures. For most players, this complexity creates a barrier between creative ideas and practical implementation. The Calculator Mod was conceived to remove those barriers by providing a streamlined, automated, and collaborative environment where anyone can design balanced factions through a simple web interface.

This project is motivated by three key goals:

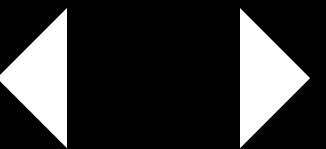
1. Accessibility: Empower players with little to no modding experience to create and share custom factions easily.
2. Balance and Fairness: Use automated formulas and validation checks to ensure all community-made factions remain competitive and balanced.
3. Collaboration: Centralize all user-created factions into a shared master mod that updates dynamically, encouraging long-term community participation.

The team's vision is to turn the modding process into an interactive and educational experience, combining game design principles with software engineering practice. By linking web technologies with game data management, The Calculator Mod not only supports creativity but also demonstrates how modern web systems can integrate with legacy gaming frameworks.



Customer Problem Statement & Requirements: Negan (the player)

- Negan, a creative Total War: Rome II player, wants to design a unique Lord of the Rings-style faction.
- He lacks the time and technical knowledge to manage complex spreadsheets and modding tools.
- Current modding methods are slow, confusing, and require deep familiarity with the Assembly Kit.
- Negan needs an easy-to-use web app that automates calculations and lets him create balanced factions quickly and intuitively.



Customer Problem Statement & Requirements: Graikos (the modder)

Graikos is an experienced modder who balances hundreds of units by hand.

He relies on personal judgment and manual calculations, leading to inconsistent or biased balance values.

The manual workload causes stress, long turnaround times, and user dissatisfaction.

Graikos needs an automated balance calculator that applies objective formulas, speeds up his workflow, and ensures fairness across factions.



System Specifications for Functional, Non-Functional and User Interface

Functional Specifications

- Faction Creation
- Unit Customization
- Calculation
- Data Persistence and Export

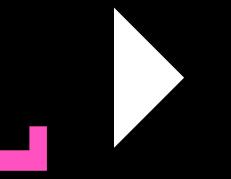
Non-Functional Specifications

- Performance and Usability
- Reliability
- Portability
- Security

User Interface Specifications

- Interface Layout
- Input Controls
- Feedback and Interactivity
- Ease of Use
- Responsiveness

Implementation and Development Tools



Content

- Built using JavaScript for logic and calculations
- Programming: JavaScript (for logic and calculations)
- Storage: Google Sheets tabs used as database tables (Factions, Units, Builds, etc.)
- Front-end: Simple Google Sheets interface (input cells, dropdowns, formulas)
- Automation: Apps Script functions for calculations, validation, and comparisons
- Version Control: Shared Google Drive for collaborative edits
- Testing: Manual data entry and formula validation within Sheets

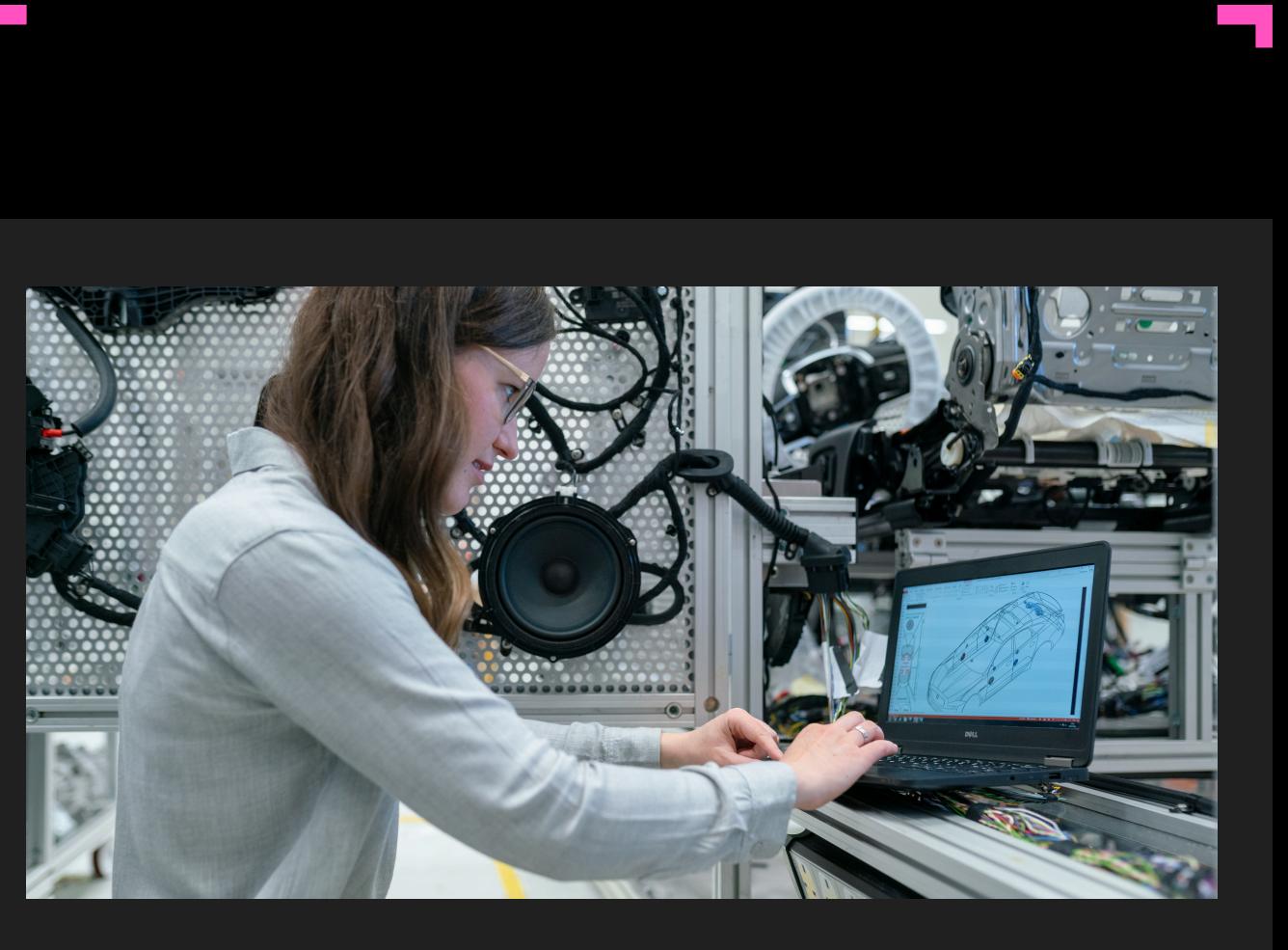
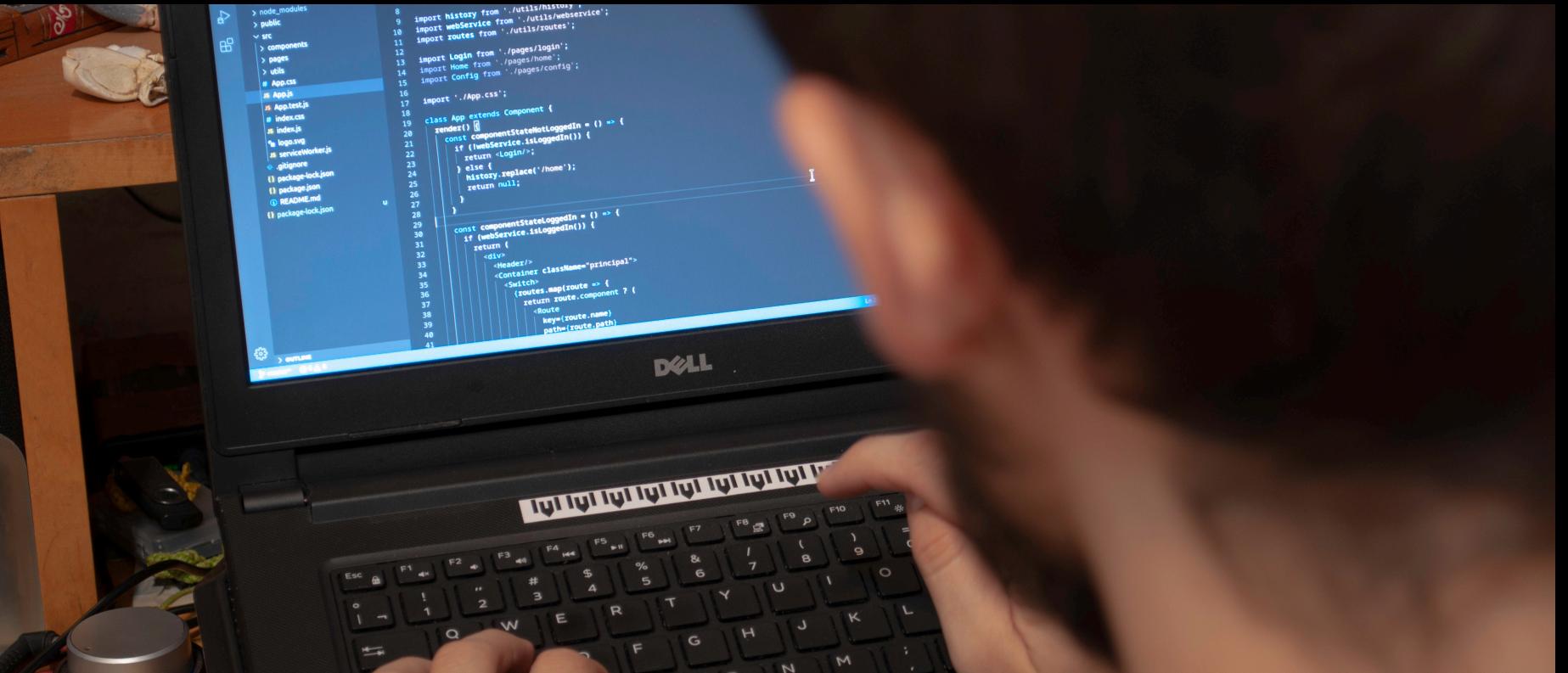
Project Challenges and Future Work

Challenges:

- Account for all users inputs
- Keeping calculations accurate when large amounts of data are entered
- Making the sheet simple to use while still powerful
- Limited visual customization
- Ensuring the JSON data stays consistent and doesn't conflict between updates

Future Work:

- Allow players to change unit visuals and edit unit abilities
- Allow online access for viewing builds without editing rights
- Improve performance for larger datasets
- Integrate export options to external formats (like CSV or JSON for mod import)
- Implement more automated testing features



- The Mod Calculator Functionality
 - User Friendly Web Application
- Future Plans

Conclusion