Wizard Quest

Sprint 1 presentation

Team members:

- Emre Acarsoy
- Luca Croci
- Tom Croft
- Will Finney
- Kazybek Khairulla
- Luca Pacitti
- Harry Taylor

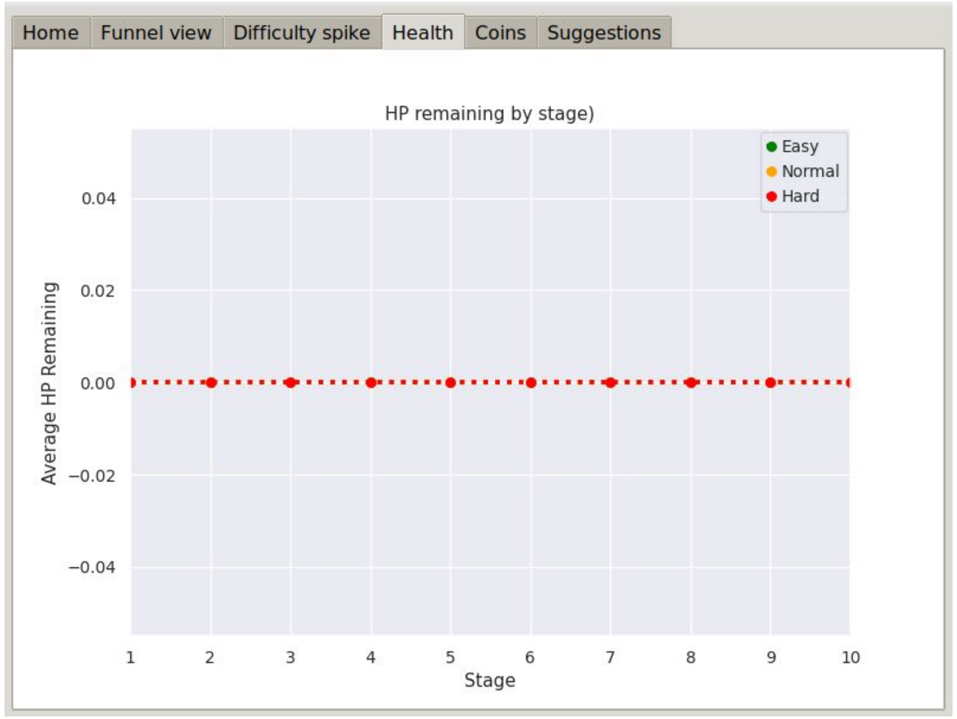
Overview

Roles

Emre Acarsoy	Telemetry Lead | Java Developer
Luca Croci	UI & UX Lead
Tom Croft	Project Lead / Scrum Master | Lead Designer
Kazybek Khairulla	Java Developer
Luca Pacitti	Testing Lead | Java Developer
Harry Taylor	Telemetry developer | Java Developer
Will Finney	Java Developer | Gameplay Designer

Problem Statement

- Game balancing is difficult for designers.
- They often have to rely on their own intuition and small-scale playtests.
- This makes it difficult to identify problem areas where players lose engagement.
- What if we had live analytics generated whilst players engage?



Approach and design:

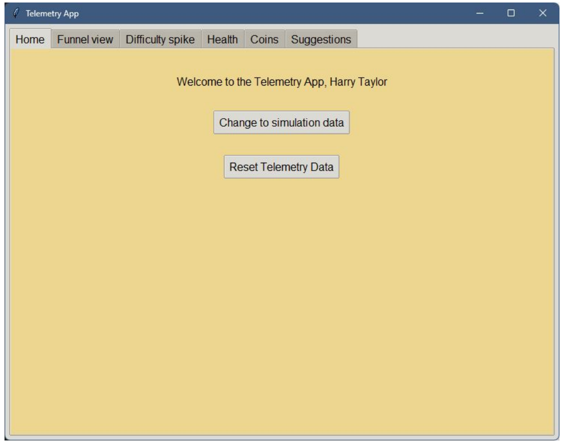
A tale of 2 applications

The game

- Wizard Quest, the game the users can play or simulate.

The telemetry visualiser

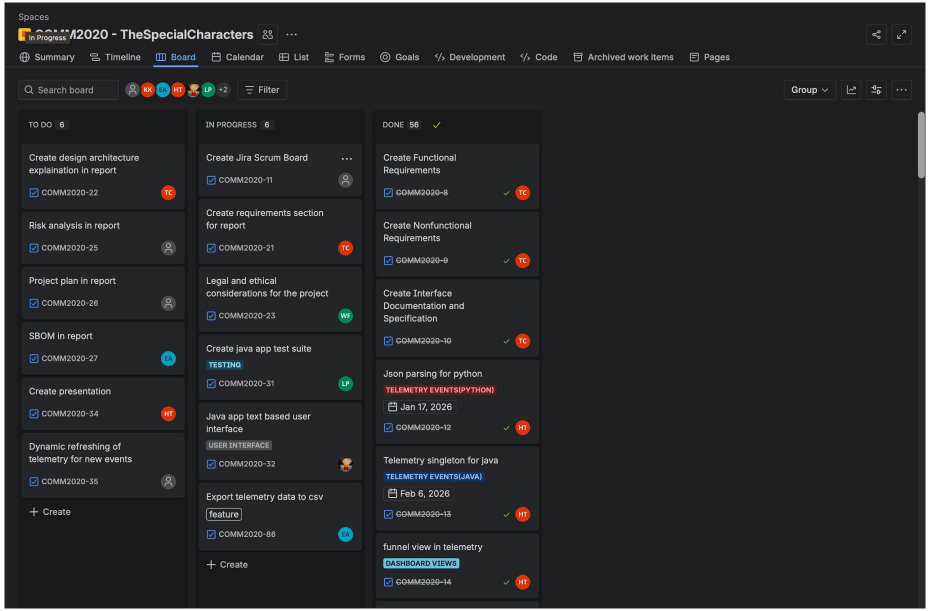
- Telemetry application, used by the game designers.



Welcome to the Telemetry App, Harry Taylor

Scrum Board:

Jira tasks, stories, epics



Spaces

COMM2020 - TheSpecialCharacters

In Progress

Summary · Timeline · Board · Calendar · List · Forms · Goals · Development · Code · Archived work items · Pages

Search board · KK EA HT LP +2 · Filter · Group

TO DO 6

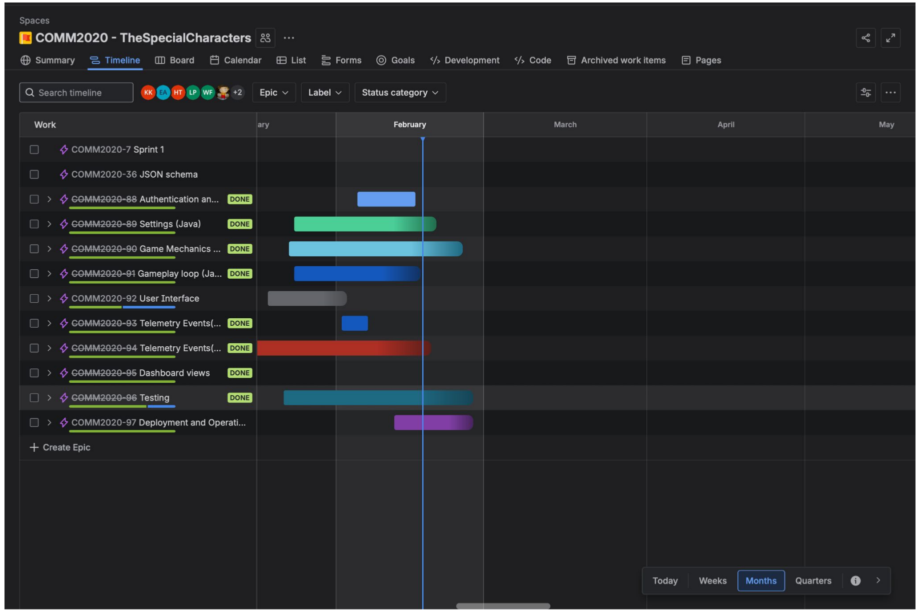
- Create design architecture explaination in report — COMM2020-22 — TC
- Risk analysis in report — COMM2020-25
- Project plan in report — COMM2020-26
- SBOM in report — COMM2020-27 — EA
- Create presentation — COMM2020-34 — HT
- Dynamic refreshing of telemetry for new events — COMM2020-35
- + Create

IN PROGRESS 6

- Create Jira Scrum Board — COMM2020-11
- Create requirements section for report — COMM2020-21 — TC
- Legal and ethical considerations for the project — COMM2020-23 — WF
- Create java app test suite — TESTING — COMM2020-31 — LP
- Java app text based user interface — USER INTERFACE — COMM2020-32
- Export telemetry data to csv — feature — COMM2020-86 — EA
- + Create

DONE 56 ✓

- Create Functional Requirements — ~~COMM2020-8~~ ✓ TC
- Create Nonfunctional Requirements — ~~COMM2020-9~~ ✓ TC
- Create Interface Documentation and Specification — ~~COMM2020-10~~ ✓ TC
- Json parsing for python — TELEMETRY EVENTS(PYTHON) — Jan 17, 2026 — ~~COMM2020-12~~ ✓ HT
- Telemetry singleton for java — TELEMETRY EVENTS(JAVA) — Feb 6, 2026 — ~~COMM2020-13~~ ✓ HT
- funnel view in telemetry — DASHBOARD VIEWS — ~~COMM2020-14~~ ✓ HT



Spaces

COMM2020 - TheSpecialCharacters

Summary · Timeline · Board · Calendar · List · Forms · Goals · Development · Code · Archived work items · Pages

Search timeline · KK EA HT LP WF +2 · Epic · Label · Status category

Work · ary · February · March · April · May

- COMM2020-7 Sprint 1
- COMM2020-36 JSON schema
- ~~COMM2020-88~~ Authentication an... DONE
- ~~COMM2020-89~~ Settings (Java) DONE
- ~~COMM2020-90~~ Game Mechanics ... DONE
- ~~COMM2020-91~~ Gameplay loop (Ja... DONE
- COMM2020-92 User Interface
- ~~COMM2020-93~~ Telemetry Events(... DONE
- ~~COMM2020-94~~ Telemetry Events(... DONE
- ~~COMM2020-95~~ Dashboard views DONE
- ~~COMM2020-96~~ Testing DONE
- COMM2020-97 Deployment and Operati...
- + Create Epic

Today · Weeks · Months · Quarters

Implementation - The Game

Features

- Encounters
- Shop
- Design Parameters
- Damage Types

Components

- Upgrades and Abilities
- Settings
- Telemetry
- Authentication

Implementation - Telemetry

Implementation

- Written in Python
- Data manipulation and visualisation libraries:
  - numpy
  - matplotlib and seaborn
- Dynamically generated game balance suggestions
- 5 views: Health per stage, averages per difficulty, etc…

Reading logs

- Events are read and parsed from JSON file.
- Users must sign in to use Telemetry App.
- Only Designers and Developers may view telemetry.
- Clear error messages for malformed JSON.

Measures of Evaluation

Is the game engaging, featuring interesting player decisions?

The player can choose from many upgrades which can improve their odds of success.

Does the system utilise less than 1GB of RAM?

Only 300mb of RAM is needed

Does the telemetry interface have valid spike detection?

Difficult areas are identifiable as spikes

Is the code product easily useable?

UI is intuitive albeit primitive

Are the telemetry app suggestions useful and easily actionable?

Suggestions are actionable, but lack precision as of limited changeable parameters

Demo

Limitations of Sprint 1

The game

- Currently only a CLI, not GUI yet.
- Only 2 stages out of 10 (short gameplay loop)
- Limited variety of enemies, no bosses during encounters
- Only 1 design parameter (starting lives) is adjustable by designers

The telemetry app

- Python Test suite incomplete
- CSV export is not functional
- Not all dashboards implemented
    - Progression Curve
    - Comparison mode
    - Fairness indicators

Looking ahead at Sprint 2

The game

- GUI
- More encounters (and bosses!)
- More comprehensive test suite

The telemetry app

- Additional data views
- Additional dynamic balance suggestions
- Profile page
- More comprehensive test suite