In Verbis Virtus

Game Player Analysis

Simone Caglio - a.a. 2023-2024

POWERED BY UNREAL TECHNOLOGY

BI&BDA

UNIVERSITA' DEGLI STUDI DI MILANO BICOCCA

Context

In Verbis Virtus is a fantasy-themed video game developed by the independent production company Indomitus (formerly Indomitus Games) and published in 2015 and still present on the Steam store.

One of the main innovative features of the game was the introduction of the ability to cast spells through the voice, a fact that led to the winning of the "Drago d'Oro" award in the Game Design category.

The development team, to celebrate 10 years since the first publication, is evaluating the opportunity to create a new chapter or a re-release in 2025.

Goal

The analysis project presented here was requested by the development group to analyze the players' gameplay and be able to make improvements to the game mechanics.

DRAGO D'ORO
Best Game Design

Process

The process involves extracting data from player logs and transforming them so that they can then be explored and analysed, reconstructing the game mechanics of different users.

In Verbis Virtus - ETL

Homework 1

Python notebook available at GitHub Repo (https://github.com/SimoneFisico/Master_BIBDA/tree/main/Homework_BDA)

Tools, Technologies and Architecture

TOOLS	VS code (Python notebooks), Valentina Studio (Client SQL), BB Edit (Text editor)
STORAGE	PostgreSQL (Sources and staging), CSV and JSON (fact)
LANGUAGES	SQL, Python (main libraries Pandas, Numpy, psycopg2)
ARCHITECTURE	On premise with PostgreSQL on Docker container

Technical and methodological choices

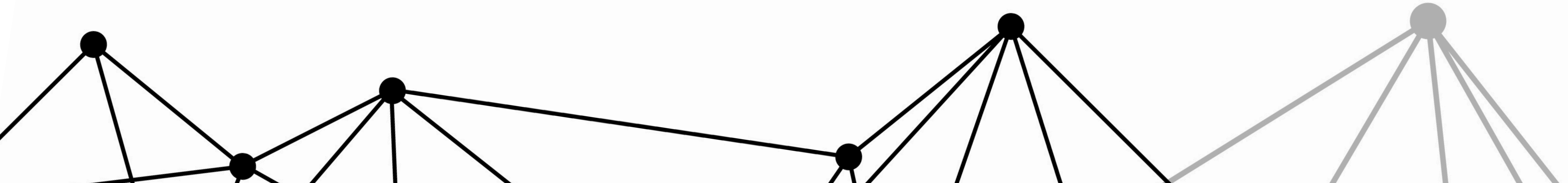
This project was born from a real need to explore the data recorded over almost 10 years to verify the scope of information contained in them.
The recorded data are logs from actual game sessions played by users who purchased the game.

Technical requirements set by the development team:

- local reproducibility of the system
- possibility of replicating all the individual steps
- no limitations on the choice of languages
- no limitations on the type of data storage as long as portability is guaranteed
- source codes

What we know for sure about the data :

- the dataset will not be further updated (the logs have been deactivated)
- it is very likely that some logs have been lost (not recorded) due to various lag problems between the player's client and the logging server
- any missing data is not recoverable (unrecorded logs are lost)



Starting data

The starting point was a SQL dump of a relational database where player logs are recorded.

The dump was then imported into a PostgreSQL DBMS in a Docker container set up locally.

The basic structure had only 4 fields:

player_id	session_id	log_id	data
the anonymized player identifier	the game session identifier	the log registration identifier	the data recorded in the log
0x01100001009ABF25	2	15	InitialPosition 6.8056 0.2367 22771.77,3085.72,-2672.39 65313,-65863,-4

A few numbers...

- The database is composed of more than 14 M records
- There are 3825 unique players
- 24 different types of logs recorded

As a pilot project to define the utility of the analysis,
500 unique player were processed
for a total of **296789** records.

Data issues

- **An encoding problem for some characters (such as commas, underscores, colons, etc.) was detected with respect to the SQL dump format.**
  It was solved with a replacement through text editor.
- **The data field contains many different information with different structures.**
  It was necessary to sample and analyze all the different types of recorded logs.
- **Some data, after the structure recognition, needed correct formatting.**
  The type casting was done at the most appropriate time, for some "simple" fields from the transition between source table and staging table, for multiple fields, in the creation of fact tables.
- **The meaning of some fields extracted from the log data are not clearly understandable.**
  They have been recorded and will be analyzed together with the developers.

Log data structure

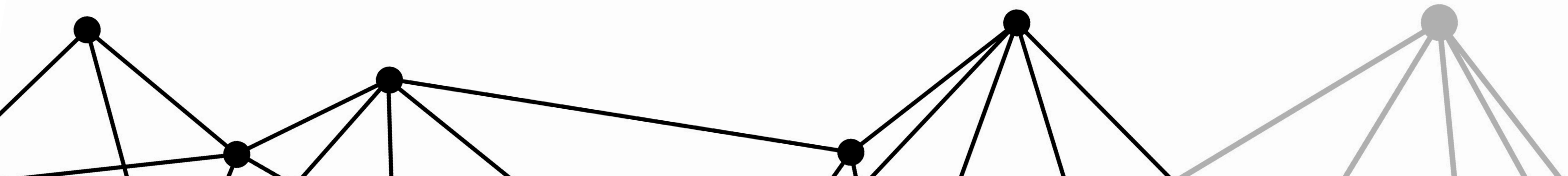
- The first "word" of the log is the type of event recorded
- 24 different types of events have been recognized:
  - all have specific information at the end of the log
  - except BeginLog and EndLog, all have real time and game time as second and third elements
  - a subset of 16 events has information (in the form of triplets) on position, rotation and velocity

Staging table

The staging table was then created by maintaining the original fields (player_id, session_id and log_id) and exploding the data present in the original data field, separating both the common and specific information.

The separation was performed using SQL commands via SQL GUI Client, populating a staging table used as a support for subsequent treatments.

game_time	real_time	event_type	position	rotation	velocity	health	other specific fields...



New data structure

Since each log had specific fields that were different from each other, a document structure was initially chosen, managed through Python dictionaries, in which to collect for each user the information of the different sessions and for each session the steps with position, rotation and speed, with which to reconstruct the paths taken.

Among the session information, the information of the reference map was extracted (present only in the BeginLog type log), the distance traveled, the total number of logs, the game time and the real time

player	session	session_logs	map_info	rotatimax_real_timeon	velomax_game_timecity	distance

Inside the session field, a list stores the event types that have the position, rotation and velocity properties (in addition to the specific fields of the event type), so as to be able to reconstruct the path taken by the player.

The choice of using a document format allows you to exclude those fields that, not being specific to the property, remain Null in a more structured format (as happens in the staging table).

Fact tables

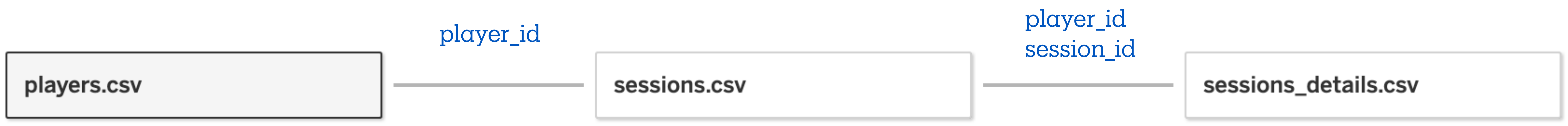
To generate the final tables, the document format was saved as as a JSON file in order to be imported and managed in a document-oriented DBMS, such as MongoDB.

However, the generated JSON file struggled to be managed by the chosen data visualization software (Tableau, see below), so CSV files were created with a relational structure, although maintaining the non-normalized data.

Three CSV files were generated:

- players.csv, with the list of unique player identifiers (useful for future enrichment of information related to the specific player)
- sessions.csv, with the list of sessions for each user and the related information
- sessions_details.csv, with the list of logs per session

CSV files still have the problem of empty fields for those properties that are not specific to the record, especially for session details (logs).

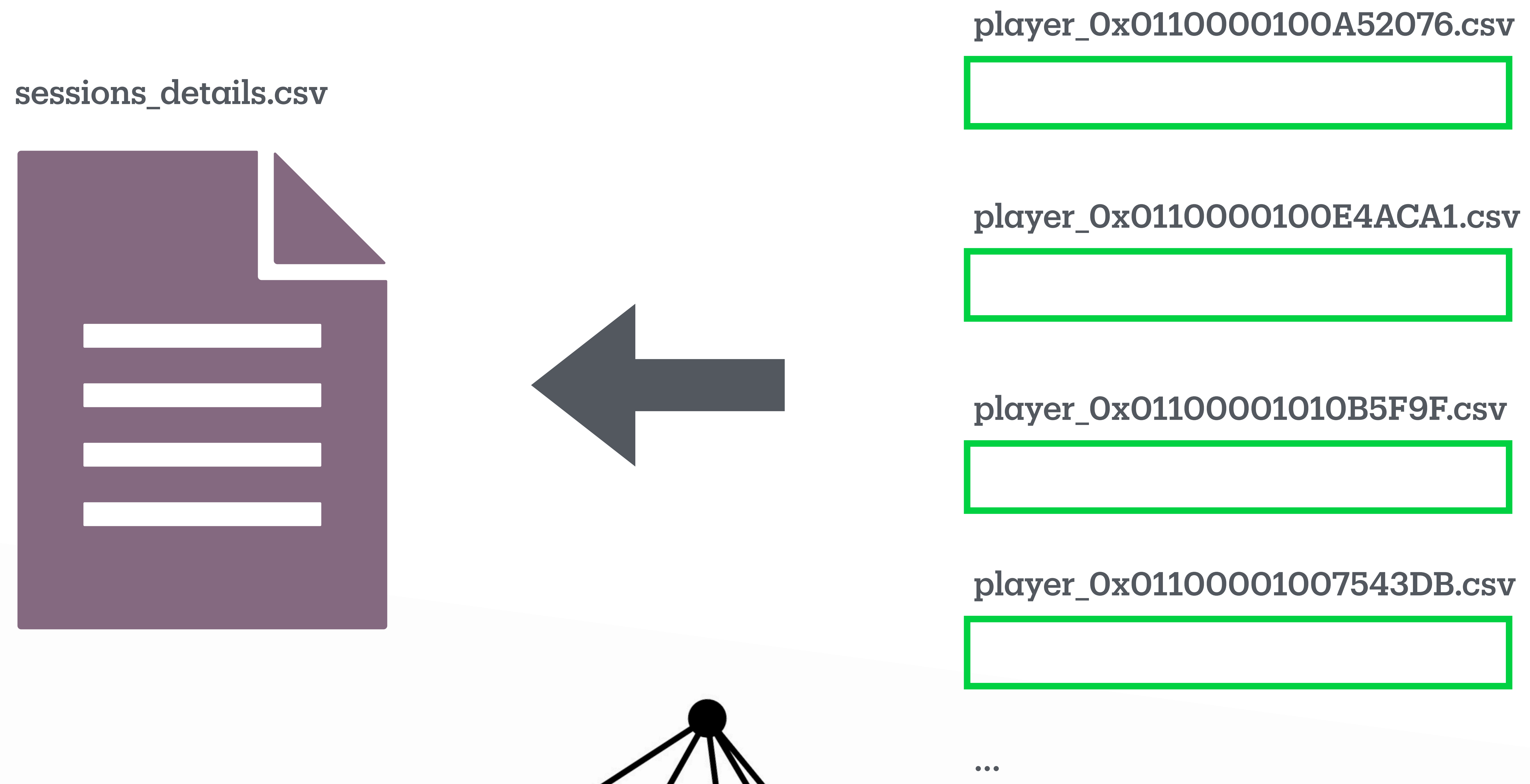


Fact tables

Given the future need (homework 2) to display the path data of each user and to limit the memory load, for each player a CSV file was created containing the list of logs that have position, rotation and speed.

In this way it is possible to load only the necessary data, limiting the resources of the local system.

The sessions_details.csv file is therefore the concatenation of the individual player_<player_id>.csv files.


sessions_details.csv
player_0x0110000100A52076.csv
player_0x0110000100E4ACA1.csv
player_0x01100001010B5F9F.csv
player_0x01100001007543DB.csv
...

Improvements - Data

- Event types without position, rotation and velocity are not currently saved in CSV files, while they are in JSON files. Specific storages can be created for further analysis together with Indomitus developers. (They are always available from the staging table)
- Meaningless fields can be made more explanatory after analysis with the development team.
- Position, velocity and rotation data can be recalibrated to the size of the game maps and referenced to specific locations.
- Rename players for easier recognition

In Verbis Virtus - Data Visualization

Homework 2

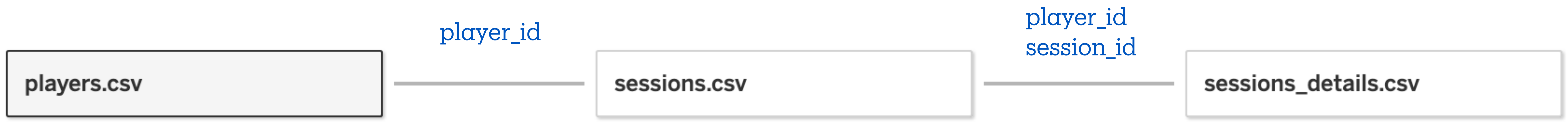
Python notebook available at GitHub Repo (https://github.com/SimoneFisico/Master_BIBDA/tree/main/Homework_BDA)
Dashboards available at Tableau Public (https://public.tableau.com/views/IVV-Analytics/Genstats)

Tools, Technologies and Architecture

TOOLS	VS code (Python notebooks), Tableau Public
STORAGE	CSV(fact tables)
LANGUAGES	SQL, Python (main libraries Pandas, Numpy, Matplotlib)
ARCHITECTURE	On premise for Python notebooks, Cloud for Tableau Public

Data for visualization

For the visualization phase, the CSV files generated in the ETL phase (Homework 1) were used, which were already formatted for this purpose, according to the structure:



Requested information

The Indomitus development team made both very specific requests regarding the base statistics to be extrapolated, and issued challenges regarding the possibility of "mimic" the user's gameplay to understand what possible problems they encountered during gaming sessions.

The choices

In order to respond to the requests made and considering that the interlocutors are developers used to "getting their hands on the code", two different paths were taken:

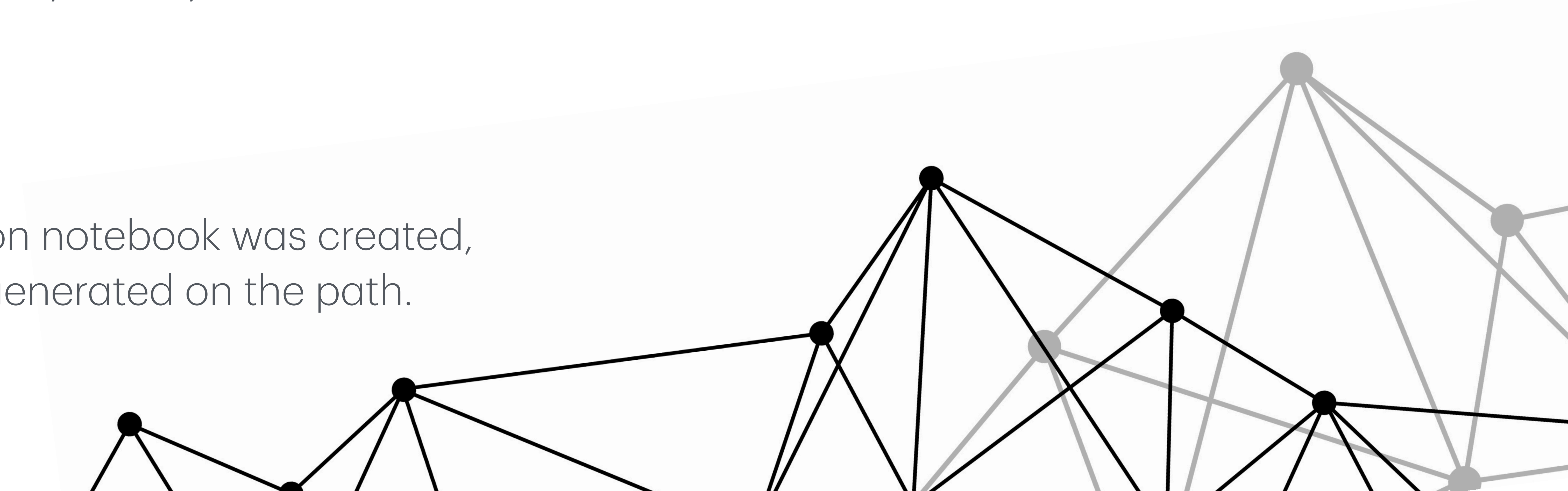
- the use of a tool for data visualization and the creation of dashboards
- the preparation of some visualization functions in a Python notebook

Dashboards for stats

To respond to the request for general and single-user statistics in which to report general information on the number of games played (sessions), their duration, the level (map) played and so on, in agreement with the developers, Tableau was chosen for the creation of the dashboards, Tableau with the possibility of exporting the data to the web (Tableau Public) for easier sharing.
Two different dashboards were therefore created, one that collects general numbers in tabular format (General stats) and a more specific one that displays the data for each player (Player stats).

Python for paths

To visualize the paths taken by the players, a Python notebook was created, which also indicates the different types of events generated on the path.



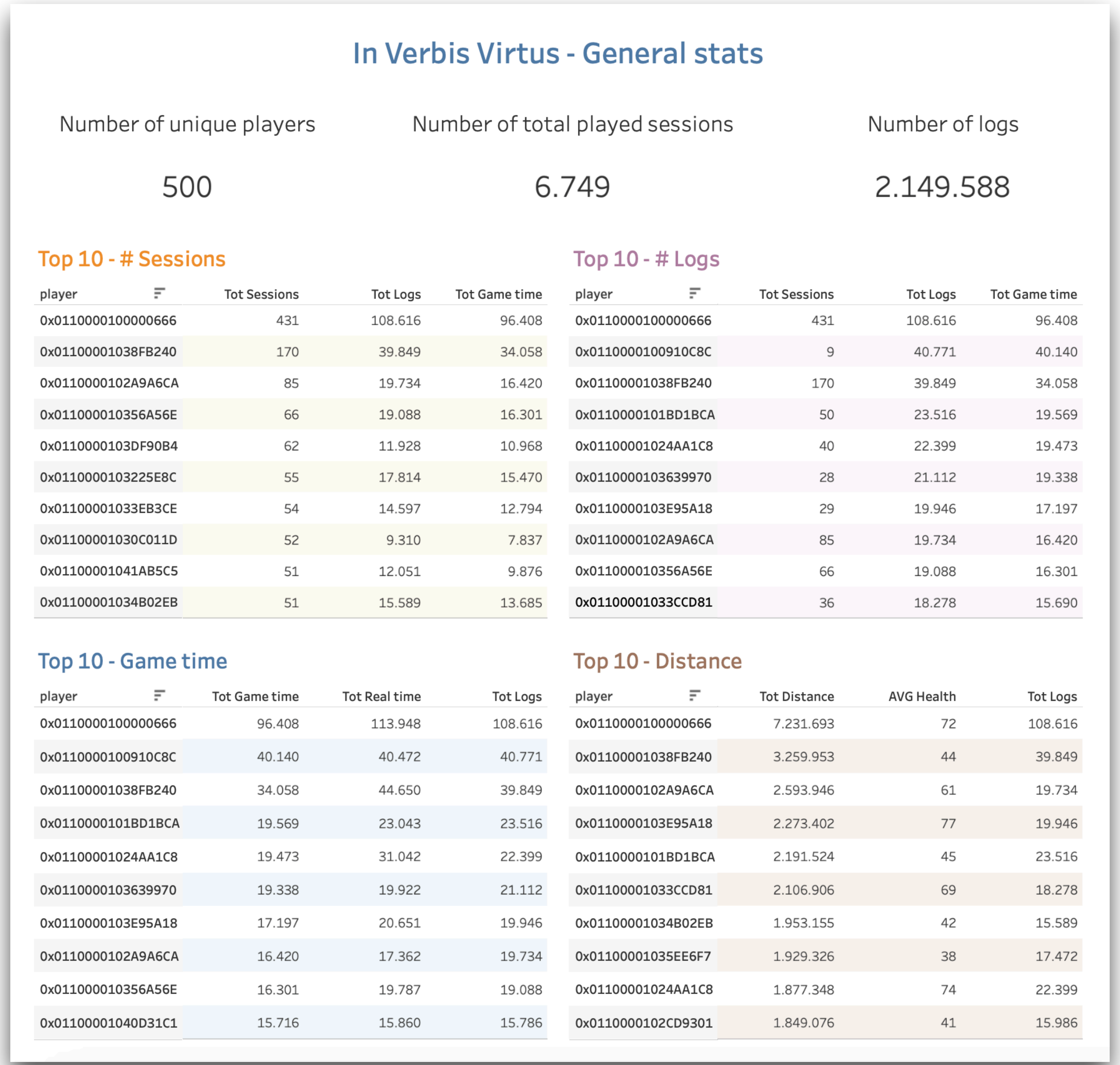
General stats

The general statistics dashboard shows the number of unique players (limited to 500 in this first phase of study), game sessions and recorded logs.

There are then four tables that show the top 10 players by number of sessions, number of logs, game time and total distance traveled during the sessions.

This first dashboard is used to give a simple overview of how users have approached the game.

The data presented here was specifically requested by the development team.



In Verbis Virtus - General stats

Number of unique players	Number of total played sessions	Number of logs
500	6.749	2.149.588

Top 10 - # Sessions

player	Tot Sessions	Tot Logs	Tot Game time
0x0110000100000666	431	108.616	96.408
0x01100001038FB240	170	39.849	34.058
0x0110000102A9A6CA	85	19.734	16.420
0x011000010356A56E	66	19.088	16.301
0x0110000103DF90B4	62	11.928	10.968
0x0110000103225E8C	55	17.814	15.470
0x01100001033EB3CE	54	14.597	12.794
0x01100001030C011D	52	9.310	7.837
0x01100001041AB5C5	51	12.051	9.876
0x01100001034B02EB	51	15.589	13.685

Top 10 - # Logs

player	Tot Sessions	Tot Logs	Tot Game time
0x0110000100000666	431	108.616	96.408
0x0110000100910C8C	9	40.771	40.140
0x01100001038FB240	170	39.849	34.058
0x0110000101BD1BCA	50	23.516	19.569
0x01100001024AA1C8	40	22.399	19.473
0x0110000103639970	28	21.112	19.338
0x0110000103E95A18	29	19.946	17.197
0x0110000102A9A6CA	85	19.734	16.420
0x011000010356A56E	66	19.088	16.301
0x01100001033CCD81	36	18.278	15.690

Top 10 - Game time

player	Tot Game time	Tot Real time	Tot Logs
0x0110000100000666	96.408	113.948	108.616
0x0110000100910C8C	40.140	40.472	40.771
0x01100001038FB240	34.058	44.650	39.849
0x0110000101BD1BCA	19.569	23.043	23.516
0x01100001024AA1C8	19.473	31.042	22.399
0x0110000103639970	19.338	19.922	21.112
0x0110000103E95A18	17.197	20.651	19.946
0x0110000102A9A6CA	16.420	17.362	19.734
0x011000010356A56E	16.301	19.787	19.088
0x01100001040D31C1	15.716	15.860	15.786

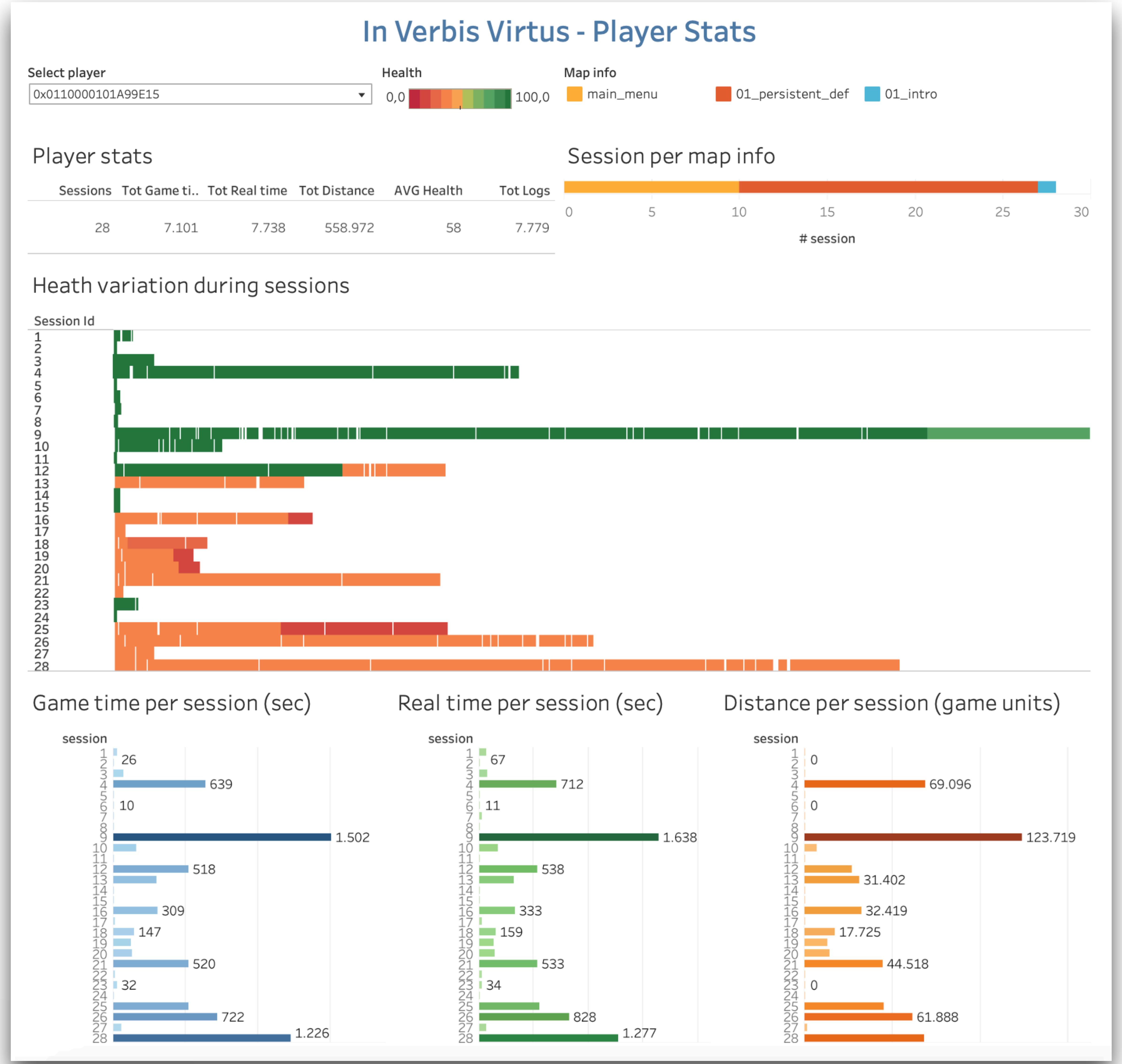
Top 10 - Distance

player	Tot Distance	AVG Health	Tot Logs
0x0110000100000666	7.231.693	72	108.616
0x01100001038FB240	3.259.953	44	39.849
0x0110000102A9A6CA	2.593.946	61	19.734
0x0110000103E95A18	2.273.402	77	19.946
0x0110000101BD1BCA	2.191.524	45	23.516
0x01100001033CCD81	2.106.906	69	18.278
0x01100001034B02EB	1.953.155	42	15.589
0x01100001035EE6F7	1.929.326	38	17.472
0x01100001024AA1C8	1.877.348	74	22.399
0x0110000102CD9301	1.849.076	41	15.986

Player stats

In the player statistics dashboard, you can select the chosen player and view the data related to him.

In addition to a series of overall numbers that follow what was already seen in the previous dashboard, the sessions are displayed divided by reference map (at the top), the health variation during the progress of the different sessions (in the center) and the distributions of game time, real time and distance traveled for each session (at the bottom).



In Verbis Virtus - Player Stats

Select player: 0x0110000101A99E15

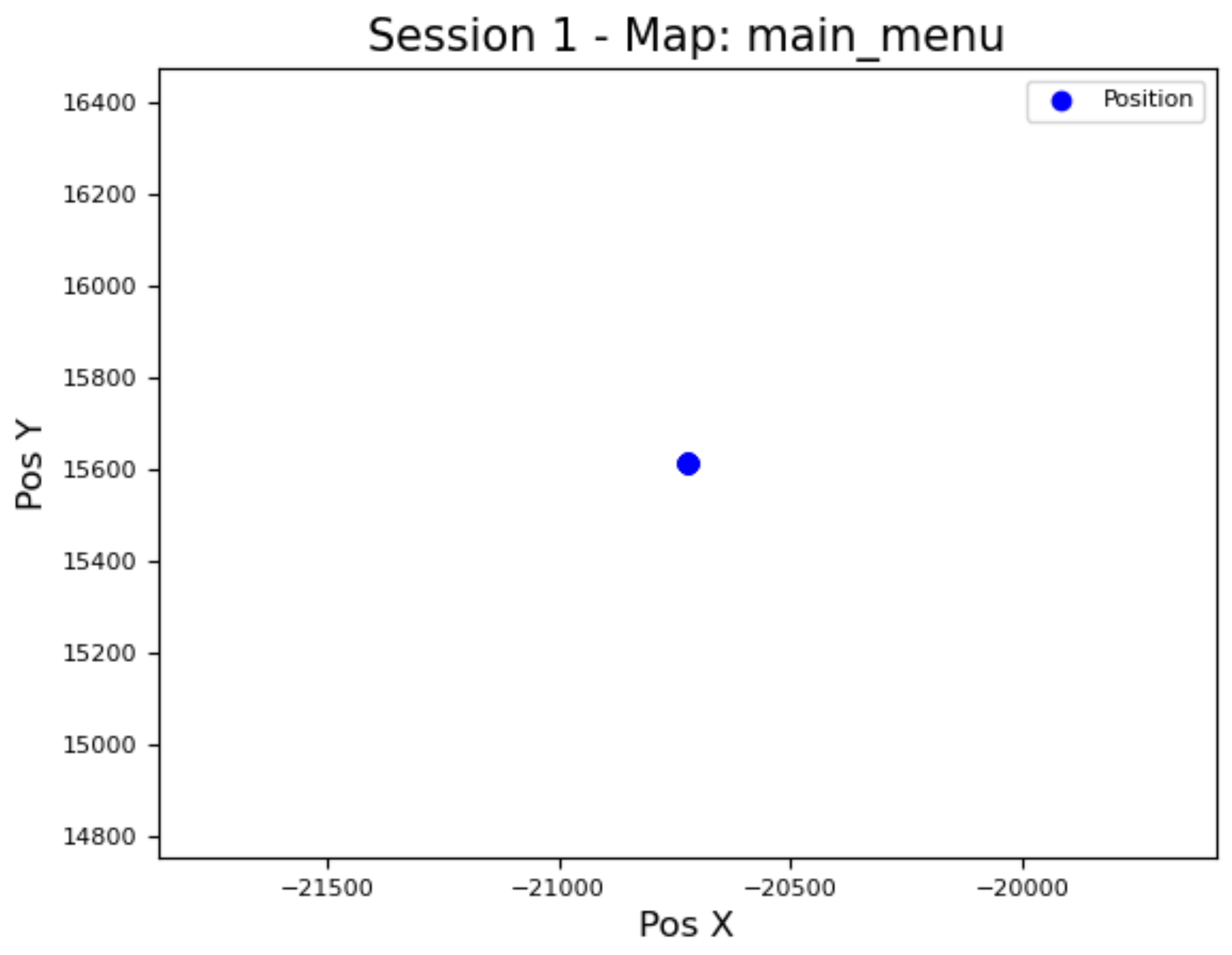
Player stats

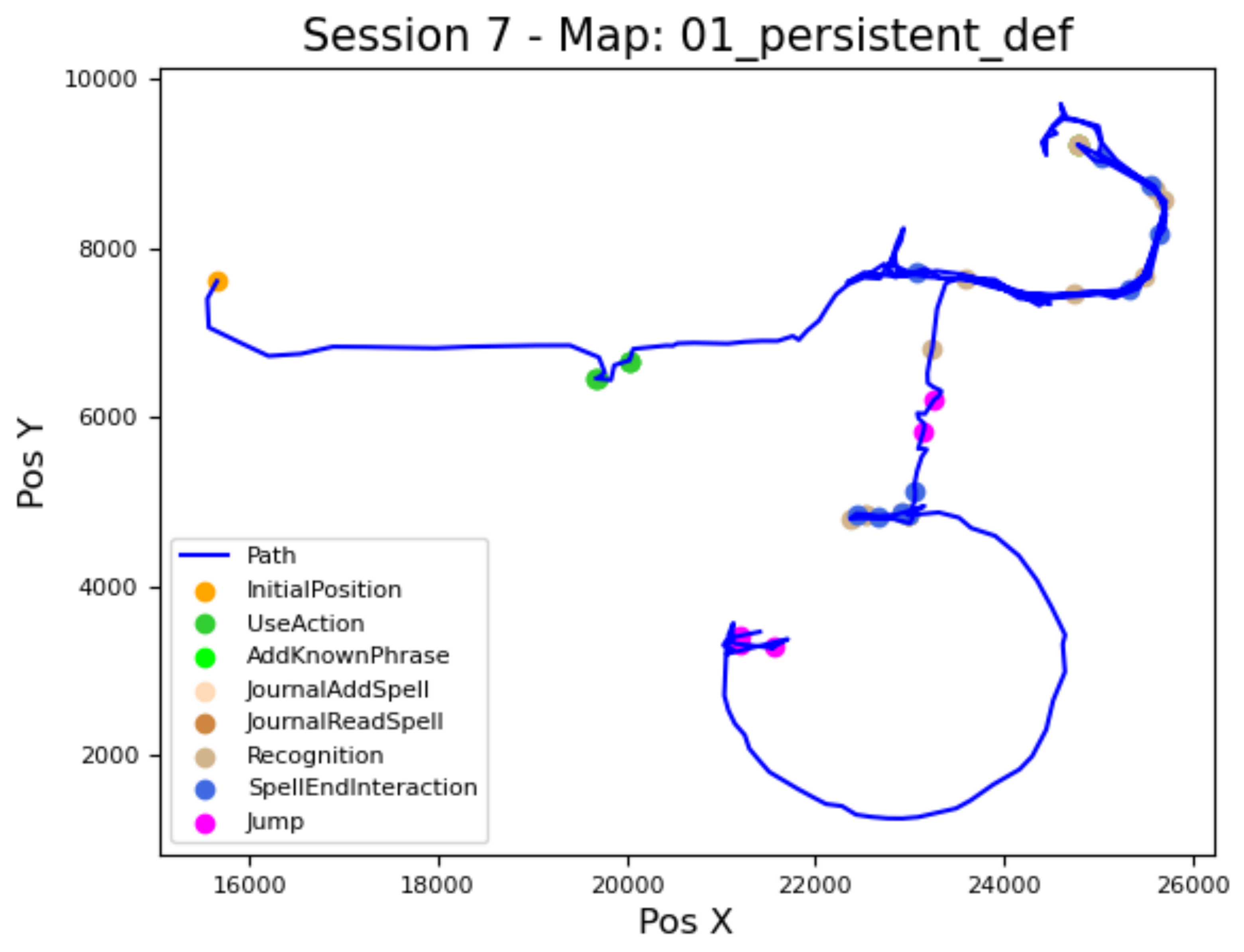
Sessions	Tot Game ti..	Tot Real time	Tot Distance	AVG Health	Tot Logs
28	7.101	7.738	558.972	58	7.779

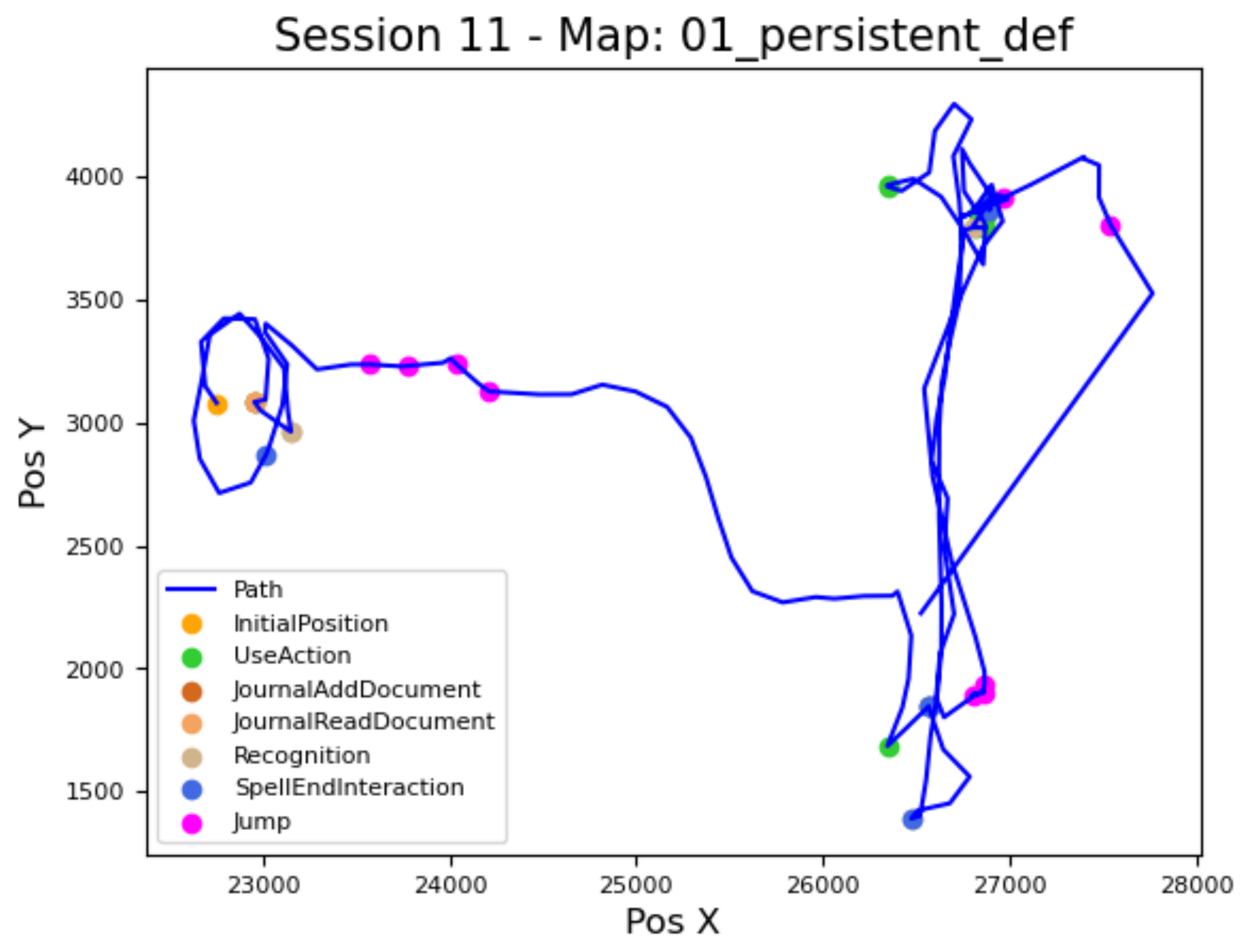
Experimentation for the reconstruction of the paths

In order to reconstruct the path taken by the user within the various game levels, divided by session, a Python notebook has been prepared that allows you to select the specific player and reconstruct the path in one or more sessions in graphic format. The different types of events that occurred are also identified along the path with different colors.

Where in a session (as in Session 1) there are no real movements, the player's position is identified with a point.







Goals achieved

General

- Effective recognition of the value of the collected data for stakeholders

Base statistics

- Displaying game statistics as per developers' needsInsert other data, with respect to the needs of the developers
- View statistics for each user to reconstruct the game phases

Challenges

- Reconstruction of the pawn’s path
- Making a notebook available to developers

Improvements

Dashboards

- Ability to directly compare the statistics of two or more players
- Insert other data, with respect to the needs of the developers, like interaction language (for voice commands) or reasons for the pawn's death

Path reconstruction

- Use interactive graphics with zoom and tooltips on different points
- Overlap paths with game maps
- area of the map displayed during the route

Other analyses

- Location-free log analysis (after interpreting data with developers)

In Verbis Virtus

Simone Caglio - a.a. 2023-2024