## Notes on extracting data for a single student from the database - Incomplete

Here is a message from Yannick (Yannick.Lefebvre@cm-labs.com):

"*I have extracted data for a single student out of that sample database in a CSV format. As you'll see, that represents just over 448000 records.*

[*https://u.pcloud.link/publink/show?code=kZEffHXZRbo0YsSGkfLcCQa7izYcG4KNIpv7*](https://u.pcloud.link/publink/show?code=kZEffHXZRbo0YsSGkfLcCQa7izYcG4KNIpv7)

*The legend for the CSV fields is in the first line of the file:*

*session\_id, start\_time, equipment name, exercise name, metric name, timestep, value, unit*

*I am asking for full labels of the names of the metrics and exercises instead of these internal variable names.*

*All of the metrics that have less entries are still useful data. It's just that our system only writes these values at a different frequency.*

*Here is a CSV file containing the scoring rules that were used for all of the sessions in the current data set.*

[*https://drive.google.com/file/d/17ppQGCikgk0Q2-IBjNh9qHkkSgGPBQrh/view?usp=sharing*](https://drive.google.com/file/d/17ppQGCikgk0Q2-IBjNh9qHkkSgGPBQrh/view?usp=sharing)

*The fields are:*

*session id, metric name, minimum value, maximum value, deduction*

*Essentially, when the max value is exceeded, the system deducts the number of points in the deduction column. In some cases, it deducts points if the value is not between the min and max at the end. The session id is the link to associate these rules to the session since they may have been changed between sessions.*

*For the pass/fail score, that status is not determined by the system, it is determined by a live instructor, based on the score and the exercise performance observed. It is therefore a subjective rating. The customer that we got this data from mostly has units without instructor stations, so the pass/fail indication is pretty much never recorded. We are working on getting data with this score filled in*.”

However, it turns out that it is not practical to obtain pass/fail data from the instructor for each completed scenario as there are too many students, so we need to rely on the automated scoring system.

However it might be practical to ask the instructor to rank the students at the end of the day.

**QUESTIONS**

Q. Why the result column only has None and not Pass or fail entries in it and what does current trainee score at that time(ArcSwipe List of Metrics - Sheet1.csv), etc means, and what is the name of that column.

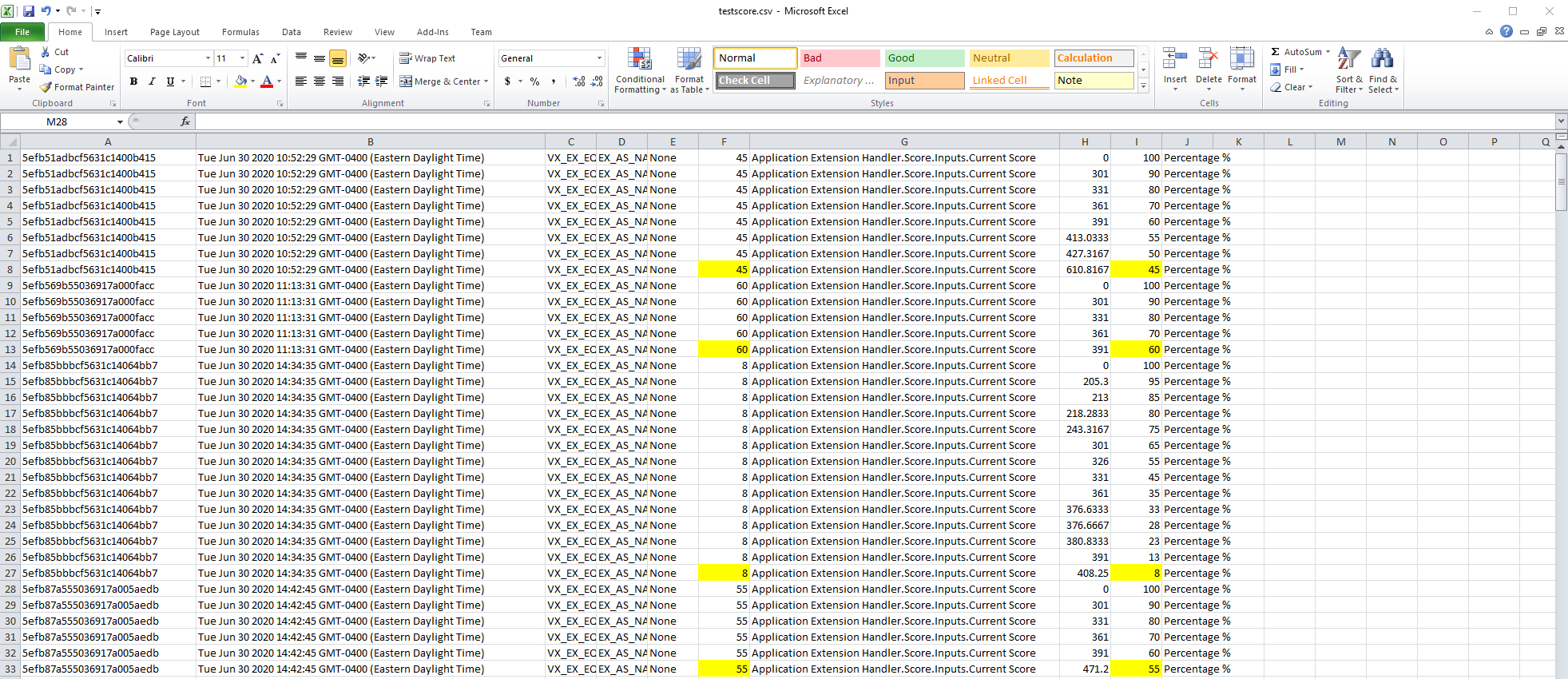
A. An indication of none means that the session was run on a system that is not attached to an instructor station, which is mostly the case with this customer that we got data from. In that situation, since there is no instructor, the system indicates no status, instead of having the instructor assign a pass or fail status.

For the Current trainer score at that time, it indicates the current score, out of a 100%, that the trainee has at that point in time. The records that I provided are recording frequently as the simulation takes place, with the timestamps indicated as part of the data fields on each line. When you get a field with the metric name "Application Extension Handler.Score.Inputs.Current Score", it is the current score that the trainee has at that time in the exercise.

Q. why the starting score is 45 for each metric because the score goes down to 0 so why beginning at why

A. The score column is not the instantaneous score at that time, it's the final score. You might want to disregard it just like equipment name and exercise name.  
All the metrics such as Application Extension Handler Score.Inputs.Current Score, ContentExtensionScene.CONS\_PERF\_MET\_CAT.Inputs.CONS\_PERF\_MET\_PATH\_TIME\_CURRENT etc start and end on same utc date-time so i think all these are initiating in parallel instead of sequential initiation (shown in the csv) so i will be restructuring it in the manner that each metrics is a column(as all have same start time) instead of a row by row

I took a closer look at the data, isolating only the lines for the metric "Application Extension Handler.Score.Inputs.Current Score" and am seeing a correlation between the score column and last metric for the Current Score (see attached image). The score column is not the instantaneous score at that time, it's the final score. You might want to disregard it just like equipment name and exercise name.



Q. what does timestep column exactly means a value of **301, 413** etc in timestep represent what ?

1. I believe this is the number of updates of the metric, some are updated on a defined frequency (every 60 frames for example which equals to roughly 1sec)

Q. For units-> value column where one gets the values for the units what does the value for time represent for example time has values 16.8, 14.3, 17.2 etc are these seconds, mins, hrs etc or some of your defined time field .

The 10th column indicates the units for the value column.

Q. For metrics around 26 of them have same number of entries as 23856 for each but the rest have somewhere between 500-40 entries. Can you please tell me which of these are not required so that i can remove those. If all are required i would need to oversample those metrics to become at least 20-30% in number of entries as compared to others,

1. this is related to the way the metrics are updated, some are updated every 60 frames (as explained before) some are updated when the value changes so the count is for sure lower than the previous category, we can see that the GOALS metrics equals 398, I am sure there are no exercise with that amount of goals but this might also help to understand what this numbers means. If we knew what the goal count was at that moment, we could maybe extrapolate a linear rule to calculate the actual metric (hoping that it is linear).  
   All of the metrics that have less entries are still useful data. It's just that our system only writes these values at a different frequency. You could fill in the column with the value from one timestamp for all later timestamps until there is a new entry for that value (What I'm describing might just be oversampling).

**CLARIFICATION**

Q. All the metrics such as Application Extension Handler Score.Inputs.Current Score, ContentExtensionScene.CONS\_PERF\_MET\_CAT.Inputs.CONS\_PERF\_MET\_PATH\_TIME\_CURRENT etc start and end on same utc date-time so i think all these are initiating in parallel instead of sequential initiation (shown in the csv) so i will be restructuring it in the manner that each metrics is a column(as all have same start time) instead of a row by row

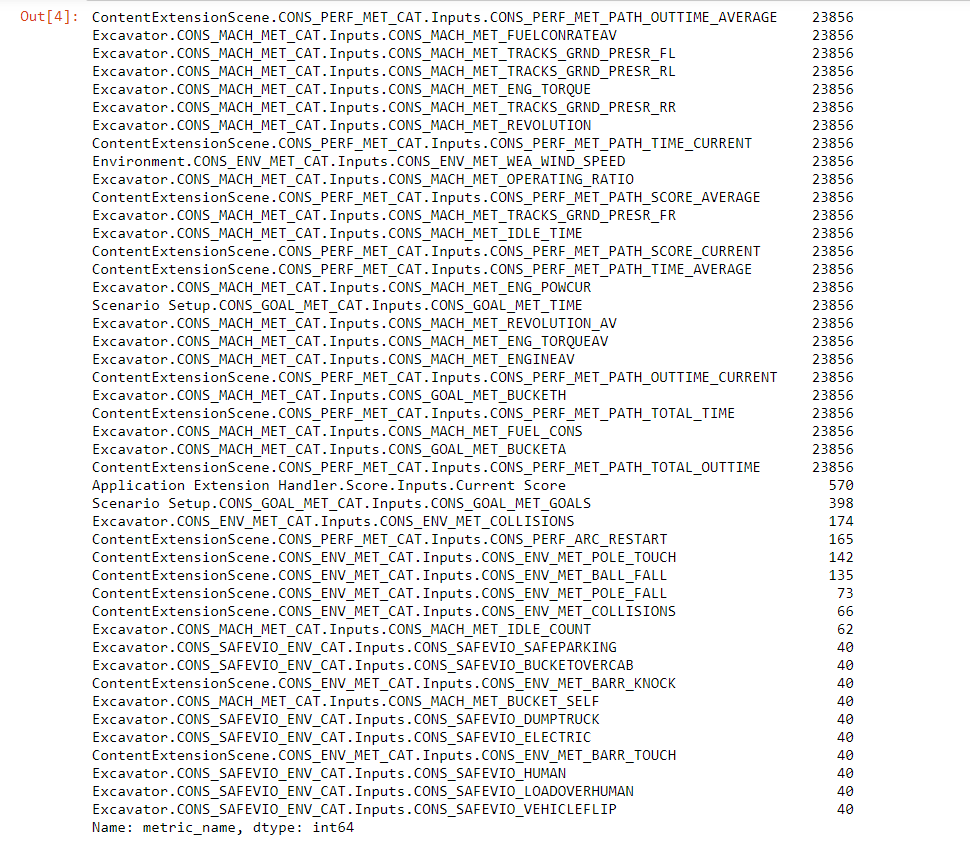
1. Yes, we have them in a bunch of rows, but they can definitely be rearranged as multi-column rows where the timestep matches.

Q. equipment\_names, exercise\_name and passfail columns have the same names for all the rows so will not be useful for training althoug will leave it as it is in the dataset for future expansion on this project.

1. Indeed, these are not useful at this point. I left them there since my first query were to pull all records of a user, who could have done different exercises, but now I pulled all sessions for a single exercise.



List with number of entries:



## Scoring Rules

Below is a CSV file containing the scoring rules that were used for all of the sessions in the current data set. The fields are:

session id, metric name, minimum value, maximum value, deduction

Essentially, when the max value is exceeded, the system deducts the number of points in the deduction column. In some cases, it deducts points if the value is not between the min and max at the end. The session id is the link to associate these rules to the session since they may have been changed between sessions.

For the pass/fail detail, as indicated before, that status is not determined by the system, it is determined by a live instructor, based on the score and the exercise performance observed. It is therefore a subjective rating. The customer that we got this data from mostly has units without instructor stations, so the pass/fail indication is pretty much never recorded. I cannot make assumptions on pass or fail other than to look at the numeric values here since I did not witness the training sessions.

<https://drive.google.com/file/d/17ppQGCikgk0Q2-IBjNh9qHkkSgGPBQrh/view?usp=sharing>

## MongoDB server setup

MongoDB server was installed.

(the version I used can be found here: <https://drive.google.com/drive/u/0/folders/18BLdfMhxI0KEokA15o-MMxBFfgN4r9aQ>

)

PATH=C:\tools\MongoDB\Server\4.2\bin:%PATH%

start C:\tools\MongoDB\Server\4.2\bin\mongod.exe --config S:\MongoDB\mongod.cfg

(batch and cfg files in <https://drive.google.com/drive/folders/18BLdfMhxI0KEokA15o-MMxBFfgN4r9aQ> )

My database was in S:\MongoDB\DB42Anonymous

Robo3T (a MongoDB database explorer tool, also in the folder above), was pointed to DB42Anonymous on “localhost:27017”.

On the command line, in Robo3T , one can do mongoexport to get the data into a text file.

### Extracting sessions

Running the following script in Robo 3T interface. It opens a few different tabs in the outputs section. The last lines do a text dump with the CSV data.

Not necessarily the most efficient, but it works:

var user = db.getCollection('users').find({ name: { $eq: "student4" }}).limit(1).toArray()[0]

var sessions = db.getCollection('sessions').find({ users\_id: { $eq: user.\_id } } ).toArray()

var numberOfSessions = sessions.length

var sessionIDs = new Array()

var keyedSessions = new Array()

for ( var i = 0; i < numberOfSessions; i++ ) {

sessionIDs.push(sessions[i].\_id)

keyedSessions[sessions[i].\_id] = sessions[i]

}

var metrics = db.getCollection('metrics').find({ session: { $in: sessionIDs }}).toArray()

var numberOfMetrics = metrics.length

print ( numberOfMetrics )

for ( var j = 0; j < numberOfMetrics; j++ ) {

tempArray = metrics[j]

tempArray['session\_start\_time'] = keyedSessions[metrics[j].session].start\_time

tempArray['description'] = keyedSessions[metrics[j].session].description

tempArray['equipment'] = keyedSessions[metrics[j].session].equipment[1]

tempArray['exercise'] = keyedSessions[metrics[j].session].scene[1]

metrics[j] = tempArray

}

for ( var j = 0; j < numberOfMetrics; j++ ) {

print( metrics[j].session + "," + metrics[j].session\_start\_time + "," + metrics[j].equipment + "," + metrics[j].exercise + "," + metrics[j].name + "," + metrics[j].time + "," + metrics[j].value + "," + metrics[j].unit )

}

If it helps getting an answer quicker, the only metrics that are in this data set are:

Application Extension Handler.Score.Inputs.Current Score

ContentExtensionScene.CONS\_GOAL\_MET\_CAT.Inputs.CONS\_GOAL\_MET\_TIME

ContentExtensionScene.CONS\_GOAL\_MET\_CAT.Inputs.CONS\_GOAL\_MET\_TOTALDIG

ContentExtensionScene.CONS\_PERF\_MET\_CAT.Inputs.CONS\_PERF\_MET\_BUCKET\_CAP

ContentExtensionScene.CONS\_PERF\_MET\_CAT.Inputs.CONS\_PERF\_MET\_BUCKET\_CAP\_AV

ContentExtensionScene.CONS\_PERF\_MET\_CAT.Inputs.CONS\_PERF\_MET\_BUCKET\_EFFITONHR

ContentExtensionScene.CONS\_PERF\_MET\_CAT.Inputs.CONS\_PERF\_MET\_BUCKET\_EFFITONL

Environment.CONS\_ENV\_MET\_CAT.Inputs.CONS\_ENV\_MET\_WEA\_WIND\_SPEED

Excavator.CONS\_ENV\_MET\_CAT.Inputs.CONS\_ENV\_MET\_COLLISIONS

Excavator.CONS\_MACH\_MET\_CAT.Inputs.CONS\_GOAL\_MET\_BUCKETA

Excavator.CONS\_MACH\_MET\_CAT.Inputs.CONS\_GOAL\_MET\_BUCKETH

Excavator.CONS\_MACH\_MET\_CAT.Inputs.CONS\_MACH\_MET\_BUCKET\_SELF

Excavator.CONS\_MACH\_MET\_CAT.Inputs.CONS\_MACH\_MET\_ENGINEAV

Excavator.CONS\_MACH\_MET\_CAT.Inputs.CONS\_MACH\_MET\_ENG\_POWCUR

Excavator.CONS\_MACH\_MET\_CAT.Inputs.CONS\_MACH\_MET\_ENG\_TORQUE

Excavator.CONS\_MACH\_MET\_CAT.Inputs.CONS\_MACH\_MET\_ENG\_TORQUEAV

Excavator.CONS\_MACH\_MET\_CAT.Inputs.CONS\_MACH\_MET\_FUELCONRATEAV

Excavator.CONS\_MACH\_MET\_CAT.Inputs.CONS\_MACH\_MET\_FUELCONSUMED

Excavator.CONS\_MACH\_MET\_CAT.Inputs.CONS\_MACH\_MET\_IDLE\_COUNT

Excavator.CONS\_MACH\_MET\_CAT.Inputs.CONS\_MACH\_MET\_IDLE\_TIME

Excavator.CONS\_MACH\_MET\_CAT.Inputs.CONS\_MACH\_MET\_OPERATING\_RATIO

Excavator.CONS\_MACH\_MET\_CAT.Inputs.CONS\_MACH\_MET\_REVOLUTION

Excavator.CONS\_MACH\_MET\_CAT.Inputs.CONS\_MACH\_MET\_REVOLUTION\_AV

Excavator.CONS\_MACH\_MET\_CAT.Inputs.CONS\_MACH\_MET\_TRACKS\_GRND\_PRESR\_FL

Excavator.CONS\_MACH\_MET\_CAT.Inputs.CONS\_MACH\_MET\_TRACKS\_GRND\_PRESR\_FR

Excavator.CONS\_MACH\_MET\_CAT.Inputs.CONS\_MACH\_MET\_TRACKS\_GRND\_PRESR\_RL

Excavator.CONS\_MACH\_MET\_CAT.Inputs.CONS\_MACH\_MET\_TRACKS\_GRND\_PRESR\_RR

Excavator.CONS\_SAFEVIO\_ENV\_CAT.Inputs.CONS\_SAFEVIO\_BUCKETOVERCAB

Excavator.CONS\_SAFEVIO\_ENV\_CAT.Inputs.CONS\_SAFEVIO\_DUMPTRUCK

Excavator.CONS\_SAFEVIO\_ENV\_CAT.Inputs.CONS\_SAFEVIO\_ELECTRIC

Excavator.CONS\_SAFEVIO\_ENV\_CAT.Inputs.CONS\_SAFEVIO\_HUMAN

Excavator.CONS\_SAFEVIO\_ENV\_CAT.Inputs.CONS\_SAFEVIO\_LOADOVERHUMAN

Excavator.CONS\_SAFEVIO\_ENV\_CAT.Inputs.CONS\_SAFEVIO\_SAFEPARKING

Excavator.CONS\_SAFEVIO\_ENV\_CAT.Inputs.CONS\_SAFEVIO\_VEHICLEFLIP

ContentExtensionScene.CONS\_ENV\_MET\_CAT.Inputs.CONS\_ENV\_MET\_SHOCK\_LOAD

ContentExtensionScene.CONS\_ENV\_MET\_CAT.Inputs.CONS\_ENV\_MET\_SHOCK\_LOADMAX

ContentExtensionScene.CONS\_ENV\_MET\_CAT.Inputs.CONS\_ENV\_MET\_STLP\_FALL

ContentExtensionScene.CONS\_GOAL\_MET\_CAT.Inputs.CONS\_GOAL\_MET\_BUCKET\_HEIGHT

ContentExtensionScene.CONS\_GOAL\_MET\_CAT.Inputs.CONS\_GOAL\_MET\_CURRENT\_MASS

ContentExtensionScene.CONS\_GOAL\_MET\_CAT.Inputs.CONS\_GOAL\_MET\_FORK\_HEIGHT

ContentExtensionScene.CONS\_GOAL\_MET\_CAT.Inputs.CONS\_GOAL\_MET\_GOALS

ContentExtensionScene.CONS\_GOAL\_MET\_CAT.Inputs.CONS\_GOAL\_MET\_TRUSS\_HEIGHT

ImpactMetrics.CONS\_ENV\_MET\_CAT.Inputs.CONS\_ENV\_MET\_CRIT\_IMPACTS

ImpactMetrics.CONS\_ENV\_MET\_CAT.Inputs.CONS\_ENV\_MET\_MAJOR\_IMPACTS

ImpactMetrics.CONS\_ENV\_MET\_CAT.Inputs.CONS\_ENV\_MET\_MINOR\_IMPACTS

Wheel Loader.Machine.Inputs.CONS\_GOAL\_MET\_BUCKETH

Wheel Loader.Machine.Inputs.CONS\_GOAL\_MET\_FORKH

Wheel Loader.Machine.Inputs.CONS\_GOAL\_MET\_TRUSSBOOMH

Wheel Loader.Machine.Inputs.CONS\_MACH\_MET\_AXLE\_LOAD\_FRONT

Wheel Loader.Machine.Inputs.CONS\_MACH\_MET\_AXLE\_LOAD\_FRONTAV

Wheel Loader.Machine.Inputs.CONS\_MACH\_MET\_AXLE\_LOAD\_REAR

Wheel Loader.Machine.Inputs.CONS\_MACH\_MET\_AXLE\_LOAD\_REARAV

Wheel Loader.Machine.Inputs.CONS\_MACH\_MET\_BRAKEPOSAV

Wheel Loader.Machine.Inputs.CONS\_MACH\_MET\_BRAKE\_PED

Wheel Loader.Machine.Inputs.CONS\_MACH\_MET\_ENGINEAV

Wheel Loader.Machine.Inputs.CONS\_MACH\_MET\_ENG\_POW

Wheel Loader.Machine.Inputs.CONS\_MACH\_MET\_FUELCONRATE

Wheel Loader.Machine.Inputs.CONS\_MACH\_MET\_FUELCONRATEAV

Wheel Loader.Machine.Inputs.CONS\_MACH\_MET\_FUELCONSUMED

Wheel Loader.Machine.Inputs.CONS\_MACH\_MET\_GEAR1

Wheel Loader.Machine.Inputs.CONS\_MACH\_MET\_GEAR2

Wheel Loader.Machine.Inputs.CONS\_MACH\_MET\_GEAR3

Wheel Loader.Machine.Inputs.CONS\_MACH\_MET\_GEARD

Wheel Loader.Machine.Inputs.CONS\_MACH\_MET\_IDLE\_COUNT

Wheel Loader.Machine.Inputs.CONS\_MACH\_MET\_IDLE\_TIME

Wheel Loader.Machine.Inputs.CONS\_MACH\_MET\_THROTTLEPOSAV

Wheel Loader.Machine.Inputs.CONS\_MACH\_MET\_THROT\_PED

Version of script which filters the unique metrics present above

var user = db.getCollection('users').find({ name: { $eq: "student4" }}).limit(1).toArray()[0]

var sessions = db.getCollection('sessions').find({ users\_id: { $eq: user.\_id } } ).toArray()

var numberOfSessions = sessions.length

print( numberOfSessions )

var sessionIDs = new Array()

var keyedSessions = new Array()

for ( var i = 0; i < numberOfSessions; i++ ) {

sessionIDs.push(sessions[i].\_id)

keyedSessions[sessions[i].\_id] = sessions[i]

}

var metrics = db.getCollection('metrics').find({ session: { $in: sessionIDs }}).toArray()

var numberOfMetrics = metrics.length

print ( numberOfMetrics )

for ( var j = 0; j < numberOfMetrics; j++ ) {

tempArray = metrics[j]

tempArray['session\_start\_time'] = keyedSessions[metrics[j].session].start\_time

tempArray['description'] = keyedSessions[metrics[j].session].description

tempArray['equipment'] = keyedSessions[metrics[j].session].equipment[1]

tempArray['exercise'] = keyedSessions[metrics[j].session].scene[1]

metrics[j] = tempArray

}

for ( var j = 0; j < numberOfMetrics; j++ ) {

print( metrics[j].session + "," + metrics[j].session\_start\_time + "," + metrics[j].equipment + "," + metrics[j].exercise + "," + metrics[j].name + "," + metrics[j].time + "," + metrics[j].value + "," + metrics[j].unit )

}

var metrickeys = new Array()

for ( var j = 0; j < numberOfMetrics; j++ ) {

if ( !metrickeys.includes( metrics[j].name ) ) {

metrickeys.push( metrics[j].name )

print ( metrics[j].name )

}

}

The following excel sheet includes the comprehensive list of all keys

<https://docs.google.com/spreadsheets/d/1N2LqkuPPtBNuLeUi4mHXD74yb21hDxtUU4ry2jLGd6M/edit#gid=0>

All metrics are updated each frame, average are computed with the current total time [average = (previous average \* (total number of frames -1) + current value) / total number of frames]

Most of the metrics in that list are provided by the simulation but we don't treat all of them to give comprehensive feedback to the user, they exist because they were part of the initial requirements of the PO and could be used for the user feedback. Some are "only" used in order to create real-time widgets (bucket capacity or height for example) and are not relevant in the reports as their value is instantaneous. Some are used in scoring rules (and should give scoring feedback to the user). Some are used to trigger a fail in the exercise (which generates a dedicated event in the HUD but has no effect on the reports) the metric itself should not appear in the report if it has not been triggered, it should generate a different report when the operator fails though.

All metrics appear as is in the dashboard and in the report of the exercise (only the last value recorded in that case) as we have no means of filtering what gets there other than deleting the metric in the exercise which is extremely cumbersome as they are part of an extension in a mechanism (one extension contains many metrics, you can only shut down the whole container). In recent training packs, we have created "metrics mechanisms" for environmental metrics with each metric in a dedicated container with a configuration so we can have a bit of agency in that regard but it is not how the metric container extension is meant to be used apparently.

Ideally dashboard, reports and scoring should be part of content so we can customize them according to the needs of the exercise and prevent the report from being polluted by irrelevant metrics.

---------------------

Looking in the sample database we're working with, I found that the exercise that had the most sessions was an exercise from our Excavator called Arc Swipe with 40 sessions.

<https://u.pcloud.link/publink/show?code=kZEffHXZRbo0YsSGkfLcCQa7izYcG4KNIpv7>

I've extracted data in a similar format as before, but with two extra fields (PassFail and Score). Values for PassFail can be None, Pass or Fail. Score is a 100% score mark calculated by the system. Here is the column order: session\_id, start\_time, equipment name, exercise name, pass/fail, score, metric name, timestep, value, unit

I've also put together more significant description of the metrics and categorized them as input or output: <https://docs.google.com/spreadsheets/d/1YrzYq8R-9vrl0gTZqpUgYTmXpc7VLSwEcYCbT3GnBPQ/edit?usp=sharing>. Of course, the real inputs of the equipment are the joysticks and pedals of the equipment simulator, but their operation directly controls the bucket angle and height, along with the engine power, so I think that it makes sense to consider them as input.