VRisk
Tools Manual

Jade Eames – Simone Farinelli

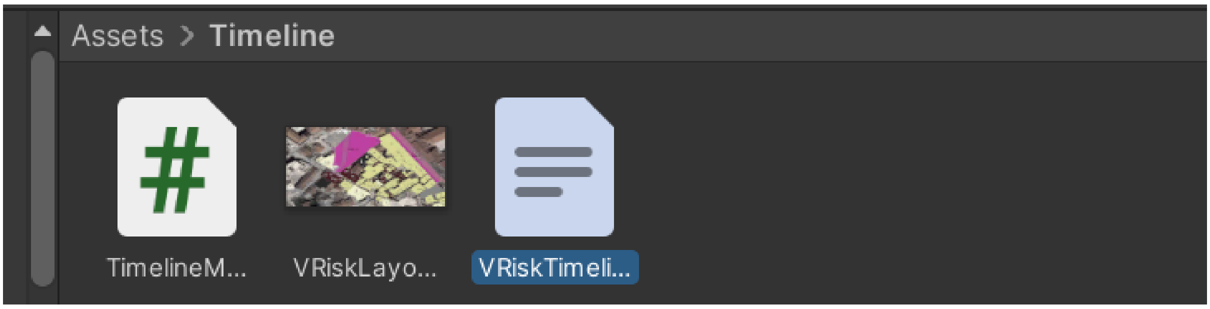
This document will showcase how to operate the different tools that have been created for VRisk.

Timeline CSV

VRisk stores the timeline of events in a .CSV file within the project files. This file can preferably be edited with Excel, but other text editors are also usable.

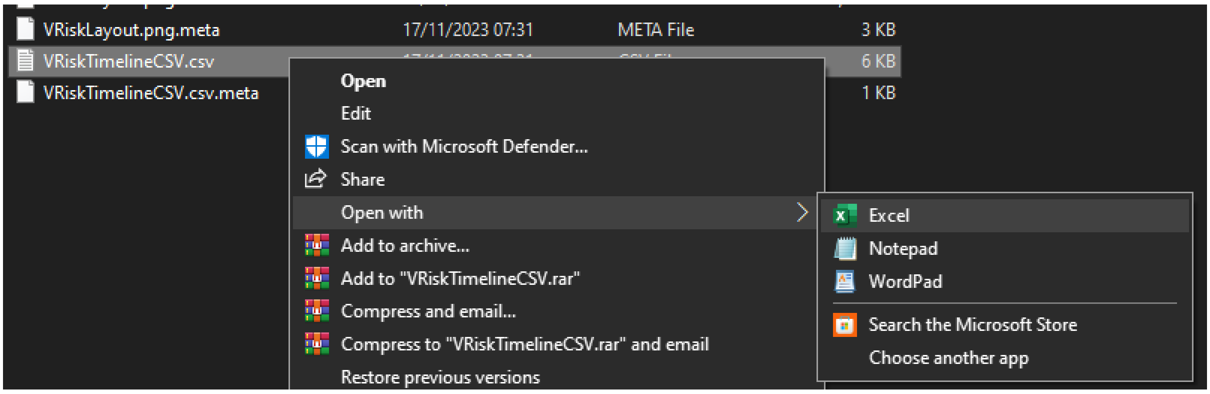
The file needs to be edited within Unity. Upon building, the file will be integrated into the .APK. This means that once the application is uploaded to the Oculus, the .CSV won't be able to be edited without re-building and re-uploading the project again.

The .CSV is stored inside Assets -> Timeline



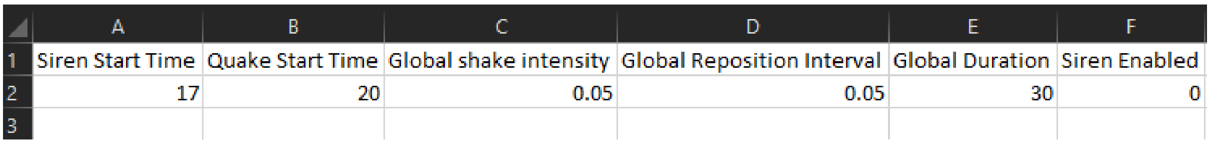
Right-click on it and select "Show in Explorer" to access the Windows folder. Keep in mind that the correct file is the first one, which does not end in .meta.

Windows will, by default, open it with Notepad, but we advise opening it with Excel.


VRiskLayout.png.meta 17/11/2023 07:31 META File 3 KB
VRiskTimelineCSV.csv 6 KB
VRiskTimelineCSV.csv.meta 1 KB
Open
Edit
Scan with Microsoft Defender...
Share
Open with
Add to archive...
Add to "VRiskTimelineCSV.rar"
Compress and email...
Compress to "VRiskTimelineCSV.rar" and email
Restore previous versions
Excel
Notepad
WordPad
Search the Microsoft Store
Choose another app

The document is divided into two sections, the simulation general values and the event timeline.

The simulation's general values are stored in the first two rows of the document:



	A	B	C	D	E	F
1	Siren Start Time	Quake Start Time	Global shake intensity	Global Reposition Interval	Global Duration	Siren Enabled
2	17	20	0.05	0.05	30	0
3						

Each value affects the following:

<u>Siren Start Time</u>:
This value determines how many seconds must pass after the player enters the simulation before the warning siren starts. It can be left as it is, as the warning sirens were an initial feature that was removed; therefore, this value does not affect anything.

<u>Quake Start Time</u>:
This value determines after how many seconds from the player entering the simulation have to pass before the earthquake starts.

<u>Global Shake Intensity</u>:
This value determines the intensity of the shake caused by the earthquake.

The value is in Units, a type of measurement that Unity uses to represent distance, see it as something like meters, changing this value will alter how much room a building has available to shake, therefore resulting in a more intense shaking.

Setting the value too high will result in buildings only moving sporadically, which significantly diminishes the illusion of continuous shaking.

<u>Global Reposition Interval</u>:
This value determines the rate at which shaking buildings will update their position inside the room available to shake.

In VRisk, to represent shaking, a building will rapidly and slightly change its position within the available room for shaking, as determined by the intensity.

Changing Global Reposition Interval alters how often per second a building will relocate itself, thus affecting the perception of how the building shakes.

The default value is set at 0.05 which means that a building will relocate every 50 milliseconds, the smallest value that can be put is 0.016, but doing so may worsen performance, while increasing the value will not have any impact on it.

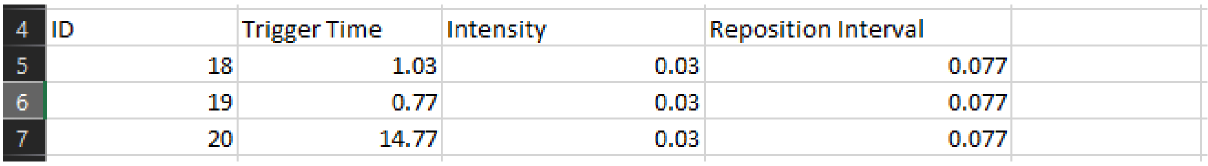
Setting the value too high will result in buildings only moving sporadically, which significantly diminishes the illusion of continuous shaking.

<u>Global Duration</u>:
This value determines how much time the earthquake will last once started, in seconds.

It is very important to notice that if this value were to be changed, <u>the timeline should be re-generated from scratch to fit the new duration</u>, a tool has been built to facilitate this action, it will be covered right after the timeline.

Siren Enabled:
This is a Boolean value that determines if the Warning Sirens will start or not, it can be left as it is, as mentioned before, the warning sirens were an initial feature that got removed.

Meanwhile, the simulation's event timeline is stored below the general values, from row 4 and on:



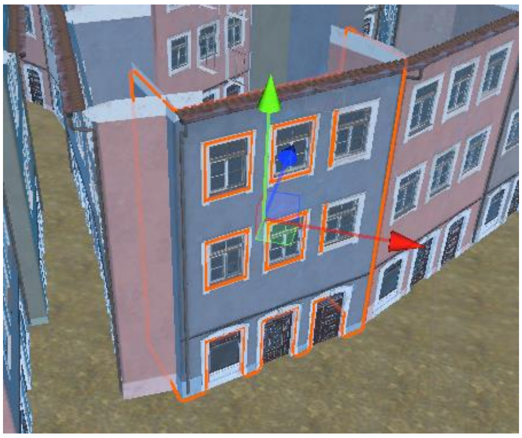
4	ID	Trigger Time	Intensity	Reposition Interval	
5	18	1.03	0.03	0.077	
6	19	0.77	0.03	0.077	
7	20	14.77	0.03	0.077	

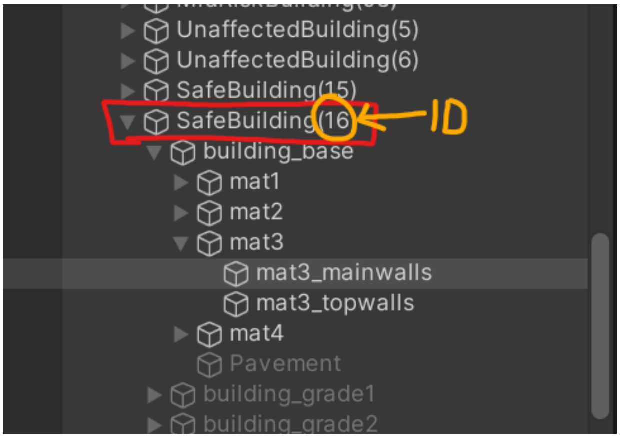
Each row added below row 4 is considered an entry in the timeline and will cause damage to a specific building, each of the values in the entry specify the following:

ID:
This value specifies the ID of the building that will receive damage.

A building's ID can be easily gathered from the Game Scene in Unity doing the following:

- Click on the interested building in the scene.
- Find the highlighted element in the hierarchy.
- Track down the first entry in the hierarchy of that building, it will contain the ID.





Trigger Time:
This value determines after how many seconds from the start of the earthquake the specified building will receive 1 level of damage.

Using the image above as an example, the first entry determines that building 18 will take damage after 1.03 seconds from the beginning of the earthquake.

Intensity & Reposition Interval:
The functionality affected by these values has been deactivated, as it would add a layer of shaking on top of the normal shaking for the buildings nearby a building who took damage, this caused a chain reaction that tanked the framerate on the Oculus, so it had been turned off.

If the functionality were to be re-enabled, these values will affect the following:

- Intensity of the additional shaking given to the nearby buildings
- Reposition Interval of the additional shaking given to the nearby buildings

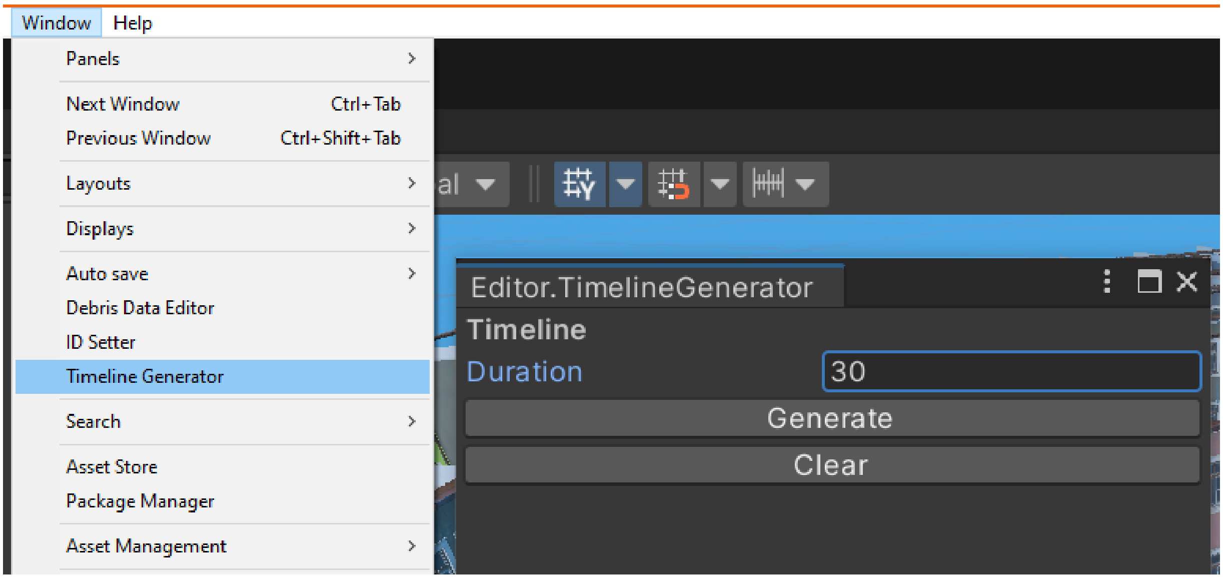
Timeline Generator

The Timeline Generator is a tool we built to quickly and automatically generate a complex timeline. A timeline created via the Timeline Generator will match all of a building's properties, but it will randomly determine when that building is damaged.

To insert a specific trigger time for a building, the timeline will have to be edited manually.

Note: Using the Timeline Generator will cause any existent timeline to be cleared and overwritten.

The tool is under Window -> Timeline Generator. Make sure that the timeline is not open in any other program before running the tool, and ensure that the tool is being opened in the Game Scene.



Duration 30

Ideally, the tools should always be used in the following order: Insert duration -> Clear -> Generate, as each of the options does the following:

Duration:
This value determines how many seconds the generated Timeline will be. This value must be the same as the “Global Duration” specified in the timeline. If these two values differ, the simulation may result in being off-sync.

Generate:
This option will open Timeline.CSV, read all the data from the buildings in Game Scene and then write a new timeline to the file.

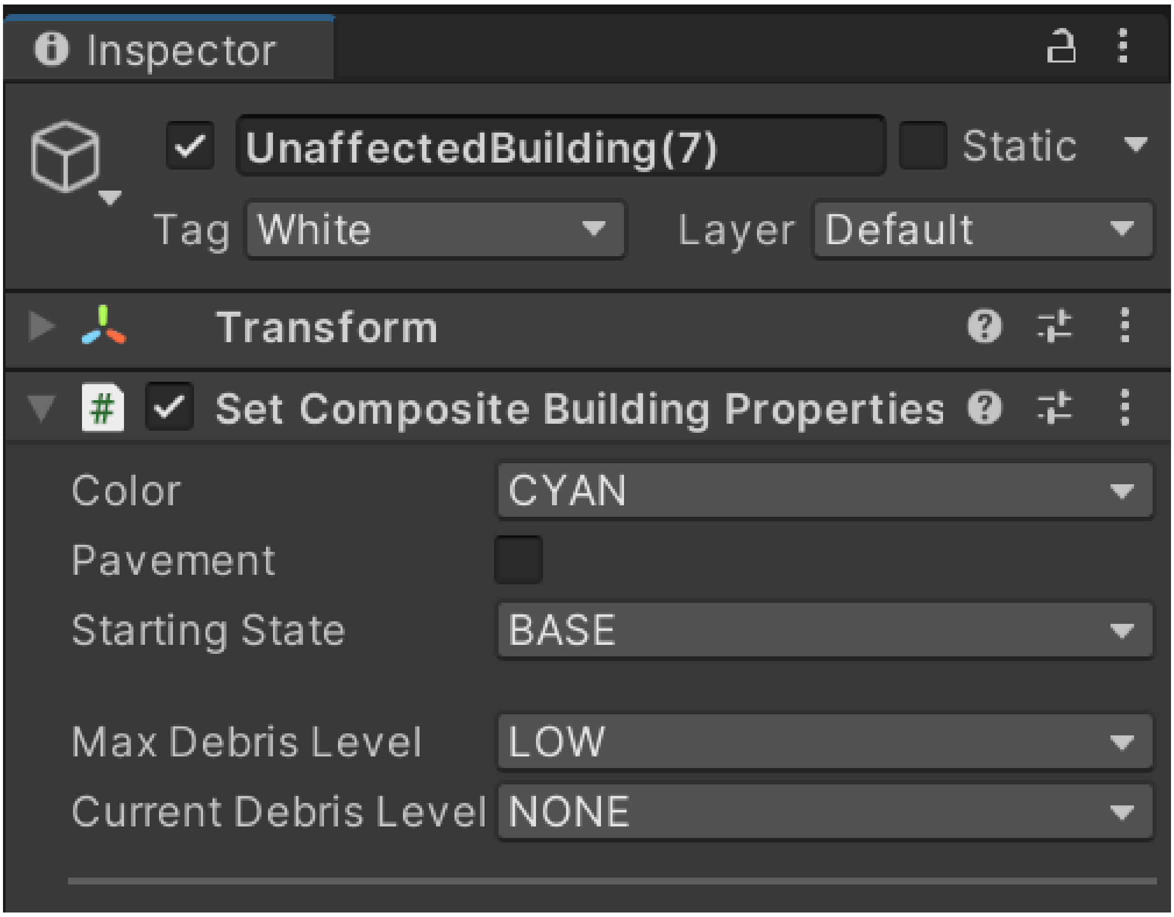
However, this won’t clear any older timeline already present, causing overlapping and the generation of a timeline that is not readable by VRisk.

Clear:
This option will open Timeline.CSV and clear any data present inside, reverting the document at its initial state.

This not only includes the deletion of the timeline but will also the reset of the simulation data (first rows of the csv) too, which will be replaced with the default data.

Building Properties

Each building in VRisk has a set of properties that can be customized, to access them, click on the base level of the building hierarchy in the hierarchy viewer, where the ID is specified, this should show the building properties in the inspector.



☑ UnaffectedBuilding(7) □ Static

Tag White

Layer Default

☑ Set Composite Building Properties

Color CYAN

Pavement □

Starting State BASE

Max Debris Level LOW

Current Debris Level NONE

Each of the customizable values will change the following:

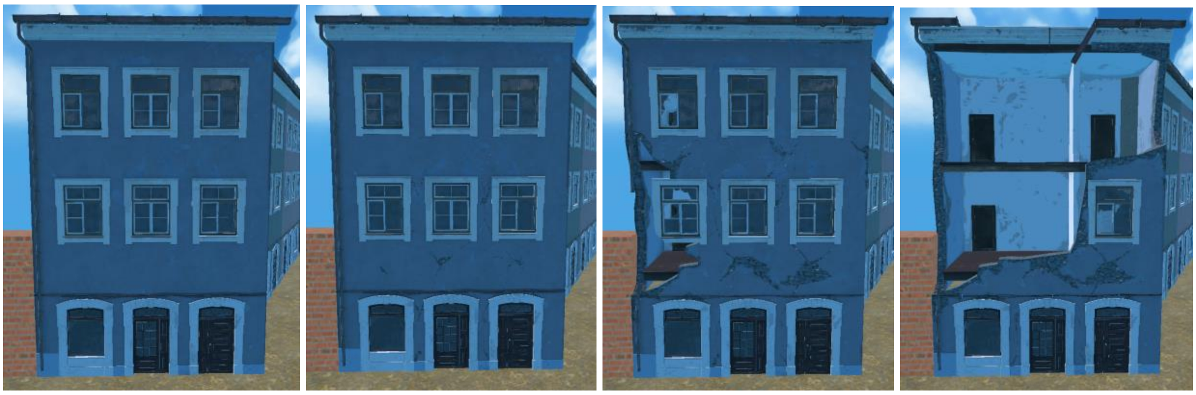
Tag:
Buildings in VRisk can be tagged with their respective danger level, which will apply the specified amount of damage to them throughout the simulation.

To reflect the reference map on which VRisk is based on, the risk tags are based on colours, which are the following:

- “White” or “Black” to tag safe buildings, these buildings won’t receive any damage.
- “Beige” to tag low risk buildings, these buildings will receive a low amount of damage.
- “Yellow” to tag medium risk buildings, these buildings will receive significant damage.
- “Red” to tag high risk buildings, these buildings will collapse.

Internally, a building categorizes damage using grades, so a Grade 1 damage will be applied with the “Beige” tag, Grade 2 with the “yellow” tag and Grade 3 with the ”red” tag, while BASE is the standard building without any damage applied to it, this would correspond to a “White” or “Black” tag.

This is what each grade of damage will look like, in order: BASE, Grade1, Grade2 and Grade3



Note: This value is used by the Timeline Generator and ID setter to create the timeline, so if any of the buildings gets their tag altered, the timeline needs to be re-build and the ID setter needs to be ran again to sort the IDs after the changes.

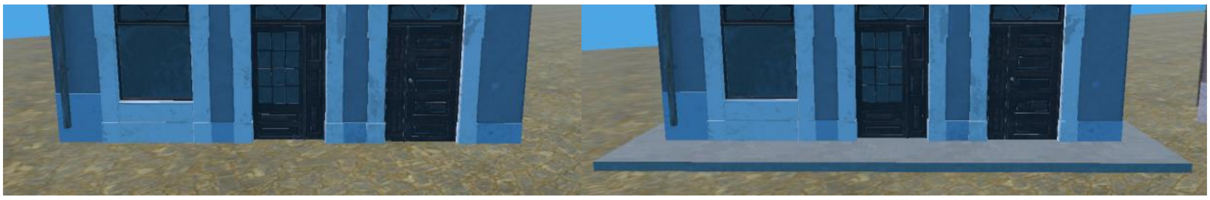
The Timeline Generator is covered above, and the ID setter is covered below.

<u>Colour</u>:
This allows for the colour of a building to be changed from a set of pre-defined ones, being Cyan, Green, Grey, Pink, White and Yellow.

<u>Pavement</u>:
This is a Boolean value which can be used to toggle on and off a building's pavement.

This feature was implemented for the tutorial, as having a pavement helps with the immersions in that scenario, however, pavement is turned off the in the Game Scene as keeping it on would give a claustrophobic feeling in small hallways, thus result in more clutter while shaking.

Example of the pavement toggled off and on.



<u>Starting State</u>:
This value defines the initial condition of a building before the simulation starts.

This value uses Grades to quantify damage, as explained before this is how buildings categorize damage internally, therefore setting this variable to a grade of damage other than BASE will cause the selected building to already be damaged before the simulation starts.

The damage grades are the same showed above in the "tag" section.

Note: Changing this value will cause buildings to behave differently from their tag, this is because every time a building takes damage from the timeline, the damage grade will be increased from the previous one.

For example, if a yellow tagged building has its initial state set to "Grade1", by the end of the quake it would have collapsed, as yellow buildings take 2 grades of damage, therefore ending in "Grade3".

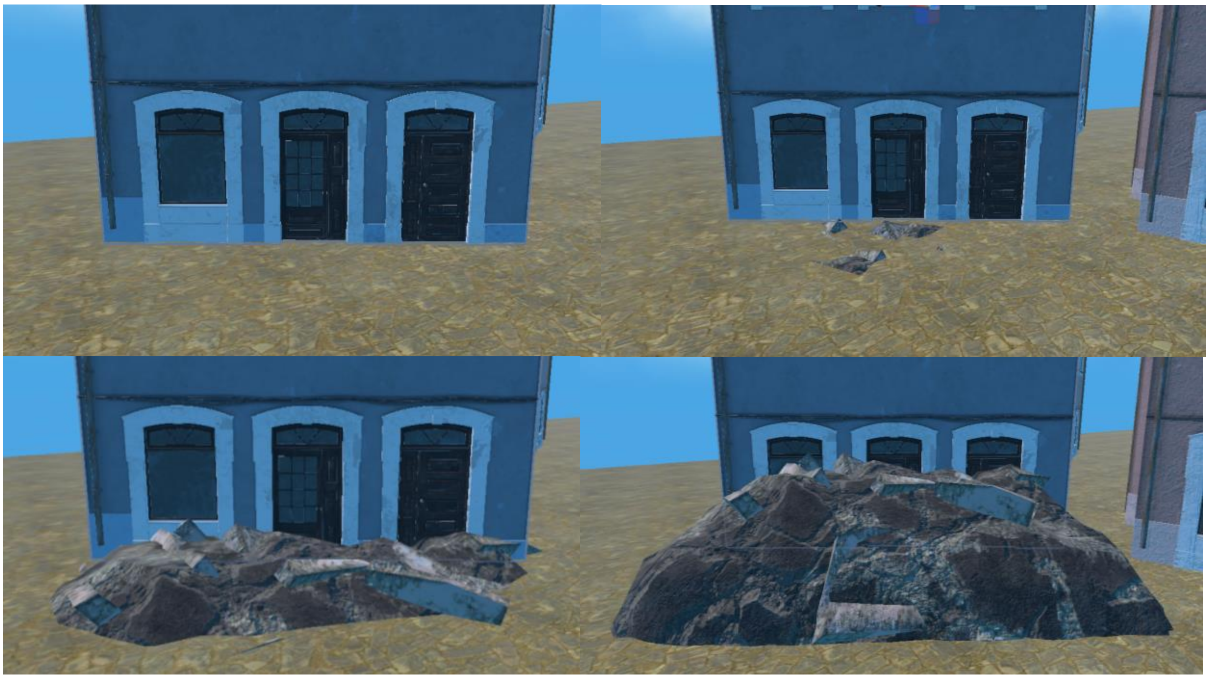
Max Debris Level:
This value limits the maximum amount of debris that can deposit in front of a building.

Buildings will accumulate debris in front of themselves when taking damage, with a respective amount of debris for each grade of damage.

Collapsed (grade3) and very damaged (grade2) buildings will be blocking the path in front of themselves when at their respective debris level.

This value allows to limit the number of debris that gets deposited in front of a building, meaning that, if a "red" tagged building is set with maximum debris level "low", the building will only cause a small amount of debris despite collapsing.

The level of debris available are the following, in order: None, Low, Mid, High.



Current Debris Level:
This value works the same as Starting State but for debris, it defines the initial amount of debris before the simulation starts.

Behaving the same as the Starting State variable, this will also mean that changing this value will alter how debris spawn despite the building grade of damage, therefore if a building current level is set a "Low" before the simulation and its tagged as yellow, it will have max debris level by the end of the quake, because the amount of debris will be increased two times.

Note: This value is affected by Max Debris Level and will not exceed it, this means if the building current amount of debris is low and the Max Debris level is set to low, the debris amount will remain unaltered throughout the simulation, even if the building is set to collapse.

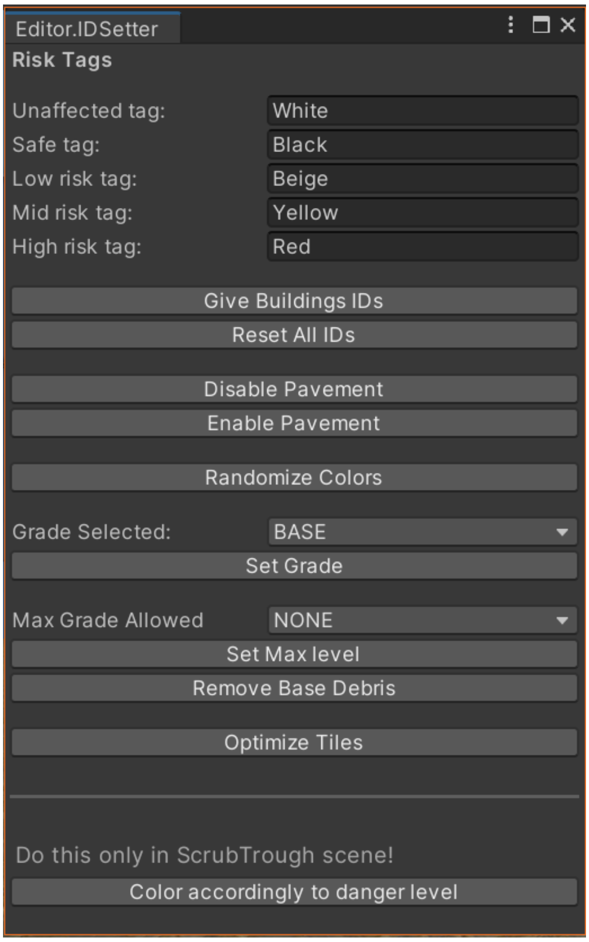
ID Setter

The ID setter is a tool that we made to quickly manage buildings inside the Game Scene, it allows grouping and property editing on all buildings easily from one point.

Note: Using the ID setter will cause building-specific properties to be overridden.

This means that if the Starting State of the buildings is changed via the ID setter, all the buildings will be set to that new value overriding the old one, however, building properties can still be set for each individual building **after** using the ID Setter.

To access it, first make sure the Game Scene is the scene that is currently open, then it can be found inside Window -> ID Setter, it should look like the following:



Unaffected tag: White
Safe tag: Black
Low risk tag: Beige
Mid risk tag: Yellow
High risk tag: Red

Grade Selected: BASE

Max Grade Allowed NONE

Do this only in ScrubTrough scene!

Risk Tags:
This section can be populated with the risk tags and their respective danger level.

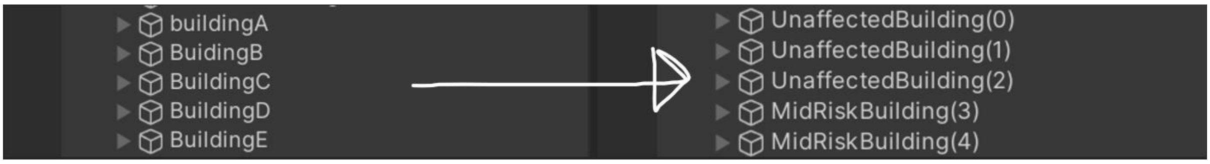
As covered in the building properties, risk tags can be assigned to buildings to specify their danger level, however, the ID Setter needs to be informed of which tags correspond to what level of risk, this can be done through this section by inserting the tag inside its specific text box.

"Unaffected" and "Safe" tags will be both categorized as safe buildings, as for optimization we removed the "Unaffected" tag as it had only few changes from the "safe" tag.

Note: We strongly recommend to not change these settings, as doing so will require all the buildings to be re-tagged one by one.

<u>Give Building IDs</u>:

This option will label buildings with their respective danger level and assign them an ID.



buildingA
BuidingB
BuildingC
BuildingD
BuildingE

UnaffectedBuilding(0)
UnaffectedBuilding(1)
UnaffectedBuilding(2)
MidRiskBuilding(3)
MidRiskBuilding(4)

As mentioned previously in the document, this is the option to be used when changes to the building's tags have been made, or if new buildings are added/removed.

We advise to use "Reset All IDs" before giving the buildings IDs, this will avoid any conflict.

Note: Regenerating a timeline will be required after modifying the IDs.

<u>Reset All IDs</u>:

This option will reset the internal ID of each building to 0.

<u>Disable Pavement & Enable Pavement</u>:

Respectively allows to enable and disable the pavement for all the buildings inside the Game Scene, will override previous settings.

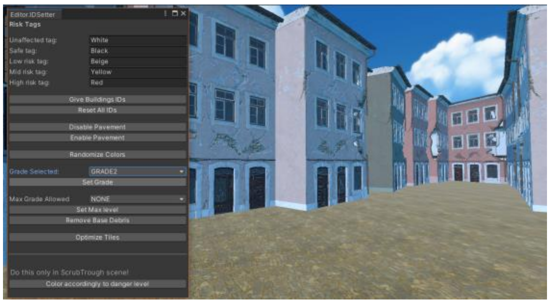
<u>Randomize Colours</u>:

This option will assign each building in the Game Scene a random colour from the ones available, will override previous settings.

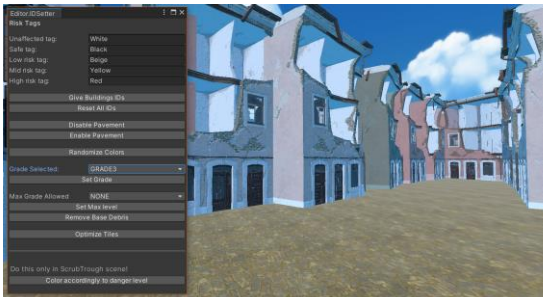
Each building's colour can be changed independently from its own building properties script after running the command.

<u>Set Grade</u>:

This option will set the "Starting State" (Grade) of each building with the one selected in "Grade Selected", will override previous settings.



Editor.IDSetter
Risk Tags
Unaffected tag: White
Safe tag: Black
Low risk tag: Beige
Mid risk tag: Yellow
High risk tag: Red
Grade Selected: GRADE2
Max Grade Allowed: NONE
Do this only in ScrubTrough scene!



Editor.IDSetter
Risk Tags
Unaffected tag: White
Safe tag: Black
Low risk tag: Beige
Mid risk tag: Yellow
High risk tag: Red
Grade Selected: GRADE3
Max Grade Allowed: NONE
Do this only in ScrubTrough scene!

Using this function will change a building’s “Starting State”, the behaviour of this value and its complications are covered in “Buildings Properties”.

<u>Set Max Level</u>:
This option will set “Max Debris Level” of each building with the one selected in “Max Grade Allowed”, will override previous settings.

Using this function will change a building’s “Max Debris Level”, the behaviour of this value and its complications are covered in “Buildings Properties”.

<u>Remove Base Debris</u>:
This option will set the “Current Debris Level” of each building to “NONE”, will override previous settings.

This function doesn’t really have a purpose now that the project is complete, It was mostly used as a developing tool, since importing and exporting buildings would cause debris to spawn, thus we made this function to quickly get rid of them.

<u>Optimize Tiles</u>:
This option will optimize the buildings model to perform better on the Oculus, this is another developer tool which we used to make VRisk.

When importing a building into the scene, its model will have a very detailed rooftop with plenty of tiles, this however can have a huge impact on performance for mobile devices like the Oculus.

This function will get rid of all the excessive tiles that are not visible from the player’s POV, resulting in less draw calls per frame, increasing performance.

Using the function on an already optimized building won’t do anything.

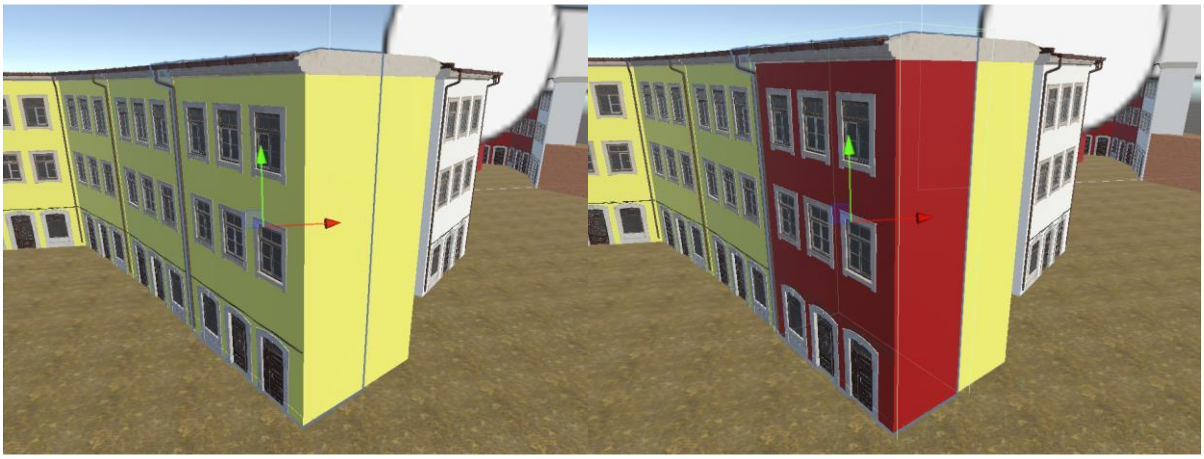
<u>Colour Accordingly to Danger Level</u>:
This option will colour the buildings of the Scrub Trough scene accordingly to their danger level.

Note: This function must <u>only</u> be used inside the Scrub Trough scene.

The risk tags of the Game Scene and Scrub Trough are not linked together; however, it is possible to change them in the Scrub Trough scene too, which can be used to match the ones in the Game Scene.

Doing so will allow this function to re-colour the Scrub Trough map accordingly to the danger level of each building, making it clearer while using the replay feature.

In this example the risk tag of building 32 has been changed to “red” from “yellow”:



Debris Data Editor

The Debris Data Editor is a tool that we made to manage the debris system, it allows to set quantity, force and behaviour of the debris in the Game Scene.

Note: We advise against using this system, since we are using composite meshes it can break easily. However, if the system needs a change to improve the user experience, feel free to continue reading this section and experiment inside Unity, we advise to make a backup before doing so.

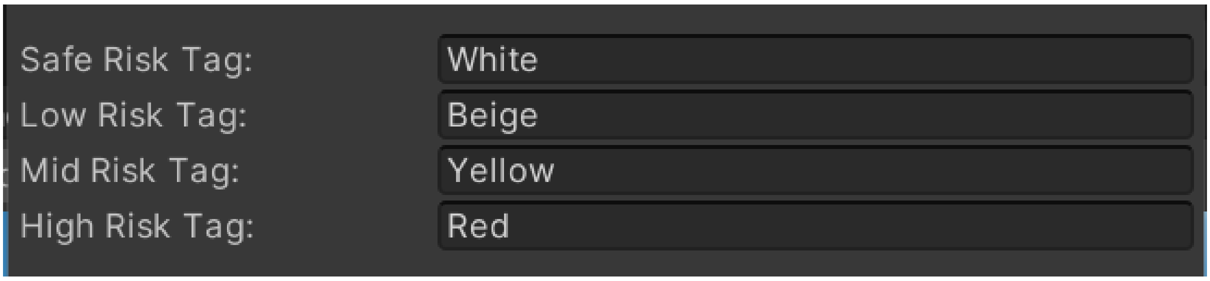
It is important to keep in mind that debris have a timeline too, however, is internal and cannot be modified outside the tool.

Note: To apply any change made it will be necessary regenerate the all debris data, this will override any previous data, how is covered below.

To access the tool, first make sure the Game Scene is the scene that is currently open, then it can be found inside Window -> Debris Data Editor, on laptops and low-resolution screen it could happen that the tool doesn’t fit properly on the screen.

Debris Data Editor: Settings

Risk Tags:



Safe Risk Tag: White

Low Risk Tag: Beige

Mid Risk Tag: Yellow

High Risk Tag: Red

This section can be populated with the risk tags and their respective danger level.

As covered in the past two tools, risk tags can be assigned to buildings to specify their danger level, the debris data editor needs it too, to understand which objects in the scene are buildings and what appropriate level of debris they should produce.

This time however, there is no “Unaffected” tag, this is because this system was implemented after we decided to give “Unaffected” and “Safe” the same behaviour.

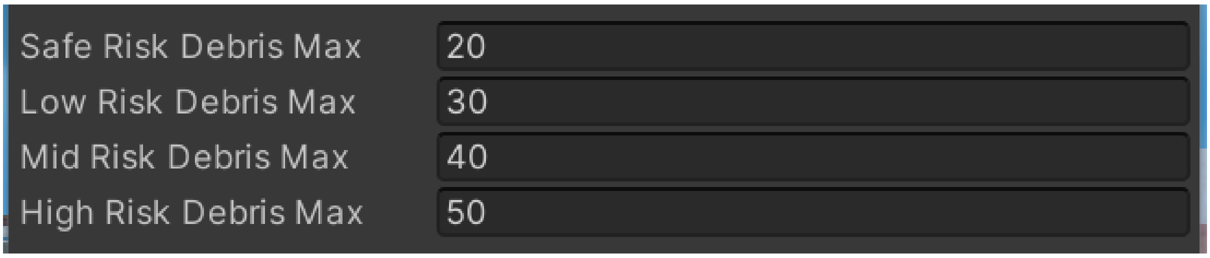
Note: Just like the in the ID Setter, we strongly recommend to not change these settings, as doing so will require all the buildings to be re-tagged one by one, plus will require the ID Setter to be updated with the new tags too.

<u>Debris Count Ranges</u>:

This section allows to set minimum and maximum quantities for each risk category of debris, this option is needed, as to prevent lag we use pooling.

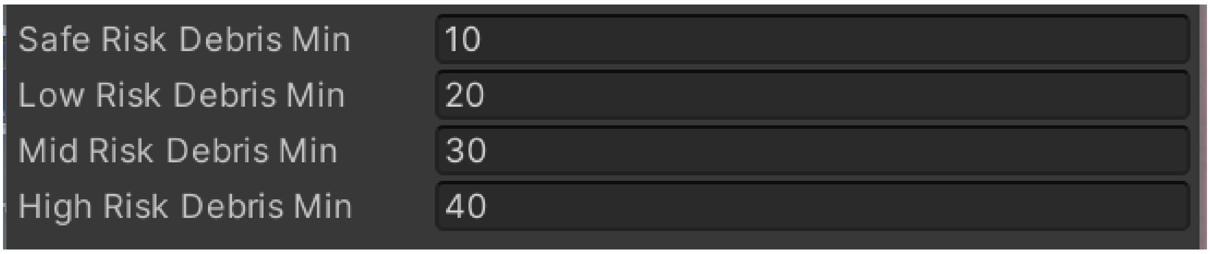
Pooling is creating and instancing the debris off-camera and moving them to the scene where they are needed, then move them back off-screen once they lifespan is over, this allows to use few debris object while giving the feeling of all the buildings dropping many debris.

The following settings will set the range of the pooling of objects.



Safe Risk Debris Max	20
Low Risk Debris Max	30
Mid Risk Debris Max	40
High Risk Debris Max	50

The “Max” settings will allow to set the maximum number of debris to be pooled for each building category, usually we want safer category to have less debris as they will be less likely to produce drop any.

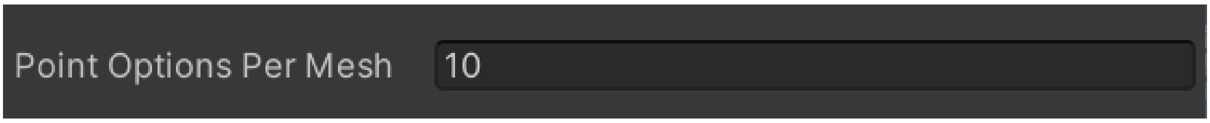


Safe Risk Debris Min	10
Low Risk Debris Min	20
Mid Risk Debris Min	30
High Risk Debris Min	40

The “Min” settings will allow to set the minimum number of debris to be pooled for each building category, this should not be too low as otherwise it will require too many debris to be pooled at runtime.
Just like before, the riskier the category the more debris should be pooled.

Note: A very high number of pooled debris could cause performance drop on the Oculus.

<u>Point Options Per Mesh</u>:



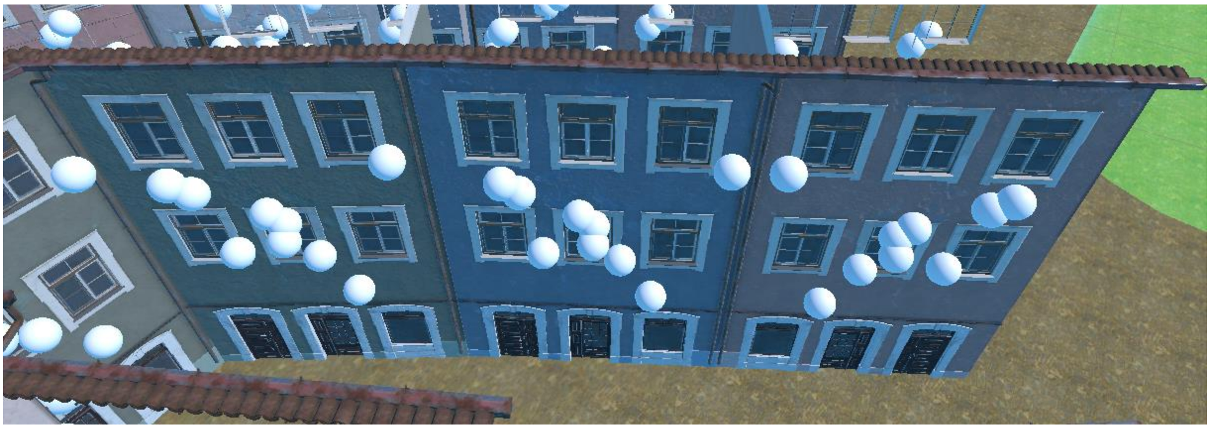
Point Options Per Mesh	10

This option allows to define how many spawn points will be placed on a building for debris to spawn.

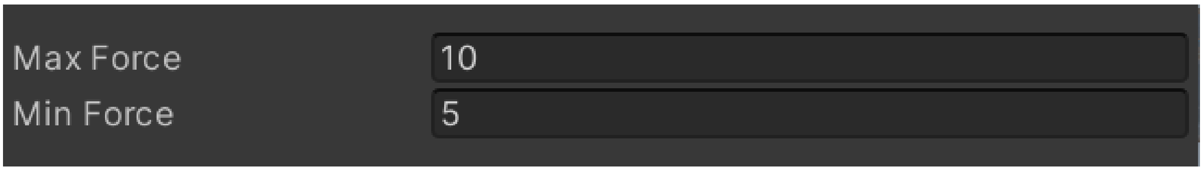
The debris data editor will map on top of a building a series of point from which debris will spawn from, those points will be different for each grade of damage of a building.

Changing this option allows for the amount of spawn points to be altered, usually we advise to keep a number under 21 for performance reasons, the number will be the value inserted -1.

The following picture shows the spawn points generated with the input of 10, their position will change every time they are regenerated.



<u>Debris Force</u>:



Max Force 10

Min Force 5

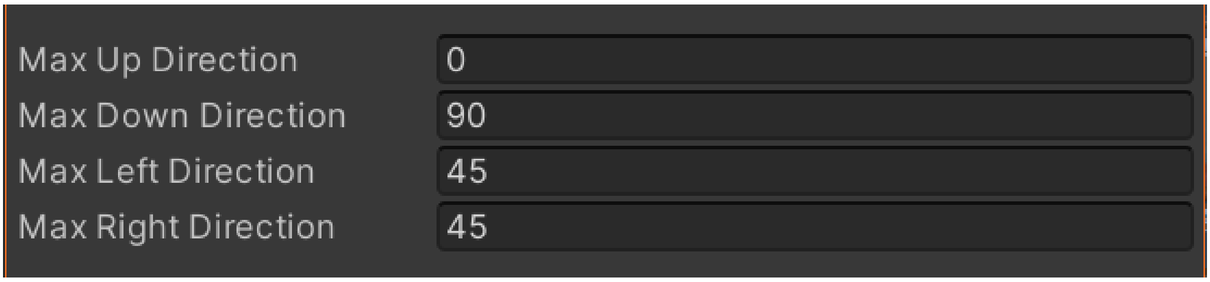
These options define the minimum and maximum physical forces applied to debris upon generation.

Debris will be spawning with a physical force applied to them, to make them look like more realistic, mimicking the force cause by the collapsing of a building, for example, this force will be a random number between the Min force and Max force, it will be saved inside the debris timeline to make it deterministic.

The unit of the value is in Unity "force", we like to see it as a speed applied to a specific direction, a value too low will cause the debris to simply fall and a value to high will cause them to shoot away at a high speed.

The direction of which the force will be applied upon spawning is covered below.

<u>Debris Direction</u>:



Max Up Direction 0

Max Down Direction 90

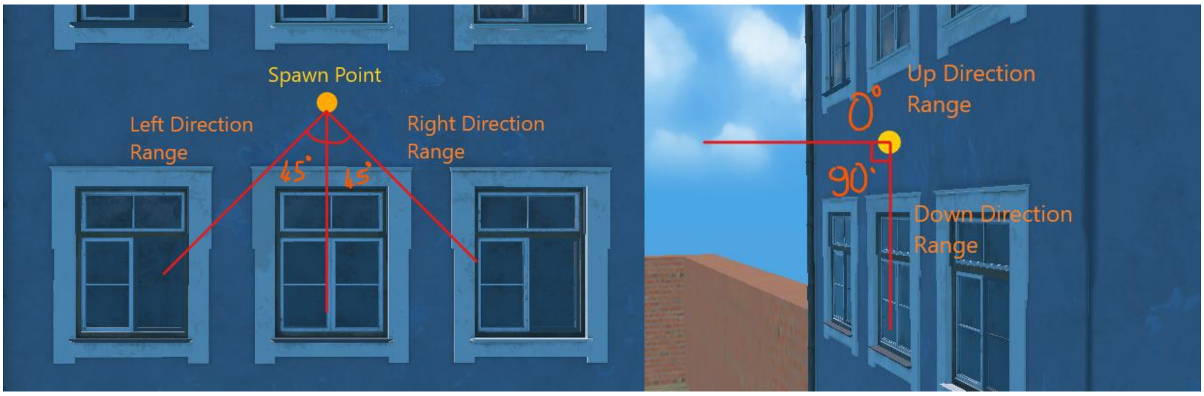
Max Left Direction 45

Max Right Direction 45

These options specify the range in degrees for the upward, downward, leftward, and rightward dispersal of debris pieces.

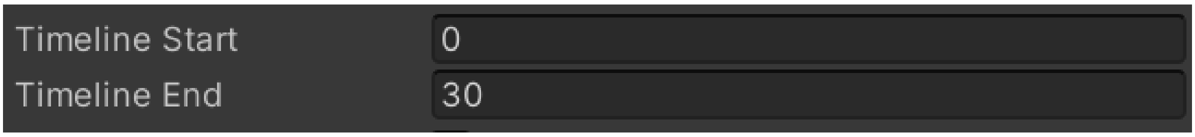
The angles inserted act as a range of degrees in each direction from the front of a building to apply a force to the debris that will be generated, the final angle will involve all 4 and will be saved inside the debris data to make it deterministic.

Visualizing this will explain it way better than words, below it is shown how the pre-defined values will act as the range of possible direction at which the force will be applied:



Note: A “Max Up Direction” too high could cause debris to be shot in the air vertically.

Timeline Start & Timeline End:



Timeline Start 0

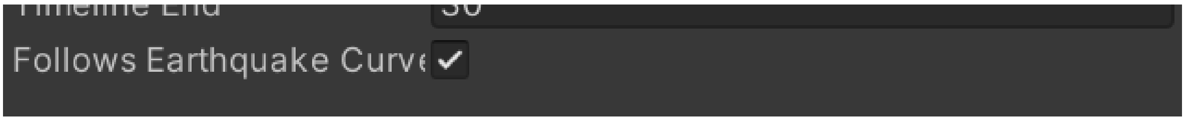
Timeline End 30

These options allow to set the start and end points of the debris generation timeline, in seconds.

However, it is important to notice that “Timeline Start” will start counting from when the earthquake begins, so if is set to 2 debris will start falling after two seconds from the start of the earthquake, we advise to keep it at 0 so debris start alongside the quake.

Meanwhile, “Timeline End” marks the end of a high quantity of debris falling, after 30 seconds from the start have passed debris will still fall but at a reduced rate.

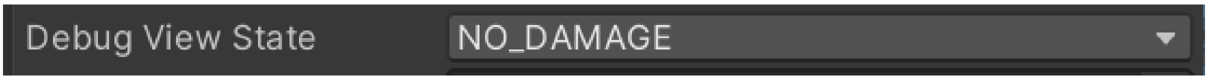
Follows Earthquake Curve:



Follows Earthquake Curv ✓

This option toggles whether the debris generation follows an earthquake intensity curve of the simulation for a more accurate rate of debris spawning.

Debug View State:

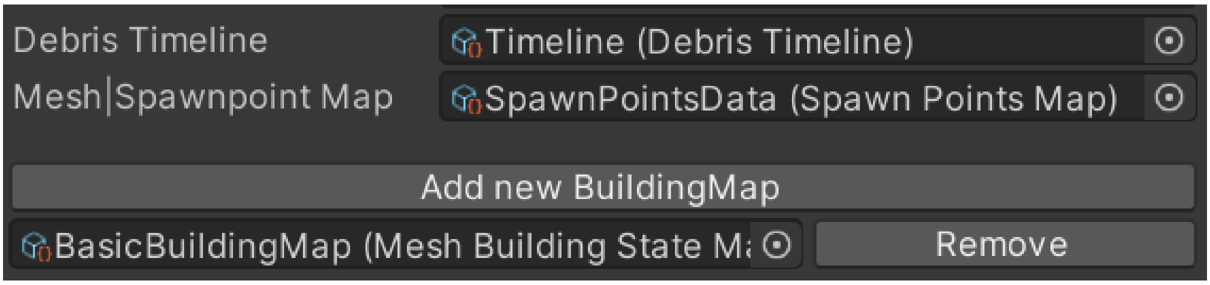


Debug View State NO_DAMAGE

This option allows to select the grade of damage of the mesh points generated that will be visualized via the debug options.

Once the option to view the debug visuals is selected, the result will be the same showed in “Point Options Per Mesh”, the “Generate Debug Visuals” option is covered below.

Debris Timeline & Spawn Maps:



Debris Timeline: Timeline (Debris Timeline)

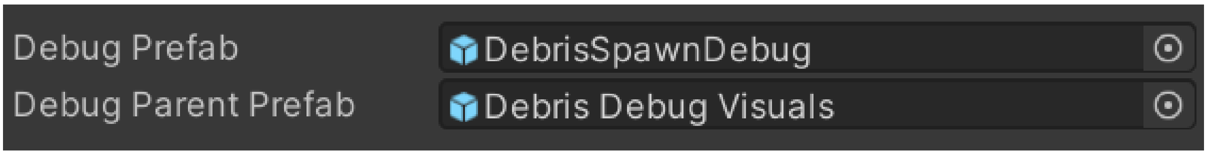
Mesh|Spawnpoint Map: SpawnPointsData (Spawn Points Map)

BasicBuildingMap (Mesh Building State Ma

Visual links to the respective timeline and spawn point map data structures, these are used internally by VRisk to handle debris data.

Note: Changing any of those values will cause the debris system to break.

Debug Prefabs:



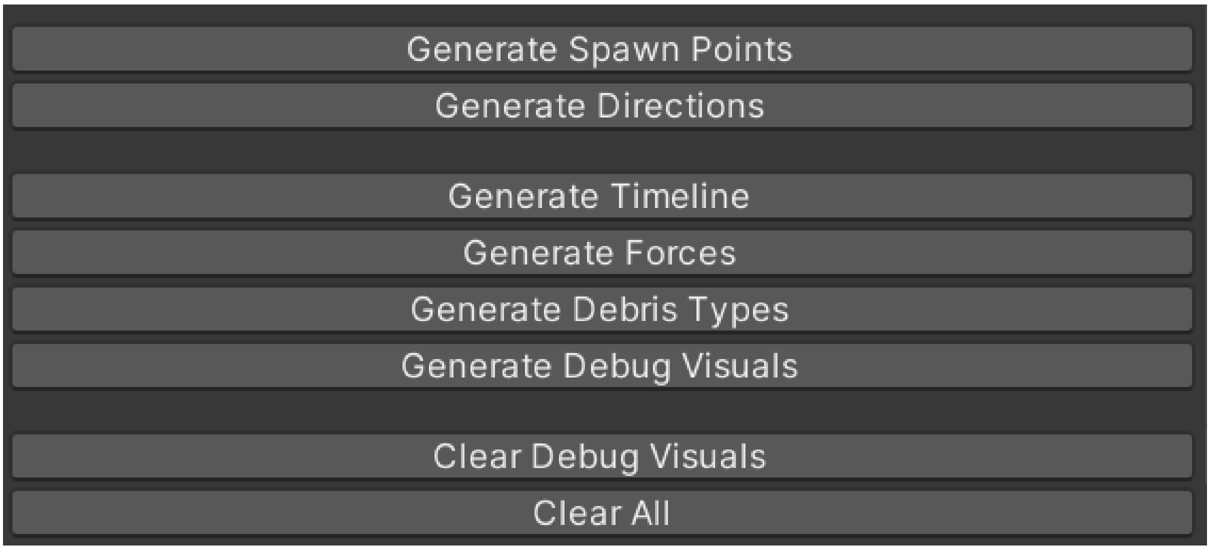
Debug Prefab: DebrisSpawnDebug

Debug Parent Prefab: Debris Debug Visuals

Assign prefabs for visual debugging purposes, those are the models that will be shown when debug visuals are turned on.

Note: Changing any of those values will cause the debris system to break.

Debris Data Editor: Functions



Each of those options will directly update its corrispondente data in the files

Generate Spawn Points:
This option will initiate the computation of spawn points for debris based on the provided mesh and building state mappings.

Generate Directions:
This option will calculate the direction vector for each debris instance with the data provided.

Generate Timeline:
This option will compile the sequence of debris generation events over the provided timeline.

Note: The timeline is internal and cannot be edited manually.

Generate Forces:
This option will calculate force intensity for each direction vector previously calculated.

Generate Debris Types:
This option will assign a random debris type to each entry in the timeline.

Generate Debug Visuals:
This option will read the value selected in "Debug View State" (covered above) and generate the respective debug visuals onto the Game Scene's buildings to allow to see where they are located.

The result will be the image attached below "Debug View State".

Note: The existing visuals should always be cleared before using this option again.

Clear Debug Visuals:
This option will remove all the debug visuals present.

Clear All:
This option will run an internal script that will delete any previously created debris data from the project.

Note: This option should always be run before regenerating the debris data to avoid overriding.

Execution order:
To avoid overriding and breaking the debris system, all these options should almost always be ran in the following order:

- To regenerate the debris data: Clear All -> Generate Spawn Points -> Generate Directions -> Generate Timeline -> Generate Forces -> Generate Debris Types

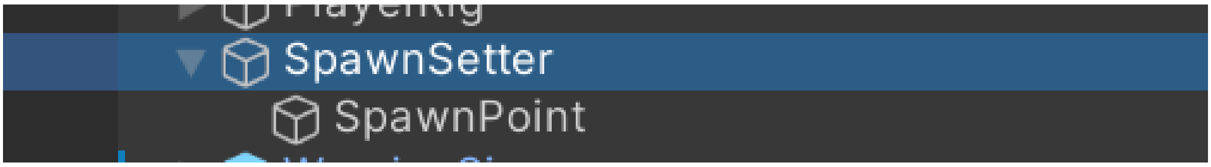
- To show debug visuals: Clear Debug Visuals -> Debug View State -> Generate Debug Visuals

Spawn Setter

The spawn setter is small tool that allows to change the player spawn location, it does so by using empty game objects in the position of the spawn points.

The spawn setter allows for multiple spawn locations, however, **having more than one means that at the start of each simulation a random one will be chosen, this is not deterministic and could be a problem when gathering data**, this of course does not happen when one is present.

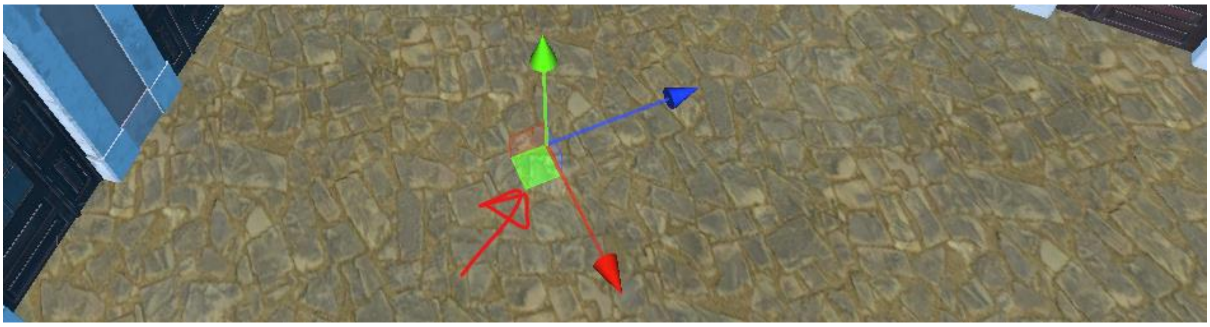
The Spawn Setter is in the hierarchy and can be accessed there, it has not UI as it works automatically with the simulation, any child under its hierarchy will be considered a valid spawn point.



A spawn point can be added by right clicking the Spawn Setter and then click on “Create empty”, this will create an empty game object under its hierarchy that will be considered a spawn point, However this will cause the coordinates of the point to be far away from the map, so a better way to do this is to just click on the existent point then press CTRL + D, this will duplicate the existing point.

It can then be moved by being clicked in the hierarchy, and after pressing W it will be able to be moved around the map freely.

We advise to move the points using only the green square to change the X and Z values without messing the Y value, which could cause the player to spawn under the map.



Scrub Through

The Scrub Through is a buildable scene that allows to re-visit player’s path via replay, it allows the replay to be seen step by step using the progress bar at the bottom.

How to run and open the replay files is covered in the Project Handbook, however this little section is to cover the controls, which are the following:

- W A S D to move the camera forward, back left and right.
- Q and E to move the camera up and down.
- Left click + mouse movement to pan the camera.
- Right click + mouse movement to rotate the camera.
- Shift + W A S D to move the camera at a high speed.