AnyWalker Tool

Map Generator Tool

November 2018

Sam Walet, Can Ur

Table of contents

Concept

AnyWalker is a Multi-Tool that turns our known ways of storing digital data to a new form of creative output for game developers and designers.

The tool combines the following:

A music based level generator for gaming.

This concept is to make a tool that takes any audio file and turns it into a platformer level. We will do this mostly through Beat and Frequency Detection along with general volume. Based on this data we will generate a level which is made up of a few basic actions the player has that can differ in length such as jumps and slides.

A visually-based level generator for gaming.

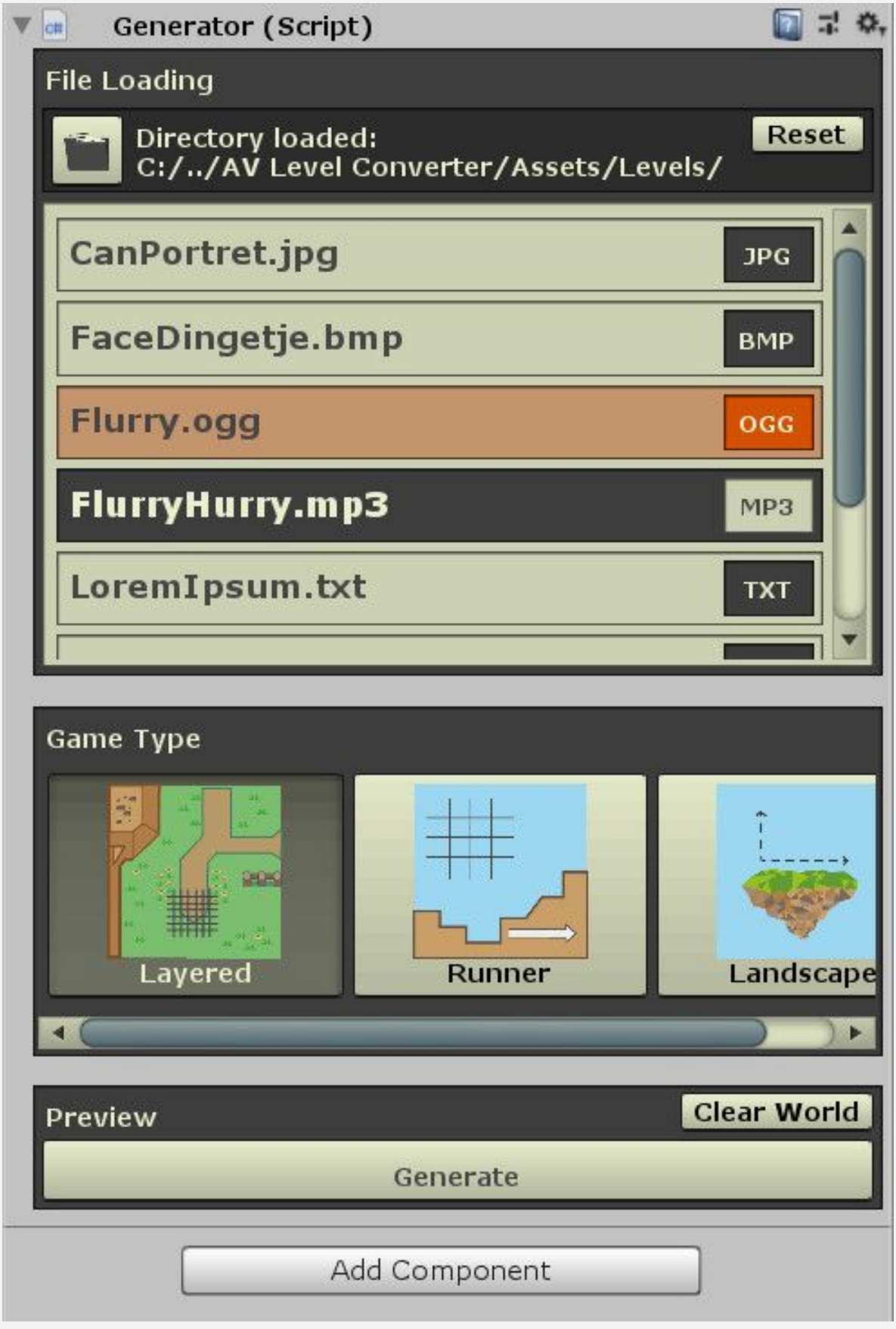
The visual side of this tool, will convert any image file into a playable game environment (in 3D). This will be achieved by scanning the colors of pixels, and encoding RGBA data into world-space coordinated generative functions. The user will be allowed to tweak some parameters, such as changing the gaming context to *Top-down, Landscape* or *Sidescroller*. These are generation types.

Another planned function (for the future) might give the user the ability to bind certain variables of the generation functions to specific colors. This will allow the tool to be much more diverse.

Or literally anything else.

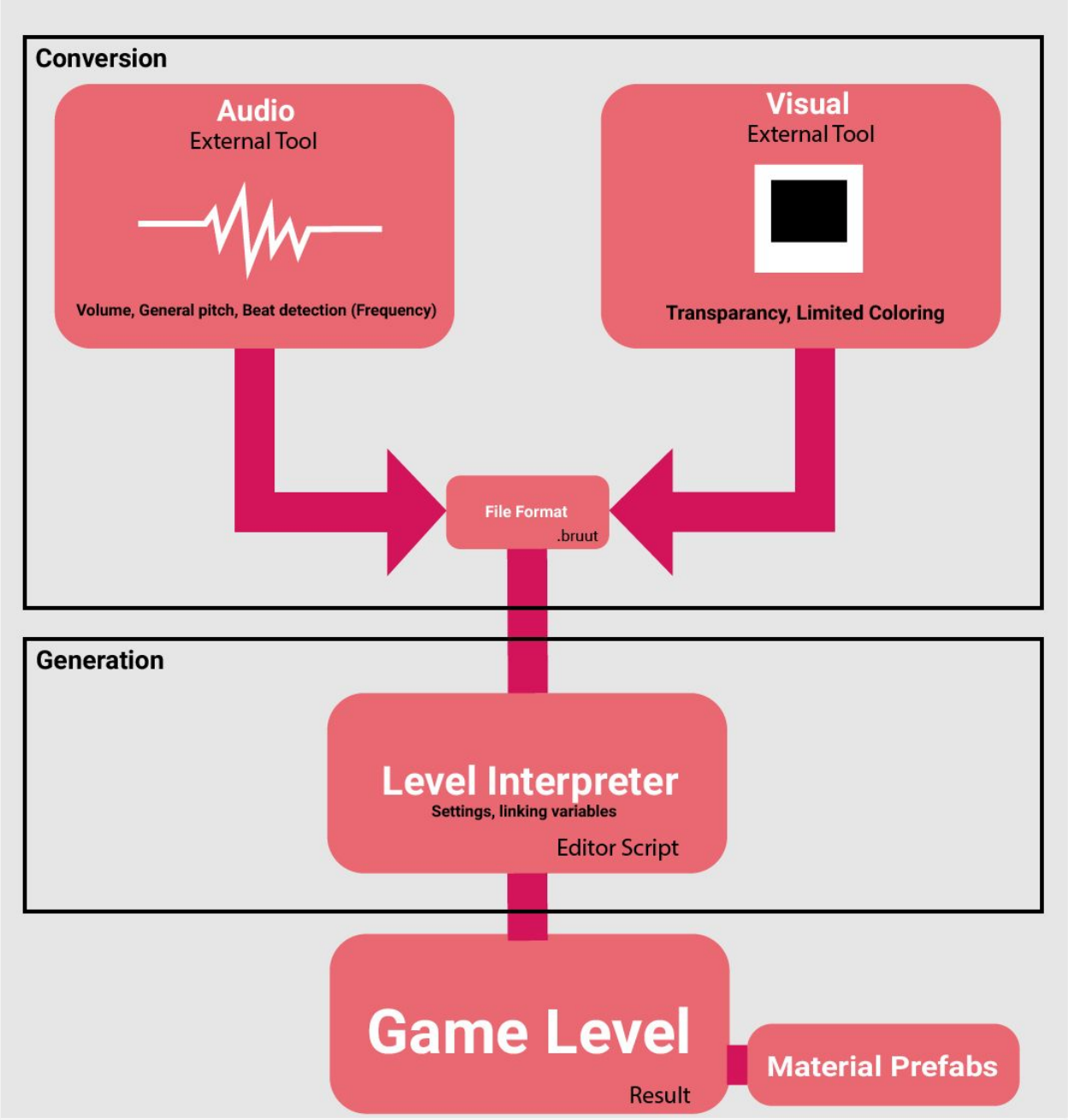
The tool will let you know if a specific file type is supported (yet) or not.
What it boils down to:
The User can convert *.txt, .mp3, .bmp, .asset, or anything* into a world!

After the generation, both of the generators will eventually convert and parse the encoded data into a single GameObject, and this will be your level. With a simple FPSController you can now walk on your generated level.


Generator (Script)
File Loading
Directory loaded:
C:/../AV Level Converter/Assets/Levels/
Reset
CanPortret.jpg
JPG
FaceDingetje.bmp
BMP
Flurry.ogg
OGG
FlurryHurry.mp3
MP3
LoremIpsum.txt
TXT
Game Type
Layered
Runner
Landscape
Preview
Clear World
Generate
Add Component

A level interpreter script file for Unity (Outdated)

This editor script will be an easy way of applying data file types to the Unity engine, and will bring them to life within a gaming context. Simple options include the scaling of the map, the intensity of the file data readings (so the map gets more or less extreme), and translate them into a different direction (so it adds depth to your level design) and more.



Visual chart of the linking systems of our concept. **(Old Iteration, the File format part is deprecated and as of now the Input)**

The chart above was the initial concept. During development some design choices were put through for the sake of consistency and simplicity. Such as, the file format (.bruut) being unnecessary. Everything is bundled within one inspector script, and all the input files *(.mp3, .txt, .jpg, .bmp, .png, etc..)* determine the algorithm that is used to read the file data and generate the world.

How to use?

The tool essentially has 4 steps in order to generate a level.

- **(1) File Selection**

This is the start of creating anything; You browse for a directory and simply select a folder. After this, the inspector script will let you know **which** files you can work with. The ones that have no generation code developed (yet) for them will be highlighted in orange/red and be unselectable.

- **(2) Game Type Selection**

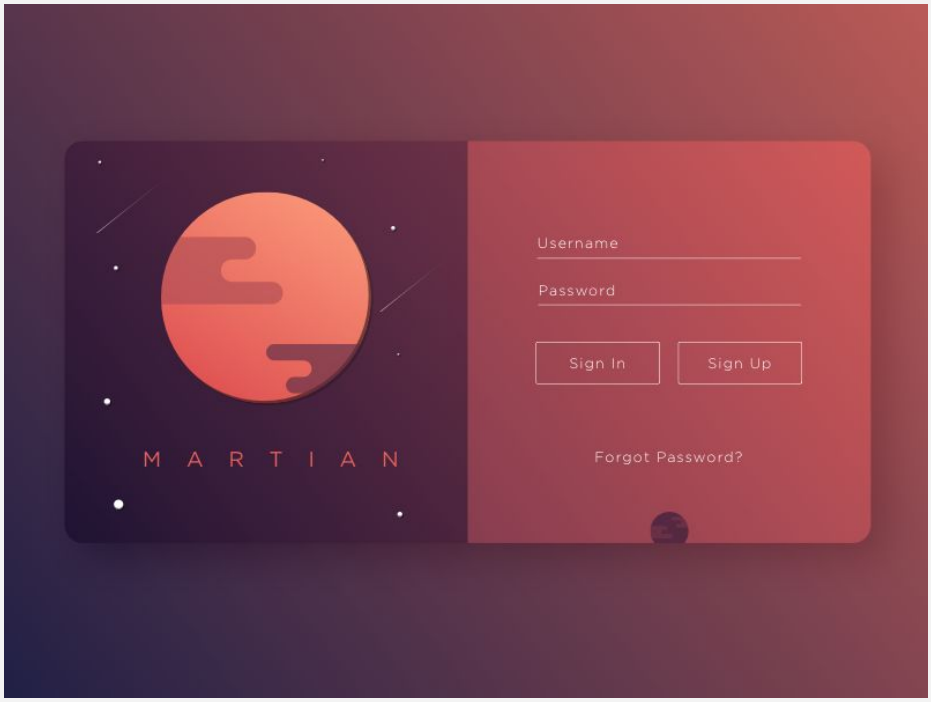
After choosing a file, you simply select one of the available Game Types. These are the *layout/rules* of how your level will generate.
If you choose **Runner** it will be a level format for a 3D platformer game. Whereas if you choose **Landscape** you will get a mountainous 3D terrain, great for survival games.
If you choose **Layered** you will get a voxel-type 3D world. This will probably work the best for indoor-environments.

- **(3) Preview Window & Modifiers**

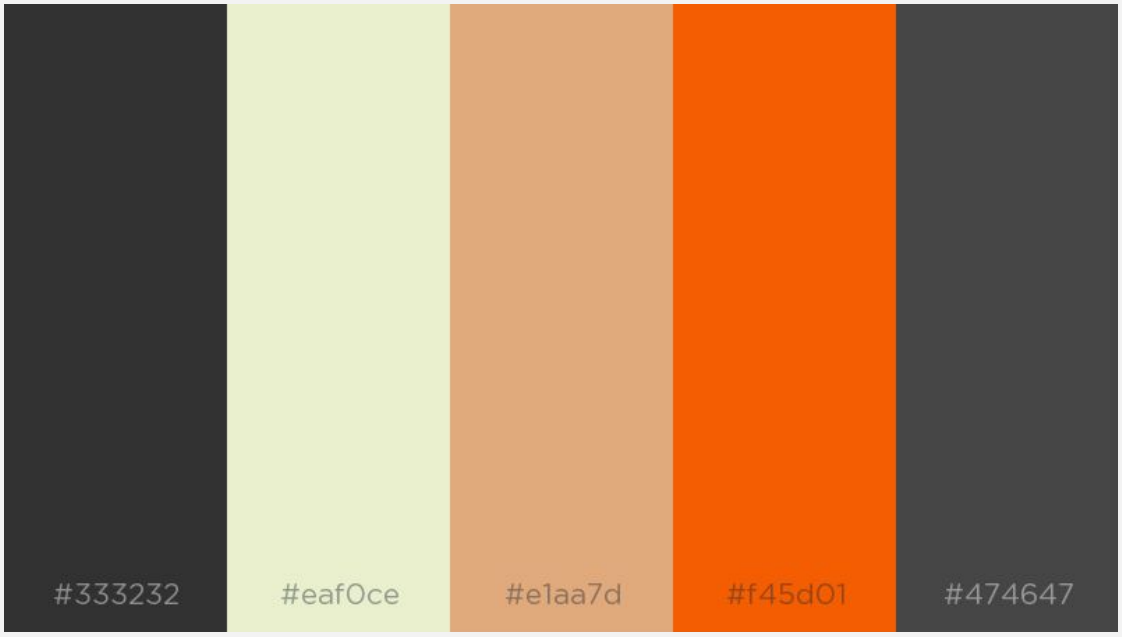
At this stage the data will be read of the selected file in part **(1)**, and will be converted to the generation format in part **(2)**. You will see a 3D preview of your generated level, and be able to pan around / tweak some stuff.
There is a bar at the bottom of your preview window with some settings, these are your modifiers. They range from simple ones as **Amplitude** to stuff like **Noise** and **Colors**. Depending on the file type, you can add Color to your landscape, change the density per data byte (so you essentially scale everything), etc. Tinker around to find out!

- **(4) Generate!**

Design



Inspiration for the UI design. **Flat, modern & simple color scheme.**



Color scheme for the A/V Level Gen Tool.

Second inspiration, reference material.

The Pico-8 cartridge/image system is an inspiration for the Visual part of this Tool.

Current Convertible Filetypes

.MP3
.WAV (Incomplete)
.BMP
.TXT
.DB
.SLN
.CSPROJ
.JSON
.LOG
.UNITY
.DOCX
.ASSET
.PDF

Our roles (in Dutch)

Sam Walet

Ik heb mij voornamelijk gefocust op het algoritme dat audio omzet naar data voor de generator. Dus het MP3Converter script. Bij de implementatie van dit script heb ik me ook wat bemoeit met de IConverType en voornamelijk de MP3 versie hiervan dus de audio gerelateerde functionaliteit komt bijna in het geheel bij mij vandaan. Ook heb ik mijn eigen experimentatie met de UI gedaan, en research naar JSON-serialisatie.

Can Ur

Ik heb de UI design gedaan, samen met het algemene programmeerwerk en het werk met de EditorWindow. Ook heb ik de generation presets en algoritmes geprogrammeerd, en dit allemaal gelinkt aan elkaar als één cohesieve eenheid.

Log

14-11-2018:

Mild brainstorming

17-11-2018:

Reference Research and Concept

19-11-2018:

Changed the concept. Made abstractions on the idea, to enhance modularity.

23-11-2018:

Color scheme, Inspirations & Design Conceptualisation

23-11-2018 (2 uur):

Messing around with FFT transform algorithms and tutorials in unity.

03-12-2018 (4 uur):

Researching and understanding algorithmic beat mapping using tutorial and downloading project to refactor for own use.

09-12-2018 (5 uur):

Modified the concept a bit, for efficiency and simplicity.

As the new plan goes, all the three main parts of the Conversion progress will rather be integrated into one Unity editor script, the Generator. This script will have a directory selection panel, after which it will have its own file-browser. Here the user will be able to click on all the files in this directory, and eventually test them on the level-ness of the files. This way, the script will handle most of the file types (for now Audio .mp3, .wav, .ogg and Image filetypes .png, .jpg, etc.)

The script will have some modifiable settings, to change the *Level Types* from Top-down to Sidescroller. (Maybe more in the future aswell)

All these things above are stored as temporary data within a generated .JSON file.

10-12-2018 (4 uur):

Added a preview editorwindow that will show your converted level, allowing you to tweak some variables and eventually generate it (putting it in your scene).

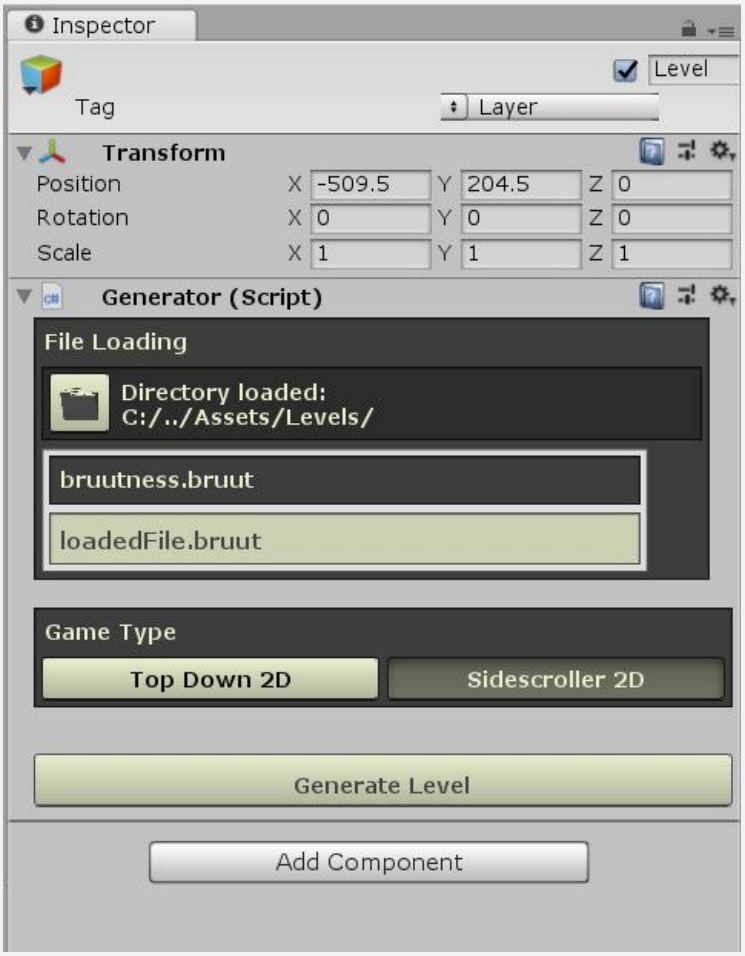
11-12-2018 (5 uur):

Refactoring spectral flux analyzer to analyze specific frequency ranges as well as returning the peaks in those ranges in dictionaries.

18-12-2018 (6 uur):

The file panel now scans for files which have a valid conversion algorithm, if this is not implemented yet, the file is unclickable. Diverse additions / bug fixes / optimizations. Code architecture is a lot more efficient now,

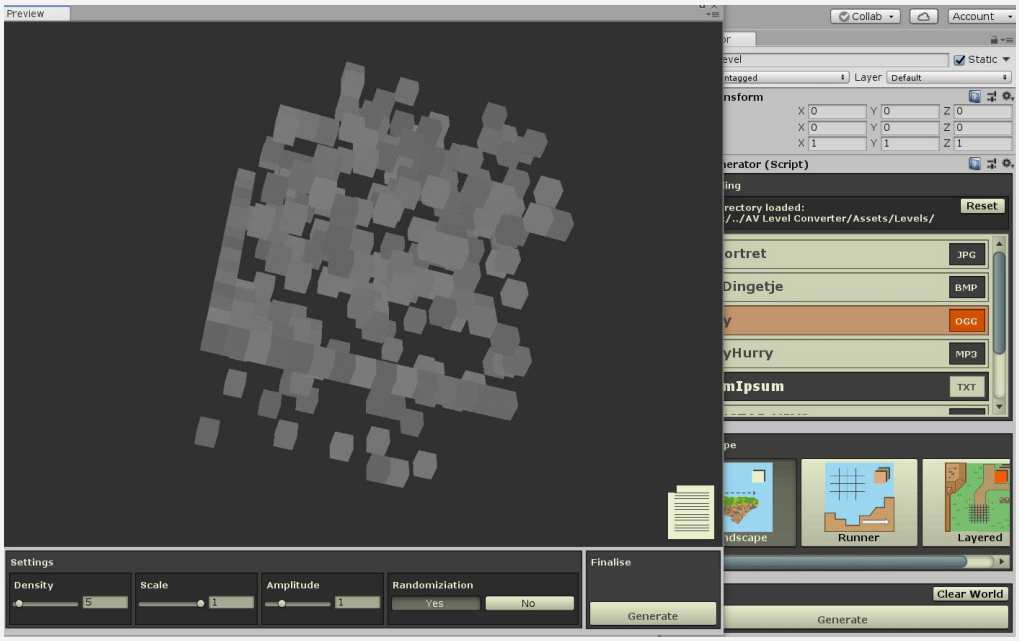
combining an interface type “IConvertType” with an abstract class inheriting from it, allowing for easy modification.

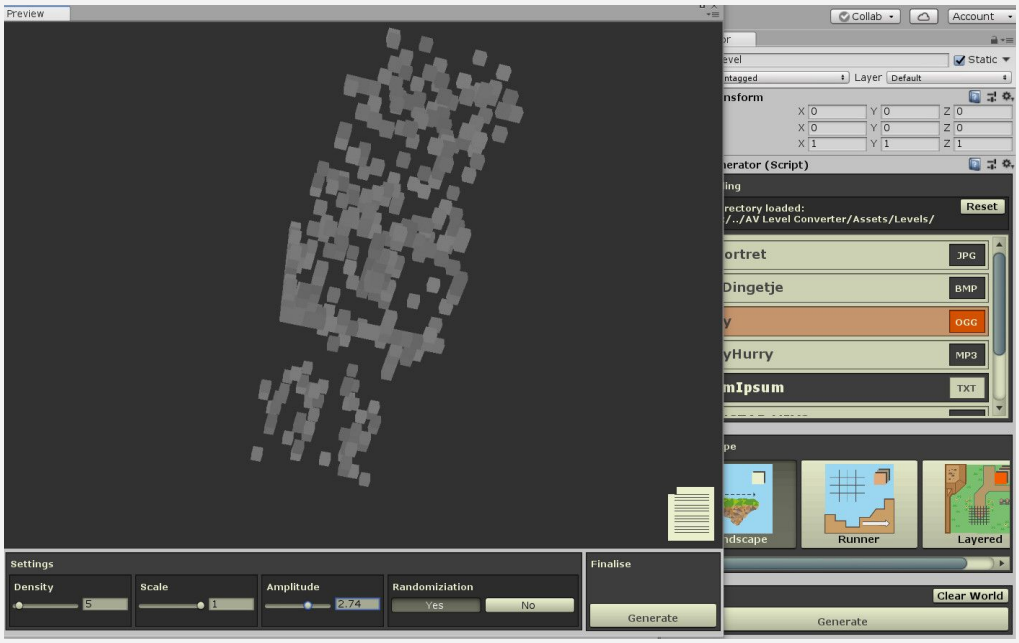

Inspector
Level
Tag
Layer
Transform
Position
X -509.5
Y 204.5
Z 0
Rotation
X 0
Y 0
Z 0
Scale
X 1
Y 1
Z 1
Generator (Script)
File Loading
Directory loaded:
C:/../Assets/Levels/
bruutness.bruut
loadedFile.bruut
Game Type
Top Down 2D
Sidescroller 2D
Generate Level
Add Component

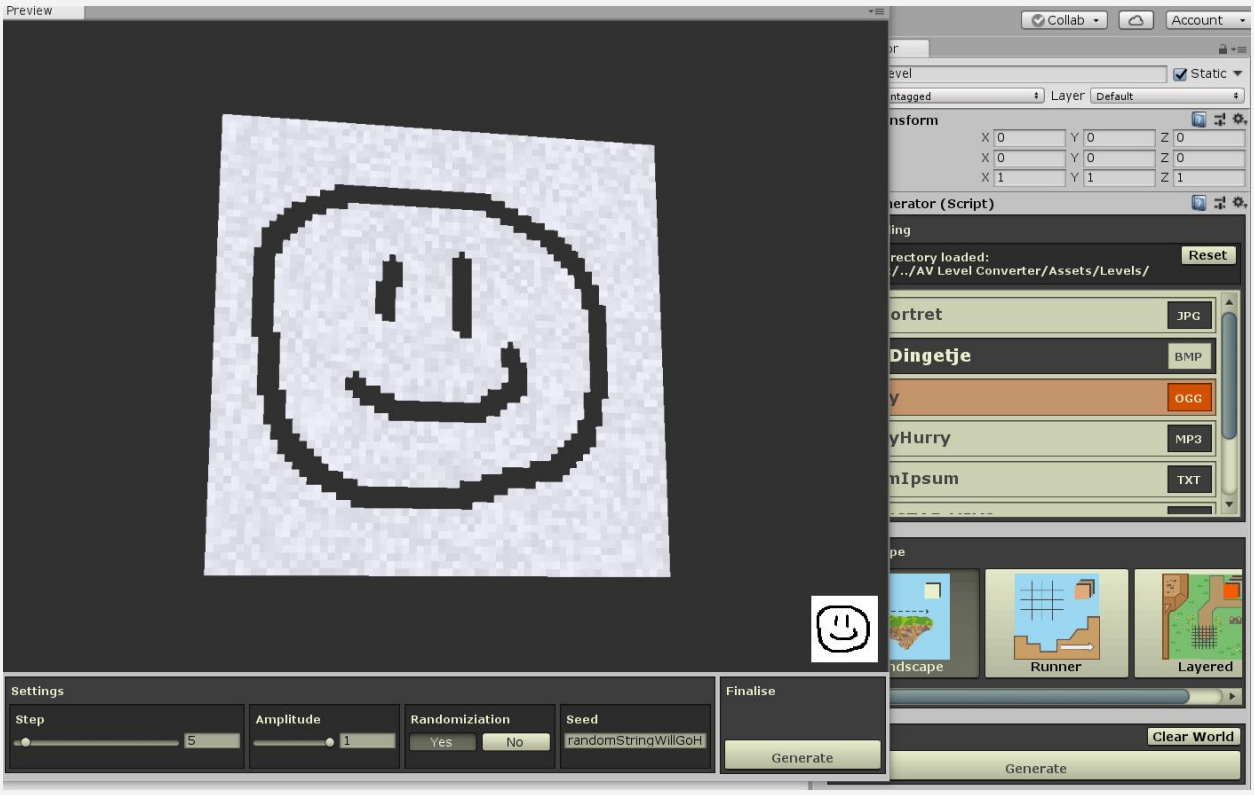

Generator (Script)
File Loading
Directory loaded:
C:/../AV Level Converter/Assets/Levels/
Reset
CanPortret
JPG
FaceDingetje
BMP
Flurry
OGG
FlurryHurry
MP3
LoremIpsum
TXT
Game Type
Landscape
Runner
Layered
Preview
Clear World
Generate
Add Component

19-12-2018 (6,5 uur):

.TXT Level Conversion method prototype developed.
Further code architecture improvements.
Last but not least, the changeable settings in the Mesh Preview window are now linked to the direct generated model.






Preview
Collab
Account
Static
Layer
Default
Reset
JPG
BMP
OGG
MP3
TXT
Runner
Layered
Clear World
Generate
Settings
Step
5
Amplitude
1
Randomiziation
Yes
No
Seed
randomStringWillGoH
Finalise
Generate

20-12-2018 (4,5 uur):

Fixed the audio FluxAnalyzer script, which streams audio files and outputs a list of volume peaks, low-, mid- and high peaks at certain times. This gets converted into leveldata, thus linking the level generator to the audio analyzer.

21-12-2018 (5 uur):

Found a way to get around Unity's unsupported .mp3/MPEG streaming, while not sacrificing the current File loading system (which doesn't use Resources.Load, allowing users to also load local files). Created a conversion method for loading raw bytes in an array through File.ReadBytes(path), which converts them to a float[], usable within an AudioClip.
https://stackoverflow.com/questions/51051559/converting-mp3-byte-array-to-float-array-to-play-with-unity

- TL;DR - Added generation support to .MP3 and .WAV files, with adjustable amplitude control.
- Added a system for the File Selection panel, which ignores certain filetypes (.meta, .sys, .gp5, .asset, .dll, etc.)

Current convertible filetypes: *.MP3, .WAV, .BMP, .TXT, .DB, .SLN, .CSPROJ, .JSON, .LOG, .UNITY, .DOCX, .ASSET*

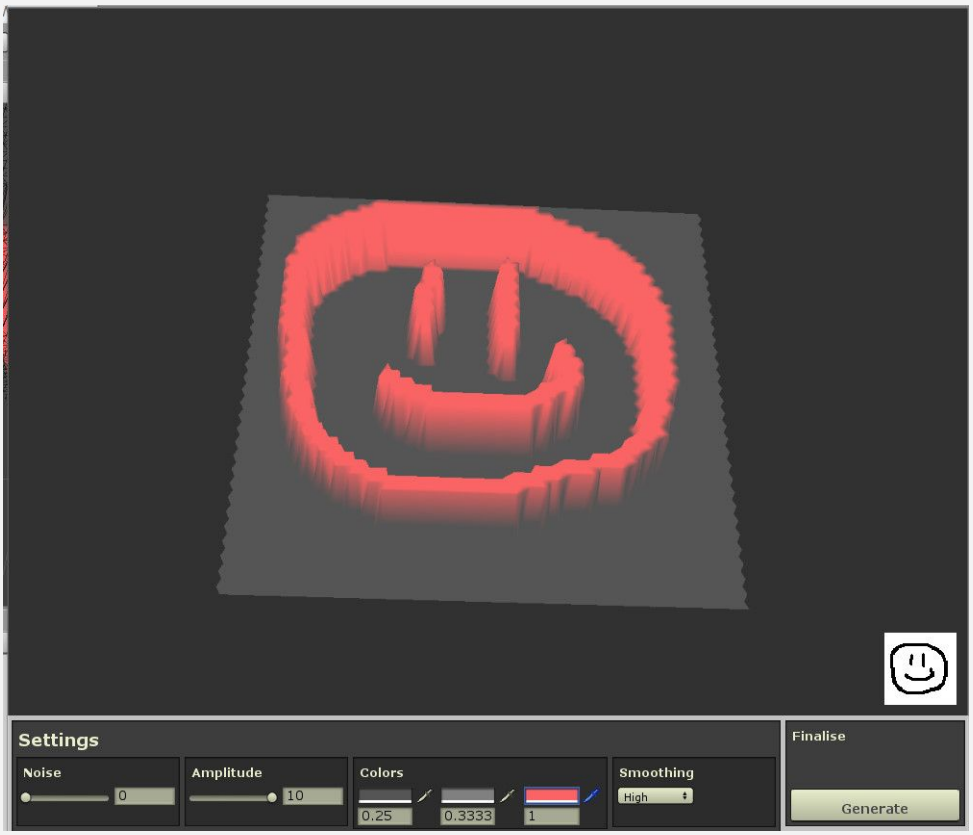
- Added Generation size caps, preventing too big files to be loaded and freezing the tool.
- Added a check whether the file is in use or not. If this is the case, the preview window doesn't open, and an error message gets printed.

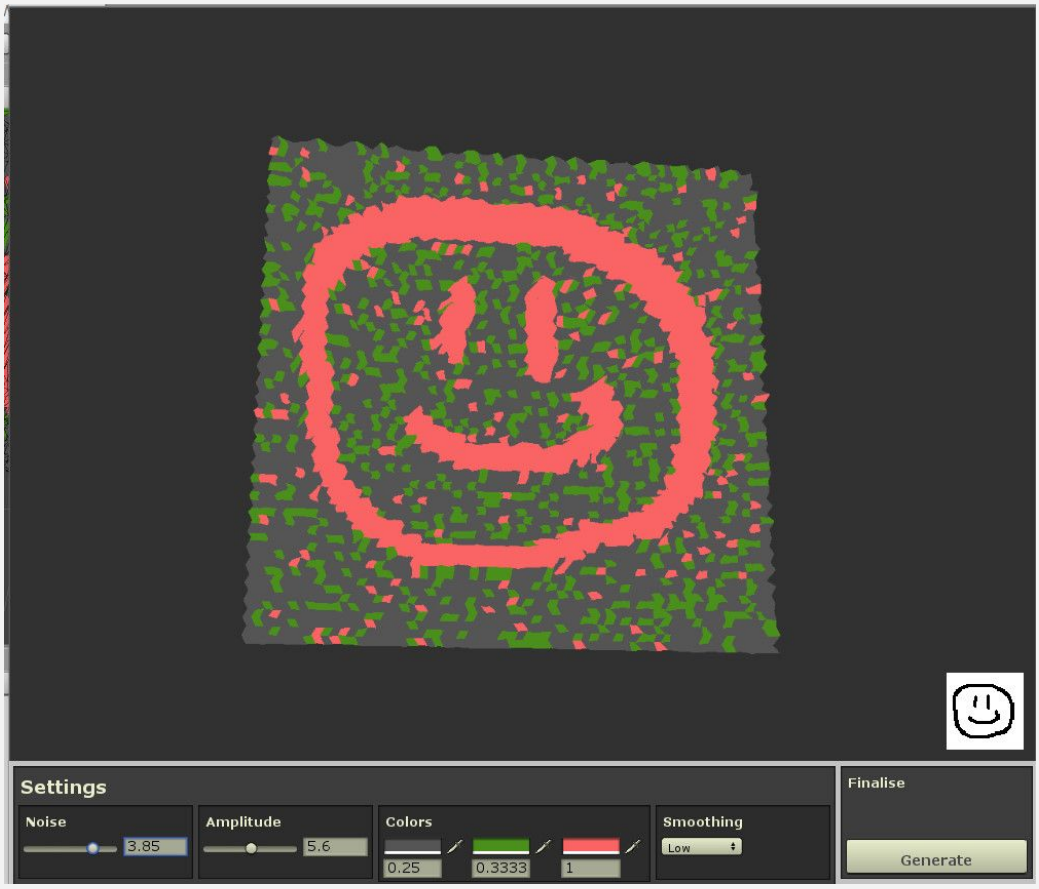
26-12-2018 (3 uur):

Finalised tweakable settings for the .BMP generation type, started working on the Infinite Looper generation (Runner).

28-12-2018 (5 uur):

Added Noise, assignable terrain colors & smoothing settings to the .BMP Terrain generation.


Settings
Noise
0
Amplitude
10
Colors
0.25
0.3333
1
Smoothing
High
Finalise
Generate


Settings
Noise
3.85
Amplitude
5.6
Colors
0.25
0.3333
1
Smoothing
Low
Finalise
Generate

09-01-2019 (2 uur):

User Experience

Added custom mouse scrolling for the GameType horizontal scrollbar.

Several cleanups regarding coding convention consistencies.
Added generated collider to the Terrain Generation type.

11-01-2019 (4 uur):

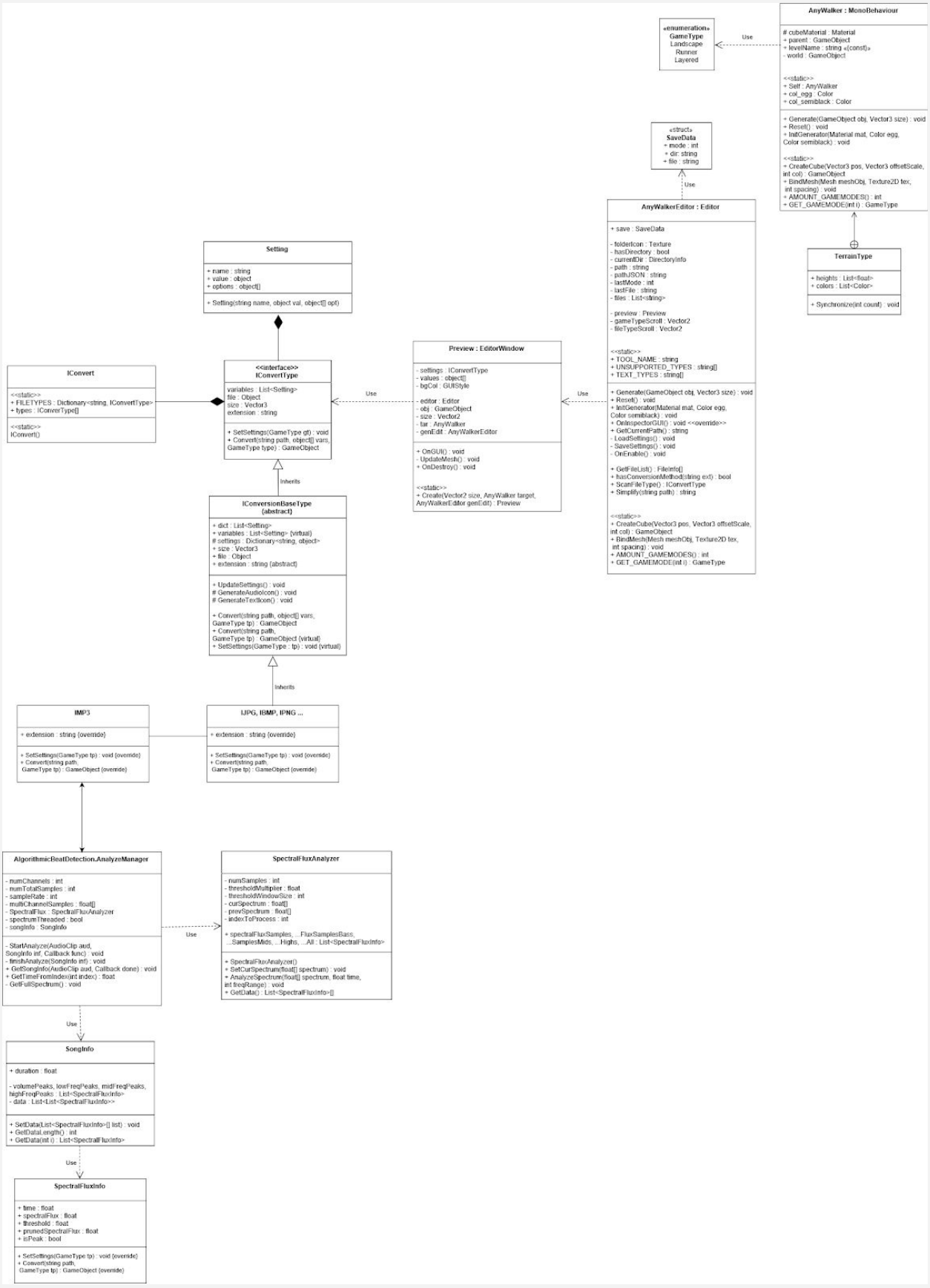
Optimization, created caps for too large files (such as .MP3 files); So it doesn't take too long to load and process data. Also polished some settings for .TXT types and tinkered with the generation settings.

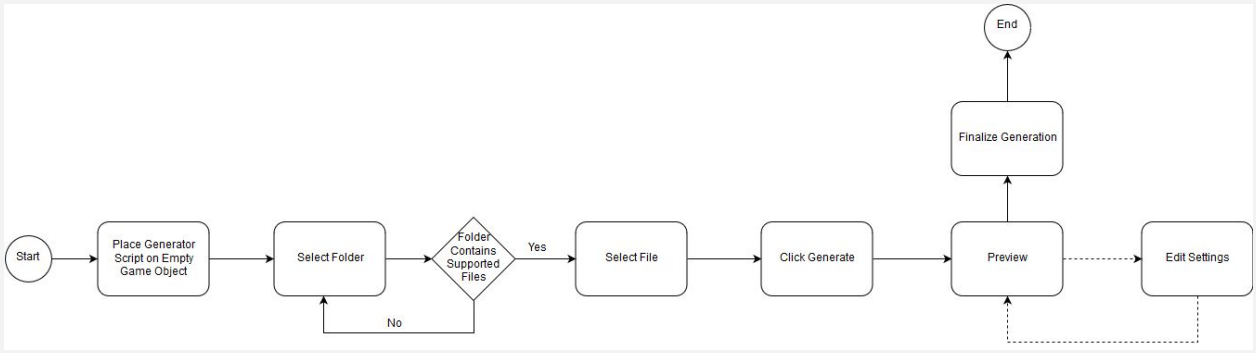
Creation of usertest form + Usertesting (4 Game Designers + 1 Technical Artist)
https://goo.gl/forms/Ax9aXkMhIfQtRfvk1

12-01-2019 (3 uur):

UML chart creation + Further documentation & Code consistency polishing

UML + User Activity Flow




End
Finalize Generation
Start
Place Generator Script on Empty Game Object
Select Folder
Folder Contains Supported Files
Yes
Select File
Click Generate
Preview
Edit Settings
No

Usertest

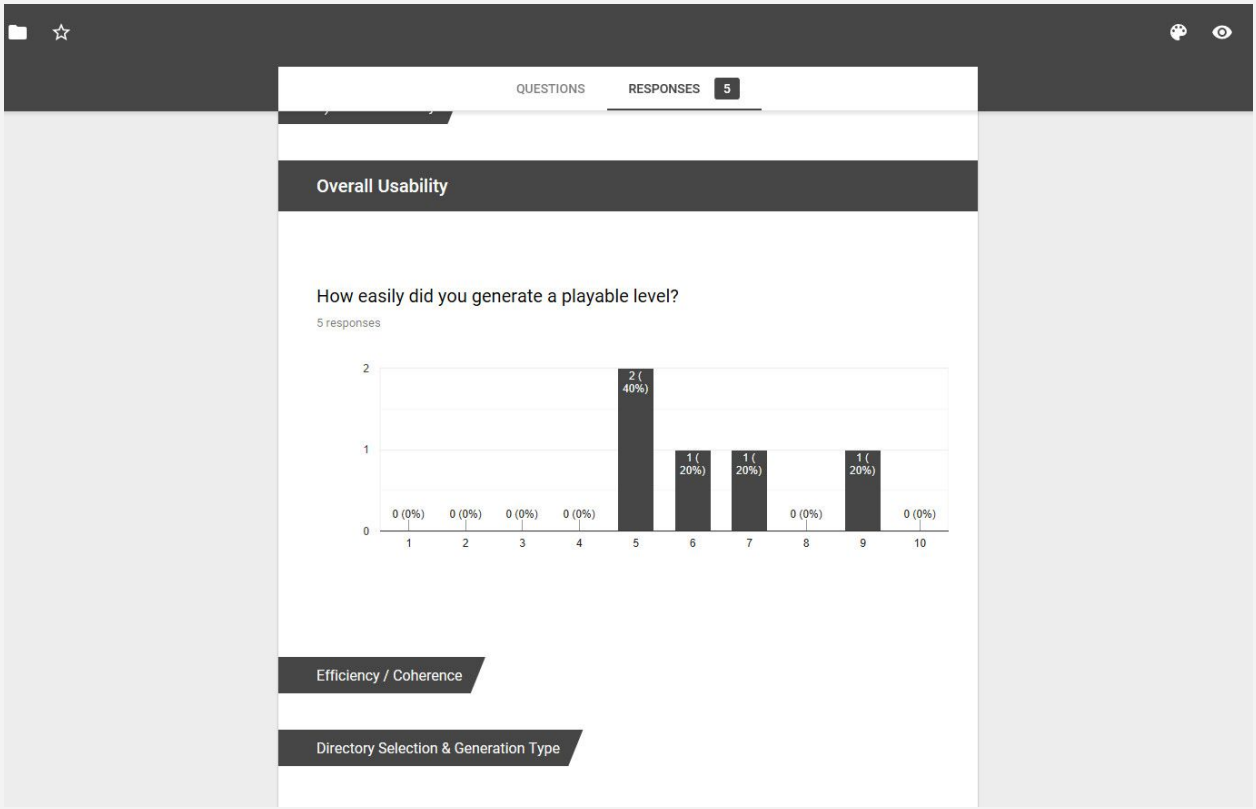
The tool was specifically designed with Game Designers as a target group in mind. We created a digital feedback form, and got the following to participate:

- 4 Game Designers

- 1 Technical Artist

Link: https://goo.gl/forms/3h2f3TM2xHBZlVr72

Here are the test results:



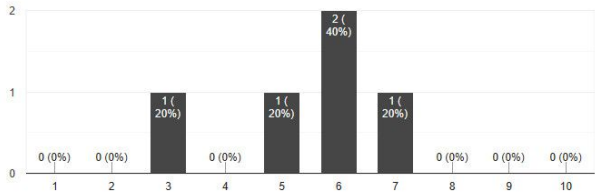
QUESTIONS RESPONSES 5

Efficiency / Coherence

Directory Selection & Generation Type

How clear are the controls / options you have as a user?

5 responses

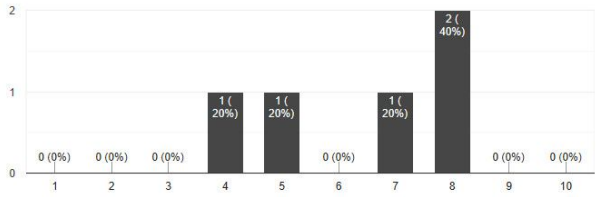


Do the Game Types (generation types) give you a fair idea of how the level will generate?

QUESTIONS RESPONSES 5

Do the Game Types (generation types) give you a fair idea of how the level will generate?

5 responses



Comments / Feedback?

3 responses

Some features, such as frequency or amplitude are hard to grasp. Especially since the effect isn't directly noticeable (at least for me) in the preview window

QUESTIONS RESPONSES 5

Comments / Feedback?

3 responses

Some features, such as frequency or amplitude are hard to grasp. Especially since the effect isn't directly noticeable (at least for me) in the preview window

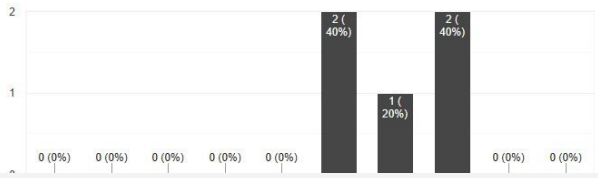
Try to add a section where you can set rules, and allgorithms

Took me a little while to get what I was supposed to be doing with these butons but it's pretty easy after a while

Generation

How clear are the settings / options you have as a user?

5 responses



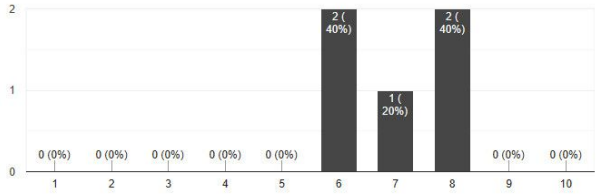
QUESTIONS RESPONSES 5

Took me a little while to get what I was supposed to be doing with these butons but it's pretty easy after a while

Generation

How clear are the settings / options you have as a user?

5 responses



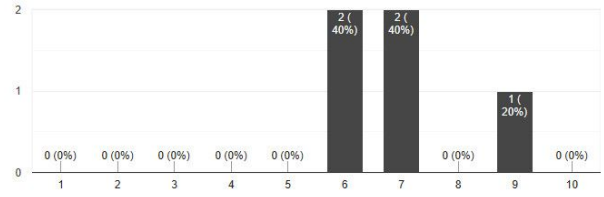
Were the choice of modifiers (such as Density, Amplitude, etc.) interesting enough to experiment with?

5 responses

QUESTIONS RESPONSES 5

Were the choice of modifiers (such as Density, Amplitude, etc.) interesting enough to experiment with?

5 responses



If not, what would you have liked to experiment with instead?

1 response

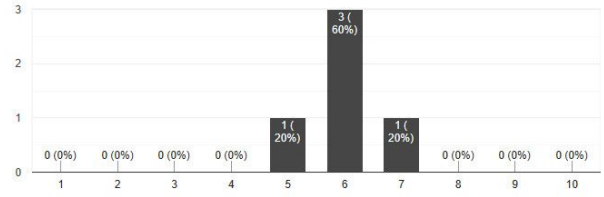
place gameobjects at pixels with certain colors

QUESTIONS RESPONSES 5

place gameobjects at pixels with certain colors

Were the modifiers usable in generating a playable level?

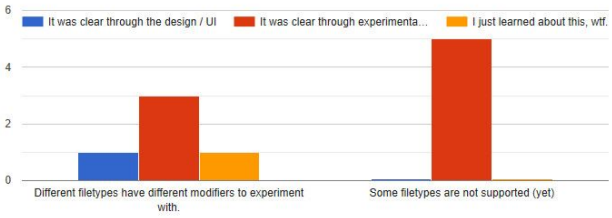
5 responses



Did you discover/know these following things:

QUESTIONS RESPONSES 5

Did you discover/know these following things:

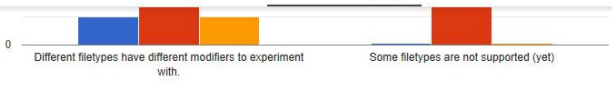


General Comments / Feedback?

3 responses

Useful for quickly prototyping levels that a designer quickly sketched together in somethiing like paint. Could maybe fit in with unity more (why is the file selector something you rolled yourself?). Would like to see something that helps place functional items in a level based on a colorpicker or something.
not possible to go back to a previous variant with ctrl-z and no way to save presets into scriptable objects for

QUESTIONS RESPONSES 5



General Comments / Feedback?

3 responses

Useful for quickly prototyping levels that a designer quickly sketched together in somethiing like paint. Could maybe fit in with unity more (why is the file selector something you rolled yourself?). Would like to see something that helps place functional items in a level based on a colorpicker or something.
not possible to go back to a previous variant with ctrl-z and no way to save presets into scriptable objects for example. sometimes a bit slow with changing parameters. parameters have a value in a textbox, but you can't type in a number here?

Regarding the question: Were the modifiers usable in generating a playable level: I think it's hard to get a true idea of how playable a level is if there aren't any controls or game yet. But I were able to imagine a game that could use the generated levels, so in that case, I do think the generated levels are usable.

Cool idea as a designer it would be very nice to have a tool like this to just quickly test out some mechanics.

Look & Feel

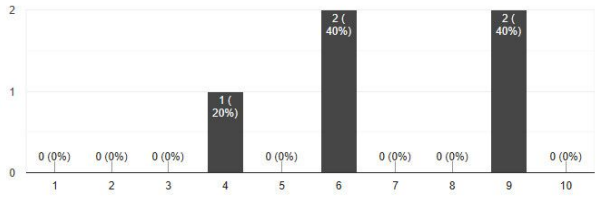
What do you rate the color scheme / overall design?

QUESTIONS RESPONSES 5

Look & Feel

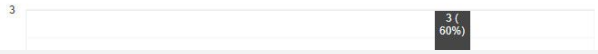
What do you rate the color scheme / overall design?

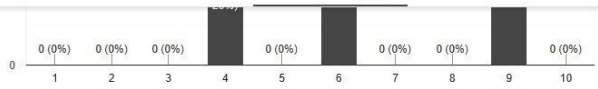
5 responses



How is the overall flow of the tool?

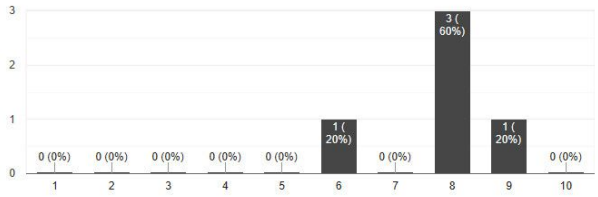
5 responses





How is the overall flow of the tool?

5 responses



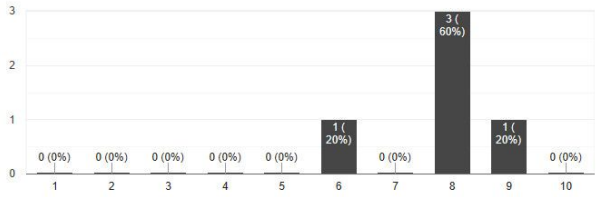
Comments?

3 responses

I would change the color of the that the unusable files are highlighted with. Currently I you use orange, which to me

QUESTIONS RESPONSES 5

5 responses



Comments?

3 responses

I would change the color of the that the unusable files are highlighted with. Currently I you use orange, which to me conveys a clickable /interactable object because the overal color scheme is very monotone and cold, which I know as unusable/uninteractable objects.

more grayish and more formal

When fucking around in the editor it did get very stuttery but I don't know if you can do anything about that but i really like the design of the tool itself

Test Conclusion

From testing we have learned that it is still kind of clunky, and that the tool could benefit from some optimisation (regarding the World Preview editorwindow).
Besides that, it can be noted that people had fun messing around in this tool, and creating levels. The UI communication needs to be a bit more coherent about whether file-generation types are implemented yet or not (which was specified with the orange color, and not being able to select those files).

These test results will be incorporated in future builds of this tool. It was very helpful to see how the tool performed in front of our target group.

Criteria

- Error handling; Corrupted files/inputs?
The current generation types are very basic; They make sure to only use usable characters, numbers or data within the filetype.
Also, file types that don't have a generation method linked to them, will not be selectable in the first place.
- Backwards/forwards compatibility: Updates of the tool; What happens when the same input is applied to different versions of the tool?
The generated levels are separated from the tool; Once the Generate button is clicked, the current configuration will be outputted into a level of GameObjects/Meshes.
However, the Generation methods per filetype are architecturally designed to be replaceable / changeable. Which means, the same file will result in a different output depending on the algorithm.
- UML + Activity/Flow/User Experience Diagram
- Design Patterns: (Interface -> Abstract Class -> Concrete Class) + Dependency Injection (IConvertType.cs) + Singleton (BMPLoader.cs) + Factory Method Pattern + Observer Pattern (SpectralFluxAnalyzer.cs / MP3 Loader Class)
- Conventions / Readability: Comments + Consistency
- User tests + Documentation and presentation 4 Game Designers + 1 Technical Artist Usertest
- Data Driven structure / Serialization: JSON

Links & Sources

Fast Fourier Transform

Digital Signal Processing (DSP) Tutorial - DSP with the fast fourier transform Algorithm
https://www.youtube.com/watch?v=HJ_-5mqUZ70

Beat Detection

[Unity Engine] Beat Detection
https://www.youtube.com/watch?v=0i9hi9k1K10

Tempo and Beat Tracking
https://www.youtube.com/watch?v=FmwpkdcAXl0

How to do FFT in Unity
https://answers.unity.com/questions/974565/how-to-do-a-fft-in-unity.html

Algorithmic Beat Mapping In Unity: Real-time Audio Analysis
https://medium.com/giant-scam/algorithmic-beat-mapping-in-unity-real-time-audio-analysis-using-the-unity-api-6e9595823ce4

Procedural Mesh Generation
https://gamedevacademy.org/complete-guide-to-procedural-level-generation-in-unity-part-1/

Melody's Escape - Procedurally Generated Platformer

Melody's Escape is a music based procedurally generated platformer which is what the tool is supposed to do. It has a handful of actions (jump, slide, attack) that are placed based on the music. There is also platform height and the balls that you pick up. Based on the title and video I would say Melody's Escape looks at a lot more info than just rhythm like pitch and volume.

Slayer - Angel of Death on Melody's Escape

https://www.youtube.com/watch?v=OhOAXXIsvfw

Audiosurf 2 - Procedurally Generated Racer

Audiosurf 2 is much simpler than Melody's Escape in many ways, it only generates notes on a three lane track. The yellow ones have to be hit and the grey ones avoided. This system is more akin to guitar hero.

Pantera - Death Rattle on Audiosurf 2

https://www.youtube.com/watch?v=VM7kQFOHI2Q

TERRORHYTHM - Procedurally Generated Beat 'em Up

TERRORHYTHM once again has a number of actions (attacks, different kinds of enemies etc.) that are done by the player based on the music. However this game has a strong focus on rhythm and the beat not being as concerned with melody so tempo and rhythm are most important here.

Metallica - Through The Never on TERRORHYTHM

https://www.youtube.com/watch?v=BKkB4vOOfWI&t=43s