DOI: https://doi.org/10.28925/2519-4135.2024.51

VOLODYMYR SNIHUR
Postgraduate student
Borys Grinchenko Kyiv Metropolitan University
ORCID ID 0000-0001-6142-8121

Processed graphics in game development: quality assurance and testing

Abstract. While AI-generated and otherwise processed assets are still relatively new topics, their less sophisticated counterparts are not. Custom-made generated graphical assets may be useful in game development as placeholder assets for prototyping, background graphics for side scrollers or simple character sprites. There is another possible use of them in the field of quality assurance and game testing, particularly live playtests (playthroughs by actual human players instead of algorithms). QA can be done by game developers, publishers should they choose to do so or players and in some cases by mod authors and mod users (provided the game in question can be modded at all). Following article explores actual, already released, practical examples of such modifications in the form that they were considered as "most useful for the end user" and fulfilling their intended task. All of them, as well as principles they are based on, can be useful for the development of future games and interactive content in professional or academic settings, since the base assumption of wanting to produce best possible quality of work that is sufficiently optimized and can

work on PCs of varying performance levels is universal to both of them.

Keywords: game development, generated textures, digital art, optimization, testing.

Introduction. Modern game development is posed to employ as much of generative AI and neural networks as possible, however, this does not mean that uses of specifically generated assets are anything new. Matter of factly, what differs these examples from previous times is the scale, fidelity and ultimate en-uses of such assets. Some of them are well-known, others obscure and yet some are so specific that might as well not exist for anyone outside of specific groups. All of them deserve exploration and to serve as a base point for further research, however non-extensive the list may be. Therefore, the purpose of this article is to provide a short analysis of some chosen examples from the standpoint of an artist actively involved in the relevant fields. We will attempt to not go into specifics of technical implementations, workflows and other questions relevant to either specific examples we have chosen to work with or the topic in general. However, we have to note that some prior familiarity with image formats is paramount to the success of any project involving generation or editing images en-masse. Some relevant specifics of how underlying game engines will work with graphics is also necessary knowledge. For example, recent versions of Creation Engine will be more than happy to see BC7 compressed DDS (DirectDraw Surface - texture format) files, whereas older versions will not. Facts like this are relevant since both versions will be present on market, usually bundled with respective games and sometimes coming with user-friendly (but still hard to use) game-development kit.

Search for missing assets. In some cases it is easier to work with the asset library by hand, in others it will be better to make custom code or leverage already existing solutions. Here we shall look at one curious case where making your own program to take care of a tedious job is the best course of action, or at least not the worst one, unless an option of ready-made mod or other solution is available. It is not uncommon for games to be modded despite and sometimes in spite of their developers actively hampering such efforts. In other cases players will on the contrary receive full support or at the very lesatsilent approval of the studio. There are examples of titles allowing for full replacement of every model, sound, texture, down to game scripts and engine-level code alterations (latter being extremely hard and time consuming to do and can stop working after even a slightest tweak was made to game executable during update).

Such is a case with Bethesda's Elder Scrolls V: Skyrim. Sufficiently savvy modders can turn it into almost anything they desire, provided there is ample amount of mods fit for the purpose and if there is none, they can make one themselves. Most common part of building mod lists starts from small fixes eliminating engine bugs, model inconsistencies and other annoyances, later progressing to texture and model replacement. While simply installing mods (in reality we can copy them in game's main folder, which consists the entirety of "installation" process) will be sufficient in most cases, their load order will need to be sorted to ensure compatibility and error-less function. This being a process deserving an exploration on its own, we however are more interested in more aesthetic part of it, since it is at this stage where it might be crucial to do two things at once: choose the intended visual style and ensure

that every relevant asset is actually updated/upgraded/replaced with their (hopefully much better) counterpart. Former is up to individual preferences while the latter can be aided with the help of several authors; two examples of such helpful creations are being looked at below.

0.01K Vanilla Textures [2] turns every possible texture into 16x16 pixel squares for the purpose of providing a clearly visible baseline "bad" texture that is to be replaced. Intended function is to override default files (game engine allows this, including multiple successive overrides of the same files) without changing them. In doing so we preserve originals and keep them as fallback solution, while replacing desired aspects with our chosen versions. This mod works perfectly well and does not overload the player's computer too much, while providing something mostly pleasant to look at while testing texture replacers. Only downside being that it does not provide as obvious a difference as it could and should have been, instead making every surface look blurry.

This perceived flaw became the reason for another author to have made their own version, which if takes up significantly more storage, does fulfill previous creation's purpose much better. Excessively Pixelated Textures [3] is dedicated to, in author's own words, "making sure you really have replaced all the textures" serving as a "blindingly obviously pixelated" baseline asset replacer. All of the images were made using a custom program made for generating images consisting of random distribution of variously colored pixels (it supports the option of producing single-color images as well). Mod itself was purpose-built to be as noticeable and chaotic as possible, highlighting everything that has any kind of texture applied (rain, UI, characters, plants, landscape, etc). While it definitely

provides a much better and easier way to detect missing or not replaced assets, one of the downsides is that the game becomes as uncomfortable to play as it can be, the upside being that it is needed only once and only to set up a good enough graphical overhaul. Other possible issue is that every image file, except for character face textures, was replaced, making normal maps (used for faking the lighting of bumps and dents), specular (used to simulate mirror reflection qualities), glow maps etc., etc. into mesmerizingly pixelated chaos. At the very least users will be sure to replace those as well if they want to (which is doubtful as in testing diffuse i.e. color information files will be the focus and have the most impact on visual quality). Nevertheless, the same principles can be applied to sounds and models if desired.

Optimization and better graphics. Obviously, it is the duty of game developers to make their product as optimized as possible if not for user enjoyment then to save on hosting costs (which is alleviated by using platforms like Steam, but not removed entirely). Sometimes users take matters into their own hands for a variety of reasons. Such attention has not escaped above mentioned Skyrim as well as the entirety of Elder Scrolls and Fallout game series. Here we will look at two more examples of using algorithmically processed assets for improving general game performance characteristics.

PIXELRIM - Skyrim Ultra Reduced Textures [7] and its companion mod Blank Normal Maps for All Textures of SSE [6] are two parts of our first example. First of them is reducing every diffuse texture to thematically chosen single color 1x1 or 4x4 pixel squares, depending on what part of its description to believe. Downloading “basic installation” archive and checking it we find that:

- actual texture size is 4x4 pixels;
- everything except for character faces was resized.

Given the number of files included (25579) it is entirely possible that another purpose-made program was used: Cathedral Assets Optimizer. [1] CAO can resize and convert files to various DDS formats, sometimes resaving them in what would be best for the game in question, since it supports multiple options. This makes every file in the above-mentioned mod to be, technically, a result of work of custom code, therefore a processed asset. When it comes to the "Blank Normal Maps" file, it is possible to use another workflow:

1. unpack base game archives with Cathedral Assets Optimizer;
2. search and delete avery file without '_n' substring in its name;
3. create blank normal map file and save it in any preferred file format (bitmap will be sufficient);
4. get names of every file remaining after step 2;
5. save new normal map using names from previous step (overwrite if needed or save to another location preserving folder structure);
6. convert new normal maps into DDS files using Cathedral Assets Optimizer.

Expected performance gain is minimal as Bethesda games are frame-rate capped at 60 frames per second maximum (higher counts are possible but will require engine-level changes as standard "fps increase" will create problems for physics engine as its physics calculations are tied to framerate).

One more creation takes a slightly different approach to optimization, instead producing entirely new, potentially rescaled, files, it tries to restore the original look of them and

only then compress using appropriate algorithms. Cleaned Skyrim SE Textures [5] uses two neural networks (here called “models”) to first restore files to original state, effectively decompressing them as best as possible and to then remove resulting compression artifacts, errors and various defects. In doing so the author aims to achieve the state close to original to have a suitable base line quality to work off of. Later these files were re-compressed using suitable compression algorithms. Upscaling however, was left to become a mod of its own since this is technically outside of the scope of the original idea. During comparisons of original textures and their cleaned versions, the latter win in terms of clarity, color range and absence of “edge highlights” of some pieces.

If the goal is to process files of a certain pixel size, then it is better to employ a Texture Sorter. [4] Custom-made program designed and made specifically for categorization, deduplication (it will flag only 100% duplicates, regardless of their appearance or file extension they supposedly have) and safe copying of the files to the work folder preserving original folder structure and original files as fallback solution. Functionally none of the steps in our workflow need to change to account for this program. Only difference is that instead of either doing the job manually or let CAO or other program of choice to decide if it should process files or not based on some, largely arbitrary, criteria, we will have a clear selection of samples to work on. Suppose we have a mod retexturing houses, rocks and furniture, each group has wildly varying texture sizes. In such a hypothetical scenario CAO would by default downscale every texture larger than target size, with upscaling functionality working the same way but on files smaller than target. However, it is overall more desirable to

have rocks and distant objects keep their original textures, potentially giving them much higher quality dimensionally larger ones if possible (reason being that game frequently uses literally the same file for regular boulders and mountainous cliffs). Sorting them lets us work on some selection of files, test them independently of each other and then merge original and various processed versions in a way that most satisfies our preferences for visual clarity, art style and its coherence and of course in general provides us enjoyable experience.

References:

1. G_k. (2019, February 9). Cathedral Assets Optimizer. Nexus Mods :: Skyrim Special Edition. Retrieved March 19, 2025, from https://www.nexusmods.com/skyrimspecialedition/mods/23316
2. Halgari. (2021, July 3). 0.01K Vanilla Textures. Nexus Mods :: Skyrim Special Edition. Retrieved March 19, 2025, from https://www.nexusmods.com/skyrimspecialedition/mods/51939
3. IllusiveMan196. (2022, May 15). Excessively pixelated textures. Nexus Mods :: Skyrim Special Edition. Retrieved March 19, 2025, from https://www.nexusmods.com/skyrimspecialedition/mods/68113
4. IllusiveMan196. (2022, March 22). Texture Sorter. Nexus Mods :: Skyrim Special Edition. Retrieved March 19, 2025, from https://www.nexusmods.com/skyrimspecialedition/mods/65365
5. Kartoffels. (2020, July 29). Cleaned Skyrim SE Textures. Nexus Mods :: Skyrim Special Edition. Retrieved March 19, 2025, from https://www.nexusmods.com/skyrimspecialedition/mods/38775

6. tal3ana. (2020, March 5). Blank normal maps for all textures of SSE. Nexus Mods :: Skyrim Special Edition. Retrieved March 19, 2025, from https://www.nexusmods.com/skyrimspecialedition/mods/33353
7. tal3ana. (2019, September 22). PIXELRIM - Skyrim Ultra Reduced Textures. Nexus Mods :: Skyrim Special Edition. Retrieved March 19, 2025, from https://www.nexusmods.com/skyrimspecialedition/mods/29185

СНІГУР ВОЛОДИМИР ІГОРОВИЧ
Аспірант
Київський столичний університет імені Бориса Грінченка
ORCID ID 0000-0001-6142-8121

Синтезована графіка в розробці ігор: забезпечення якості та тестування

Анотація. Хоча зображення, створені штучним інтелектом та оброблені в інший спосіб, є відносно новою темою, їхні менш складні аналоги не є такими. Згенеровані на замовлення графічні ресурси можуть бути корисними в розробці ігор як замінники для макетування, фонова графіка для сайдскролерів або прості спрайти персонажів. Існує ще одне можливе використання їх у сфері забезпечення якості та тестування ігор, зокрема, живих плей-тестів (проходження гри реальними гравцями, а не алгоритмами). Контроль якості можуть здійснювати розробники ігор, видавці, якщо вони вирішать це зробити, або гравці, а в деяких випадках автори та користувачі модів (за умови, що гру взагалі можна модифікувати). У наступній статті розглядаються реальні, вже випущені,

практичні приклади таких модифікацій у тому вигляді, в якому вони вважаються «найбільш корисними для кінцевого користувача» і виконують своє призначення. Всі ці приклади, а також принципи, на яких вони побудовані, можуть бути корисними для розробки майбутніх ігор та інтерактивного контенту в професійному або академічному середовищі, оскільки базове припущення про бажання створити максимально якісний продукт, який є достатньо оптимізованим і може працювати на комп'ютерах з різним рівнем продуктивності, є універсальним для обох випадків.

Ключові слова: розробка ігор, згенеровані текстури, цифрове мистецтво, оптимізація, тестування.

Список джерел:

1. G_k. (2019, February 9). Cathedral Assets Optimizer. Nexus Mods :: Skyrim Special Edition. URL: https://www.nexusmods.com/skyrimspecialedition/mods/23316 (дата звернення: 19.03.2025)

2. Halgari. (2021, July 3). 0.01K Vanilla Textures. Nexus Mods :: Skyrim Special Edition. URL: https://www.nexusmods.com/skyrimspecialedition/mods/51939 (дата звернення: 19.03.2025)

3. IllusiveMan196. (2022, May 15). Excessively pixelated textures. Nexus Mods :: Skyrim Special Edition. URL: https://www.nexusmods.com/skyrimspecialedition/mods/68113 (дата звернення: 19.03.2025)

4. IllusiveMan196. (2022, March 22). Texture Sorter. Nexus Mods :: Skyrim Special Edition. URL: https://www.nexusmods.com/skyrimspecialedition/mods/65365 (дата звернення: 19.03.2025)

5. Kartoffels. (2020, July 29). Cleaned Skyrim SE Textures. Nexus Mods :: Skyrim Special Edition. URL: https://www.nexusmods.com/skyrimspecialedition/mods/38775 (дата звернення: 19.03.2025)

6. tal3ana. (2020, March 5). Blank normal maps for all textures of SSE. Nexus Mods :: Skyrim Special Edition. URL: https://www.nexusmods.com/skyrimspecialedition/mods/33353 (дата звернення: 19.03.2025)

7. tal3ana. (2019, September 22). PIXELRIM - Skyrim Ultra Reduced Textures. Nexus Mods :: Skyrim Special Edition. URL: https://www.nexusmods.com/skyrimspecialedition/mods/29185 (дата звернення: 19.03.2025)