

**Міністерство освіти і науки України
Луцький національний технічний університет**

**Пустюльга Сергій Іванович
Самчук Володимир Петрович**

ДИСКРЕТНО-ВОКСЕЛЬНИЙ ПАРАМЕТРИЧНИЙ ДИЗАЙН У MAGICAVOXEL

**НАВЧАЛЬНИЙ ДОВІДНИК
для студентів технічних ЗВО**



**Луцьк
Вежа-Друк
2026**

УДК 514.18

П-89

Рекомендовано до друку Вченою радою ЛНТУ,
протокол № 11 від 23.04.2026 р.

Затверджено науково-методичною радою ЛНТУ,
протокол № 9 від 22.04.2026 р.

Рецензенти:

О. В. Мостовенко – доктор технічних наук, професор кафедри нарисної геометрії та інженерної графіки КНУБА.

Є. В. Пугачов – доктор технічних наук, професор кафедри основ архітектурного проектування, конструювання та графіки Національного університету водного господарства та природокористування.

Н. В. Складенко – доктор мистецтвознавства, професор кафедри архітектури та дизайну ЛНТУ.

Пустюльга С. І.

П-89 Дискретно-воксельний параметричний дизайн у MagicaVoxel : навчальний довідник / С. І. Пустюльга, В. П. Самчук. – Луцьк : Вежа-Друк, 2026. – 285 с.

У навчальному довіднику авторами розглянуто основні можливості програми MagicaVoxel: від встановлення та базових інструментів до створення складних композицій і експорту в потрібний формат для подальшої роботи або рендерингу. Тут, у структурованому порядку, наведені практичні приклади, поради, а також вправи для самостійного створення різного роду воксельних моделей.

Значна увага приділена способам створення проєктів із використанням дискретно-параметричного підходу. Кожен розділ включає ілюстрації та покрокові інструкції, щоб зробити навчання легким і зрозумілим.

УДК 514.18

© Пустюльга С. І., Самчук В. П. 2026

ЗМІСТ

Вступ до воксельної графіки та огляд програми MagicaVoxel	4
1 Завантаження та встановлення MagicaVoxel.....	8
2 Інтерфейс і основні функції програми	9
3 Режими перегляду і камери у MagicaVoxel	50
4 Панелі візуалізації, рендеринг у MagicaVoxel	50
5 Експорт та інтеграція у зовнішніх програмах	62
6 Симуляція руху та анімація у MagicaVoxel.....	65
7 Основи параметричного проектування.....	72
8 Воксельне моделювання як основа дискретно- параметричного дизайну	92
9 Дискретно-воксельний параметричний дизайн у середовищі MagicaVoxel	107
Практикум.....	126
Ресурси та посилання.....	282
Список літературних джерел	283



ВСТУП ДО ВОКСЕЛЬНОЇ ГРАФІКИ ТА ОГЛЯД ПРОГРАМИ MAGICAVOXEL

Уявімо собі малювання не на аркуші паперу і навіть не на плоскому екрані, а у самому просторі. Користувач ніби ліпить зображення із кольорових пікселів, тільки не у двовимірному, а у тривимірному світі, де кожен «піксель» – це об’ємний кубик, який має товщину, висоту і глибину.

Ці маленькі кубики називаються вокселями (voxel — volume pixel, тобто «об’ємний піксель»). І саме з них можна створювати що завгодно: казкові міста, героїв фантастичних пригод, сцени з відеоігор або об’ємні зображення (рис. В.1).

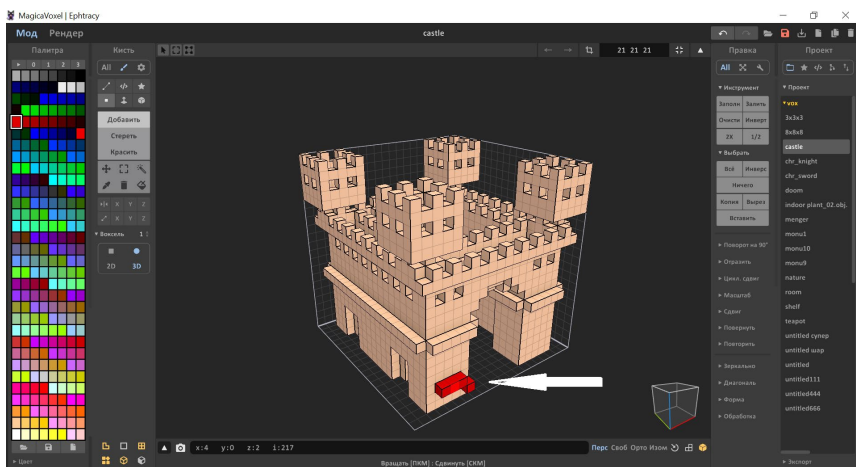


Рисунок В.1

Такий підхід до творчості кардинально змінює відчуття від цифрового малювання. Він не обмежує роботу плоскою поверхнею чи складними 3D-редакторами, де кожен крок вимагає технічних знань. Світ воксельної графіки простий, але гнучкий. Він підходить для набуття навичок 3D-моделювання і вдало поєднує креативність та технологію.

Щоб краще зрозуміти, що таке воксель і чим він особливий, варто згадати еволюцію цифрової графіки. У 1980-



90-х роках, коли комп'ютери тільки входили в наше життя, уся графіка була піксельною – це були невеликі кольорові точки, які складали зображення на екрані. Старі ігри на приставках Nintendo або перші «танчики» на комп'ютерах – класичні приклади піксель-арту.

Піксельна графіка залишилася популярною і до сьогодні, зокрема завдяки популярності ретро-стилю. Проте світ рухається вперед, і виникла потреба у створенні *тривимірної* піксельної графіки. Таким чином з'явилися вокселі.

Вперше ідея воксельного рендерингу почала застосовуватись ще у 1990-х роках у науковій візуалізації та медичній графіці. Згодом її підхопили ігрові розробники. Але справжню популярність воксельна графіка здобула завдяки таким проєктам, як Minecraft – хоча технічно це не зовсім вокселі, але візуальний стиль дуже схожий.

На цьому фоні виникла потреба у простих і зручних інструментах для створення воксельних моделей. Одним із найкращих рішень стала програма MagicaVoxel – легка, безкоштовна і дивовижно потужна для свого розміру.

MagicaVoxel – це програма для створення, редагування та рендерингу воксельних моделей. Вона була створена незалежним китайським розробником Єфремом Чжаном (псевдонім Ephtrasy) і з часом перетворилася на справжній феномен у світі цифрового мистецтва. Її люблять як новачки, так і досвідчені дизайнери – і це не випадково.

Серед численних 3D-редакторів MagicaVoxel вирізняється:

1. Легкістю входження для користувача. Не потрібно знати складну 3D-геометрію чи текстурування, щоб розпочати створення моделі. Достатньо мишки та просторової уяви.
2. Маленьким розміром. Установчий файл MagicaVoxel зазвичай займає близько 5 мегабайтів.
3. Потужними можливостями. Програма має вбудований рендер-рушій, підтримку матеріалів, світла, тіней,



а також функції експорту у формати .OBJ, .PLY, .VOX та інші.

4. Кросплатформеністю. MagicaVoxel працює на Windows і macOS без потреби в додатковому встановленні драйверів чи плагінів.

5. Активною спільнотою. Користувачі регулярно публікують власні роботи, обговорюють техніки, діляться шаблонами і навіть створюють власні моди для програми.

І головне – програма безкоштовна. MagicaVoxel – це не демонстраційна версія і не обмежене ПЗ. Це повноцінний інструмент, доступний кожному.

Цей навчальний довідник призначений для тих, хто вперше знайомиться з MagicaVoxel або хоче систематизувати свої знання. Він підходить для:

- користувачів, які хочуть опанувати основи 3D-моделювання в ігровій чи освітній сферах;
- викладачів, які шукають доступний інструмент для досліджень новацій у цифровому мистецтві чи технологіях;
- дизайнерів, які хочуть створити прототипи, піксель-арт або концепти в стилі ретро 3D;
- розробників ігор, які використовують воксельну графіку у своїх проектах;
- художників, які шукають нову форму самовираження.

Хоча існує чимало програм для 3D-моделювання, наприклад таких як Blender, Tinkercad, ZBrush, саме MagicaVoxel забезпечує унікальний баланс простоти, швидкості та креативності. MagicaVoxel – це ніби LEGO у світі 3D-графіки: зручна, модульна, зрозуміла і водночас здатна здивувати результатом моделювання.

Незважаючи на те, що MagicaVoxel є ефективним інструментом для створення тривимірних моделей у воксельній графіці, програмне середовище не передбачає наявності розвинених вбудованих механізмів параметричного чи процедурного моделювання. Функціонал програми



орієнтований передусім на ручну або напівручну побудову форм, що обмежує можливості автоматизованої генерації складних геометричних структур.

Водночас реалізація дискретно-воксельного параметричного підходу в MagicaVoxel є можливою за умови дотримання визначених правил побудови та введення системи контрольованих параметрів. Керування моделлю може здійснюватися шляхом варіювання кількості та розташування вокселів відповідно до заданих шаблонів, закономірностей або алгоритмічних принципів. Такі операції часто виконуються поза межами самої програми – із застосуванням скриптів, зовнішніх інструментів або через свідомо організоване повторення структурних модулів.

Дискретна природа воксельного простору має принципове значення для параметричного дизайну, оскільки всі змінні у цьому середовищі також набувають дискретного характеру. Це означає, що зміна параметра не спричиняє плавної неперервної трансформації форми, а веде до поетапної структурної перебудови геометрії. Такий підхід відкриває додаткові можливості для аналітичного дослідження формоутворення, контролю топологічної структури об'єктів та керування їх морфологічними характеристиками.

У навчальному довіднику авторами розглянуто основні можливості програми: від встановлення MagicaVoxel та її базових інструментів до створення складних композицій і експорту у потрібний формат для подальшої роботи або рендерингу. Тут, у структурованому порядку, наведені практичні приклади, поради, а також вправи для самостійного створення різного роду воксельних моделей.

Значна увага приділена способам створення проєктів із використанням дискретно-параметричного підходу. Кожен розділ включає ілюстрації та покрокові інструкції, щоб зробити навчання легким і зрозумілим.



1 ЗАВАНТАЖЕННЯ ТА ВСТАНОВЛЕННЯ MAGICAVOXEL

Перед початком створення перших об'єктів потрібно встановити програму на комп'ютер. Цей процес є простим і не потребує спеціальних навичок чи налаштувань.

Офіційним джерелом для завантаження MagicaVoxel є сайт <https://ephtracy.github.io/> – це сторінка автора на платформі GitHub. Слід завантажувати програму тільки з офіційного джерела, щоб уникнути шкідливого програмного забезпечення (ПЗ) (рис. 1.1).

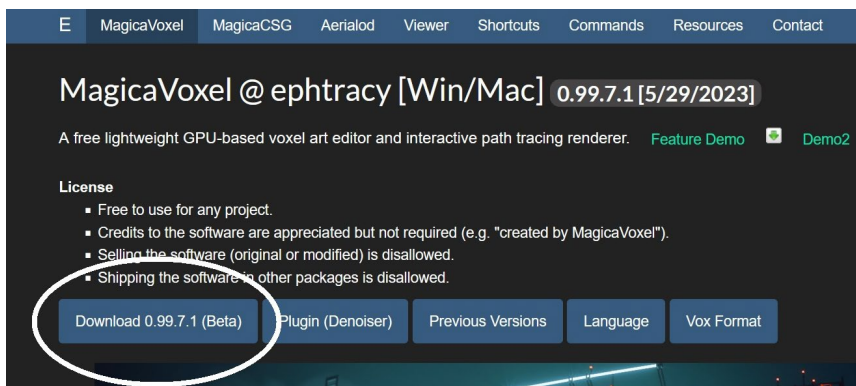


Рисунок 1.1

Для Windows завантажується архів .zip із назвою MagicaVoxel-0.99.XX-win64.zip. Для macOS – окрема версія. Програма не потребує інсталяції. Просто необхідно:

1. Завантажити архів.
2. Розпакувати у зручну теку (наприклад, C:\MagicaVoxel).
3. Запустити MagicaVoxel.exe – і програма буде готова до роботи.

У кореневому каталозі розміщено папки: vox/ – для проєктів, render/ – для рендерів, palette/ – для палітр, model/ – для зразків і головний файл – MagicaVoxel.exe.



2 ІНТЕРФЕЙС І ОСНОВНІ ФУНКЦІЇ ПРОГРАМИ

Після запуску MagicaVoxel (наприклад, версії MagicaVoxel-0.99.7.2) перед користувачем відкривається основне робоче вікно програми. Інтерфейс можна умовно поділити на шість ключових зон (рис. 2.1):

1. 3D-простір для моделювання (Scene/Viewport).
2. Панель інструментів (Tool Panel).
3. Палітра кольорів (Palette Panel).
4. Панель рендерингу (Render Panel).
5. Панель властивостей/шарів (Edit/Model/World/Render Tabs).
6. Панель проєкту (Project Panel).

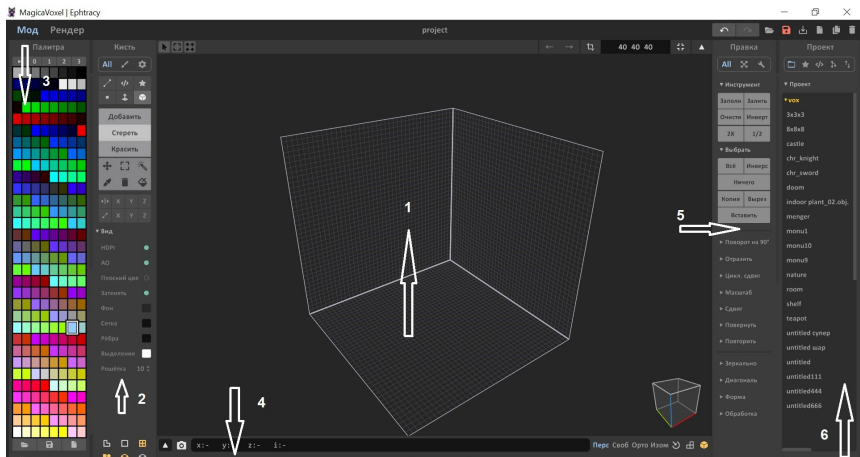


Рисунок 2.1

2.1 Робоча сцена – 3D-простір

У центрі вікна розташоване 3D-простір (сцена), на якому користувач може створювати свої моделі. Це основна частина інтерфейсу, що відображає воксельну сітку – кубічну структуру, в якій будуються об'єкти (рис. 2.2).



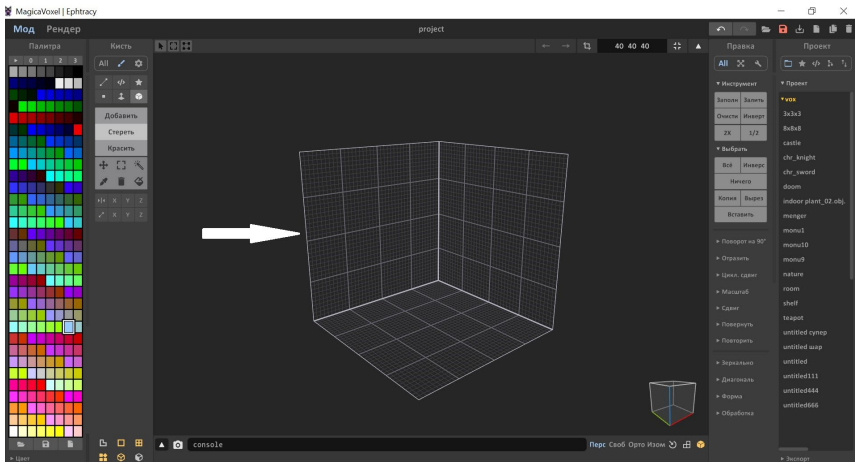


Рисунок 2.2

Цей простір підтримує:

- обертання сцени – за допомогою правої кнопки миші;
- зум – коліщатком миші;
- переміщення камери – середньою кнопкою миші (коліщатко).

Сцена має напівпрозору сітку, яка слугує орієнтиром, а також курсор (червоний воксель), що показує активну позицію для редагування.

2.2 Два режими роботи у програмі

У MagicaVoxel є два основні режими роботи, які перемикаються через вкладки «Редагування» (рис. 2.3) і «Модель» (рис. 2.4) або клавішею Tab. Один режим призначений для редагування моделі (вокселів), інший — для переміщення, масштабування й перейменування об'єкта без редагування вокселів.

«Редагування» – це головний режим, у якому відбувається створення і редагування моделі. Саме тут працюють усі інструменти з лівої панелі: додавання вокселів (Attach), стирання (Erase), фарбування (Paint), вибір кольору



(Pick), побудова форм (Box, Line, Pattern) тощо.

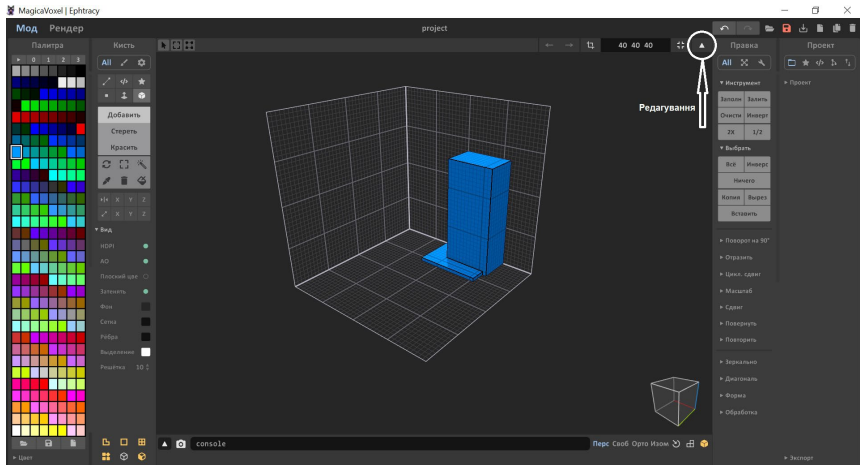


Рисунок 2.3

Цей режим активується автоматично при запуску програми і відповідає за творчу частину роботи. У цьому режимі користувач безпосередньо змінює вміст моделі – її «тіло» із вокселів. Ознаками включеного режиму є:

- у сцені видно воксельну сітку та курсор (жовтий або червоний кубик);
- активні всі інструменти лівої панелі;
- при натисканні курсора в сцені – додаються, фарбуються або видаляються кубики;
- права панель містить вкладку редагування з опціями для пензлів, виділень, трансформацій і копіювань.

Режим редагування дозволяє створювати будь-яку складну структуру – від простої коробки до архітектурного ансамблю. Якщо користувачу потрібно щось змінити у формі об'єкта, можна завжди повернутися у режим «Редагування» (Edit).

Режим «Модель» – це інший режим, який дозволяє керувати всією моделлю як єдиним блоком, але не дає можливості змінювати її вокселі. У цьому режимі користувач



не може додати чи видалити кубик – можна тільки перемістити або змінити положення моделі в сцені.

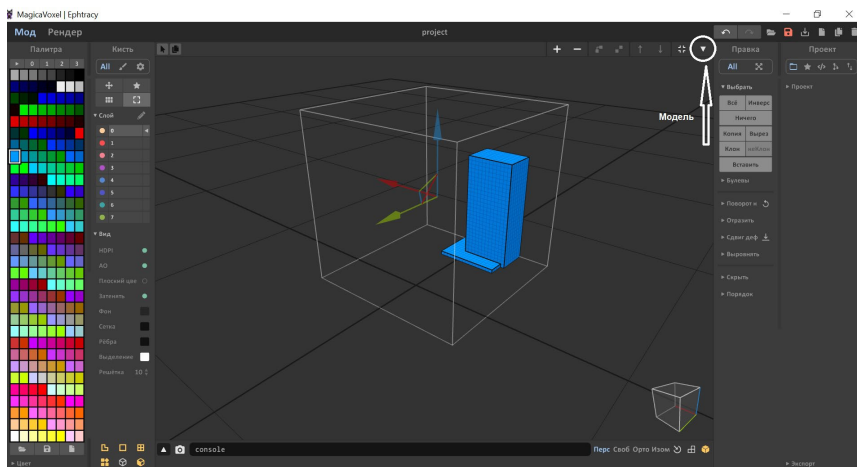


Рисунок 2.4

Цей режим особливо корисний, коли:

- користувач закінчив моделювання і хоче просто змістити об'єкт;
- йде робота з кількома об'єктами у сцені;
- потрібно змінити назву об'єкта, його центр чи положення відносно сцени.

В режимі «Модель» можна здійснювати:

- «Move» – переміщення моделі по координатних осях X, Y, Z.
- «Rotate» – обертання моделі навколо осей.
- «Scale» – масштабування (пропорційно або по осях).
- «Pivot» – вибір точки обертання.
- «Rename» – зміну назви об'єкта.
- «Center» – вирівнювання моделі по центру сцени.

У цьому режимі інструменти редагування не працюють. Курсор не з'являється. Навіть якщо натиснути в сцені – нічого не зміниться. Це створено для того, щоб уникнути випадкових змін моделі після завершення роботи над нею.



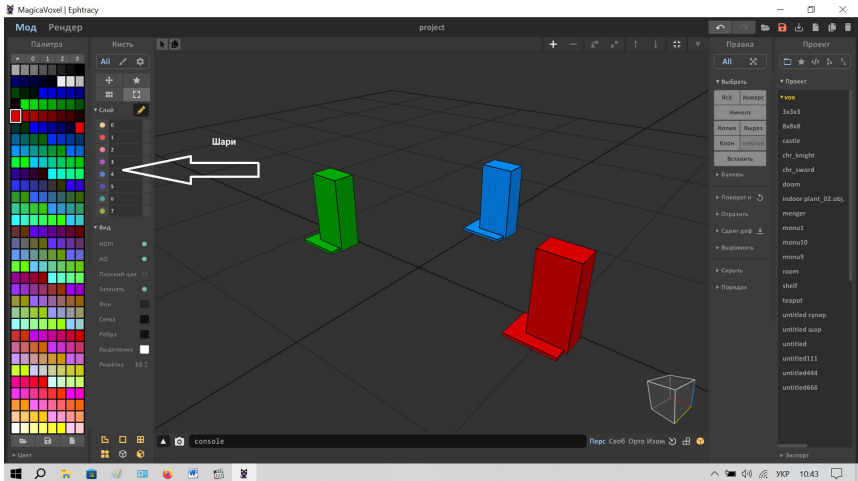


Рисунок 2.5

У режимі «Модель» існує можливість працювати не лише із однією моделлю, а з декількома об'єктами одночасно. Це досягається за допомогою системи окремих шарів, які в програмі називаються «Layers».

У MagicaVoxel шари – це окремі моделі, які можуть бути частинами великої сцени. Кожен шар містить окрему воксельну структуру та має власну позицію у просторі (рис. 2.5). Панель «Шари» розташована у лівій частині екрана (нижче секцій руху моделі). Тут відображається список усіх шарів поточної сцени – зазвичай нумерованих від 0 до 127. За замовчуванням активний лише шар 0 (тобто лише одна модель у сцені).

Що можна робити з шарами?

1. Активувати інший шар. Щоб активувати шар, слід натиснути на його номер. Усі інші в цей момент стають «замороженими».
2. Приховати/показати шар. Кнопка у вигляді кола дозволяє тимчасово вимкнути відображення певного шару. Це зручно, якщо працювати із великою сценою.
3. Створити новий шар. Кнопка «+» додає новий шар



(нову порожню модель). Його можна використати для нової частини сцени (наприклад, дах, двері, дерево).

4. Клонувати шар. Кнопка копіювання створює повний дублікат активного шару – з усіма вокселями та властивостями.

5. Видалити шар. Кнопка «-» видаляє вибраний шар без можливості відновлення, тож бажано зберігати резервні копії проєкту.

6. Переміщення шару у просторі. Для кожного шару можна встановити позицію (X, Y, Z), масштаб і обертання – це налаштовується у режимі «Модель».

Слід знати, що:

- кількість шарів обмежена – у версії MagicaVoxel 0.99.6.4 максимально доступно 128 шарів;

- один шар – це одна воксельна модель. Усі вони розміщуються у спільному просторі сцени, але не «зливаються» між собою;

- редагування можливе лише для активного шару;

- у кожного шару може бути власне ім'я (задається у вкладці «Модель»).

2.3 Панель інструментів в режимі «Редагування» (ліворуч від 3D-графічного простору)

Панель інструментів у MagicaVoxel розташована ліворуч від головного вікна і є джерелом усіх дій, які виконуються над моделлю (рис. 2.6).

Кожна кнопка на цій панелі представляє певну функцію або інструмент, що дозволяє створювати, змінювати або видаляти вокселі. Наприклад, найпростіший інструмент – це «Attach», тобто «Додати» – дозволяє додати нові кубики у сцену. Після вибору кольору з палітри та натискання на сцену воксель з'являється у вказаній позиції. Це наче гра у цифровий конструктор.

Головні інструменти (режим «Редагування») (рис. 2.6).



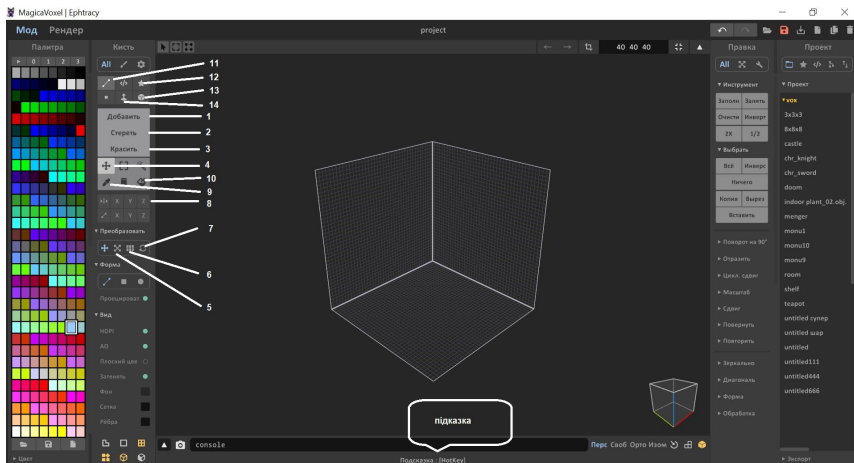


Рисунок 2.6

- 1 – «Attach» – Додати (T) – додавання вокселів.
- 2 – «Erase» – Стерти (R) – видалення вокселів.
- 3 – «Paint» – Фарбувати (G) – фарбування.
- 4 – «Move 1» – переміщення області.
- 5 – «Move 2» – масштабування області.
- 6 – «Move 3» – створення масиву області.
- 7 – «Move 4» – обертання області.
- 8 – «Flip» – віддзеркалення.
- 9 – «Pick» – вибір кольору з моделі.
- 10 – «Fill» – заливка.
- 11 – «Line» – малювання ліній, прямокутників, кіл.
- 12 – «Pattern» – повторювані візерунки.
- 13 – «Box» – створення прямокутних об’ємів.
- 14 – «Face» – малювання або видалення площин.

Інструмент «Erase» – це ластик. Можна швидко видалити зайві кубики, просто натиснувши на них. А інструмент «Paint» дає змогу перефарбовувати вже існуючі об’єкти в сцені, не змінюючи їхньої форми. Таким чином, можна зберегти структуру, але змінити кольорову гаму.



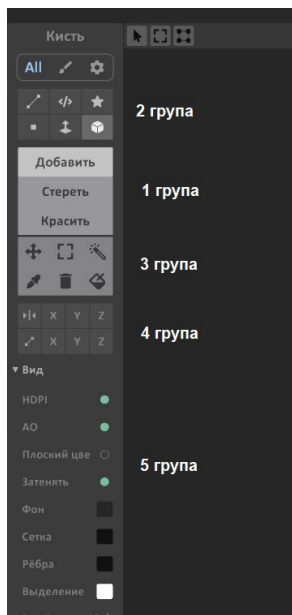


Рисунок 2.7

Є також інструмент «Move» – за його допомогою можна переміщати частини сцени або всю модель повністю. Інструмент «Rotate» повертає вибрані об’єкти, а «Scale» дозволяє збільшити або зменшити масштаб елементів. Ці дії особливо важливі, коли створюються складні об’єкти або змінюється розташування деталей без втрати пропорцій.

Кожна кнопка панелі інструментів має піктограму. При наведенні на неї – внизу з’являється підказка.

Кнопки згруповані за призначенням (рис. 2.7) і мають колірне виділення. Вибраний інструмент підсвічується.

2.4 Інструменти геометрії (пряма лінія, прямокутник, коло) – рис. 2.8

Інструмент працює в режимі «Додати» і має три геометричні форми: пряма лінія, прямокутник і коло.

«Лінія» – інструмент для створення прямих ліній.



Інструмент зручний для побудови витягнутих об'єктів. Після вибору інструмента необхідно:

1. Натиснути у початковій точці.
2. Провести до кінцевої точки.
3. Відпустити кнопку миші.

MagicaVoxel автоматично з'єднає ці дві точки воксельною лінією. Для кращого контролю рекомендується працювати в ортографічному режимі перегляду: «Ortho».

«Прямокутник» – інструмент для створення прямокутників, паралельних основним площинам проєкцій.
«Коло» – інструмент для створення кіл (рис. 2.8).

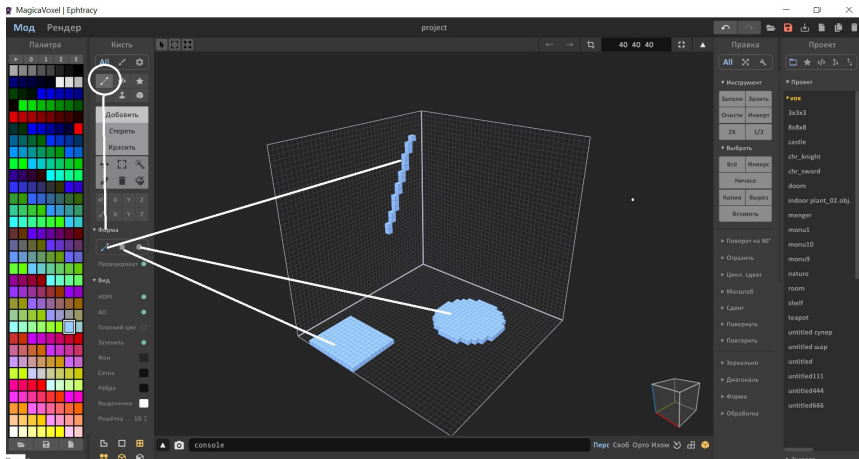


Рисунок 2.8

2.5 Інструменти вокселів (ламана, плоска фігура, воксель керованого розміру) – рис. 2.9

«Воксель у режимі» «Додати» (рис. 2.9) – це найпростіший спосіб додати один воксель. Цей режим активується через кнопку «Воксель». Після цього кожне натискання лівої кнопки миші додасть один окремий кубик. Це зручно, коли користувач працює із невеликими деталями в картині.



Для того щоб побачити прив'язку кубика до координатної сітки, краще увімкнути сітку через: інструмент «Сітка». Так буде простіше точно позиціонувати окремі воксели.

Цей інструмент включає можливості коригування розміру вокселя в картині та його геометричної конфігурації.

«Воксель у режимі» «Стерти» або «Фарбувати» відповідно працює на видалення вокселів з моделі або їх фарбування.

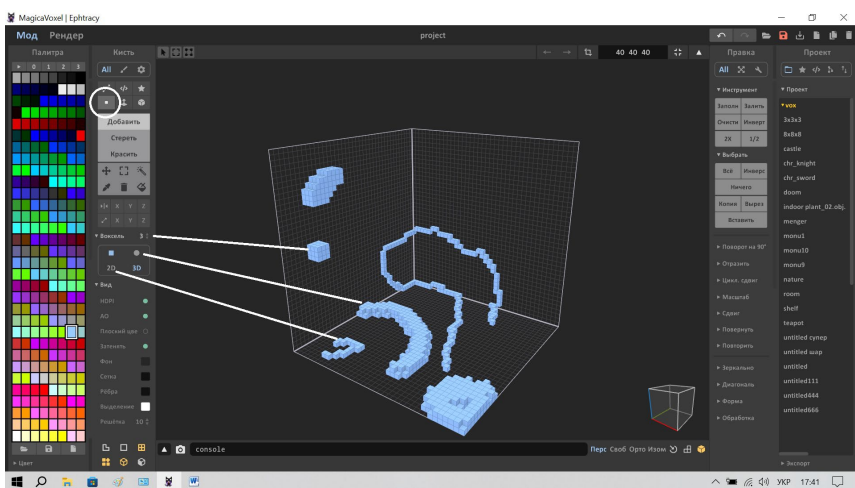


Рисунок 2.9

2.6 Інструменти редагування граней (втягування, геометрія, колір) – рис. 2.10

«Додати» + «Face» – це режим, який дозволяє додавати кубики лише на видиму сторону об'єкта, не заходячи вглиб моделі. Наприклад, якщо потрібно надбудувати фасад, дах або плитку, не змінюючи внутрішню структуру самого об'єкта – інструмент працюватиме оптимально.

У такому режимі MagicaVoxel сам обирає тільки ті сторони, які доступні для редагування, – тобто ті поверхні, які



бачить камера.

Це може економити багато часу при створенні об’єкта, оскільки не потрібно обертати модель, щоб дістатися до потрібної області чи грані вручну.

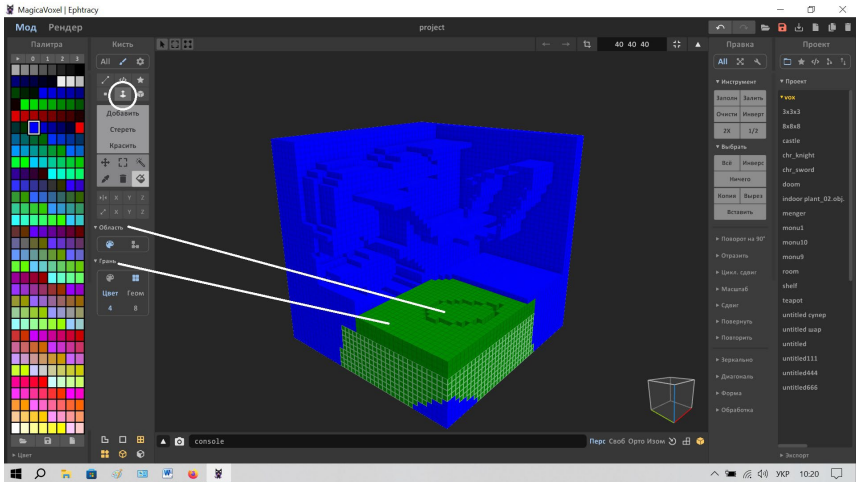


Рисунок 2.10

2.7 Інструменти створення патернів (геометрія, колір) – рис. 2.11

«Додати» + «Патерн» – це особливий режим, за допомогою якого можна створювати повторювані структури або декоративні елементи.

Після активації цього режиму можна малювати періодичні лінії, решітки, цегляні кладки тощо.

Працює це так:

- задається будь-який об’єкт у панелі властивостей (наприклад, дискретна модель кривої);
- далі слід просто рухати пензлем по сцені – і програма автоматично накладає обраний шаблон (рис. 2.11).

Цей режим ідеально підходить для плитки, панелей, фасадів, мостових, де потрібно багато повторюваних блоків,



але немає часу класти кожен блок вручну.

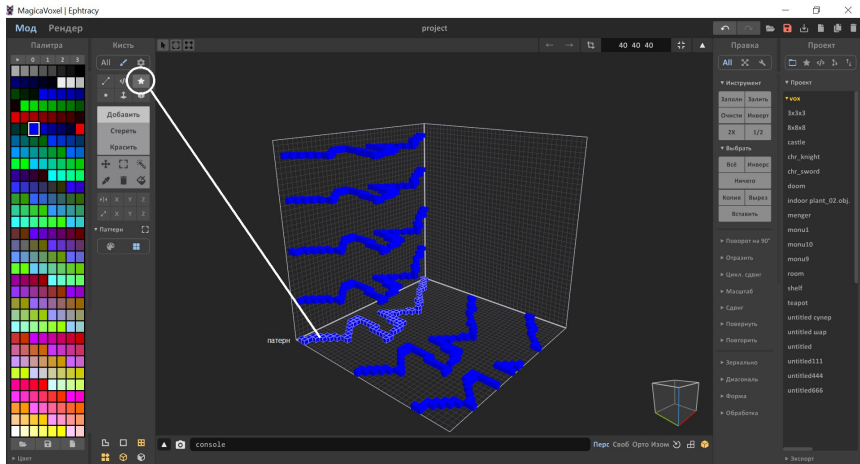


Рисунок 2.11

2.8 Інструменти редагування кубоїдів (геометрія, колір) – рис. 2.12

Один з найзручніших інструментів режиму «Додати» – є (Вох) інструмент «Коробка». Його можна обрати через меню вгорі на панелі інструментів (рис. 2.12).

У цьому режимі користувач не просто клацає один воксель за раз, а можете натисканням і розтягуванням створювати одразу об'ємну фігуру – коробку.

Це дуже зручно для побудови основи будівель, стін, прямокутних об'єктів. Наприклад, щоб створити стіну будинку, достатньо один раз натиснути і провести курсором у двох напрямках – і ви отримаєте блок із множини вокселів одразу (рис. 2.12).

Пензлик-коробка працює у поєднанні з палітрою кольорів: перед тим як класти кубики, слід переконатися, що обрано потрібний колір у палітрі праворуч.



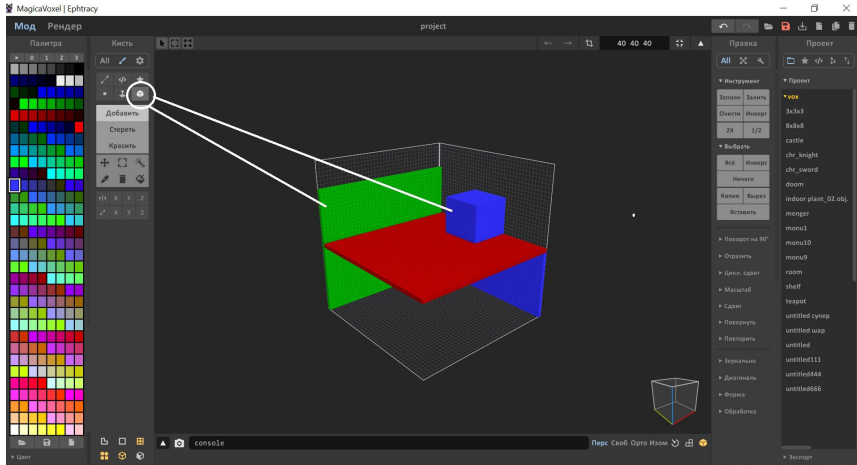


Рисунок 2.12

2.9 Режим редагування моделі (переміщення, масштабування, створення масивів, обертання) – рис. 2.13

Будь-який об'єкт або його частину можна редагувати, переміщуючи його у просторі. Для цього у MagicaVoxel є функції руху:

- «Переміщення» – активує рух або сцени, або виділеної частини об'єкта по осях X, Y, Z (достатньо потягнути його мишкою).
- «Масштабування» області – змінює розміри області (корисно для декоративних об'єктів).
- Створення «Масиву» області – створює ідентичні копії, які можна змінювати.
- «Обертання» області – обертає об'єкт навколо осей.

Для точності відтворення рухів слід увімкнути сітку сцени та ортографічний вид.



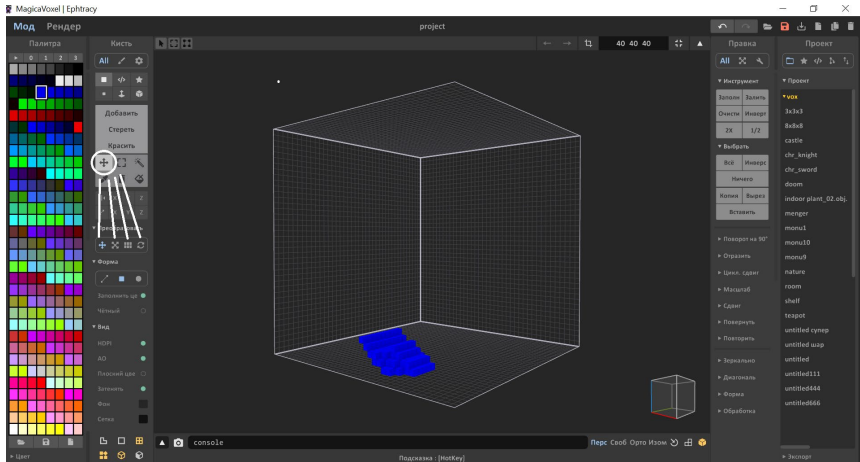


Рисунок 2.13

2.10 Режим виділення (рамкою, пензлем, кубідом) – рис. 2.14

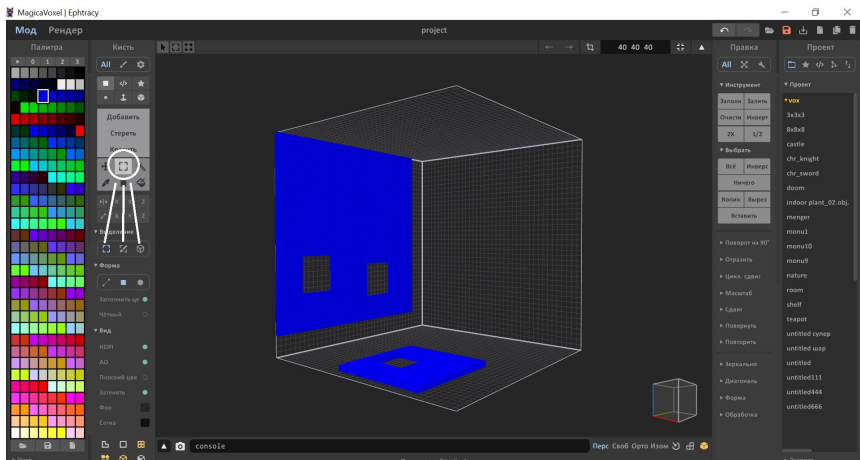


Рисунок 2.14

Режим виділення (області або грані) – рис. 2.15.



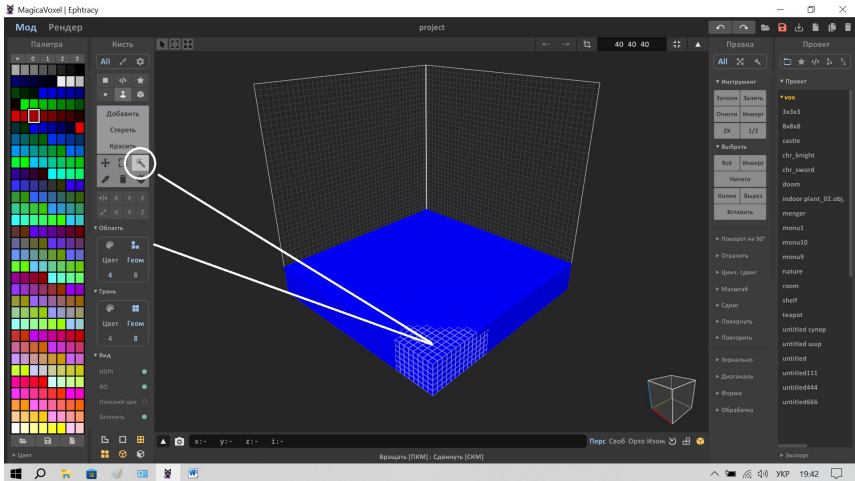


Рисунок 2.15

2.11 Палітра кольорів

Палітра кольорів – одна із ключових частин інтерфейсу MagicaVoxel. Палітра містить до 256 кольорів. Вона розташована зліва і виглядає як сітка з кольорових квадратиків (рис. 2.16). Це ті кольори, якими користувач може розфарбовувати свої моделі. Щоб обрати колір, достатньо просто клікнути по ньому – і він стане активним.

Щоб змінити колір, слід натиснути лівою кнопкою миші внизу палітри функцію «Колір» – відкриється редактор кольорів, де користувач може змінити відтінок, насиченість і яскравість. Для складних проектів часто корисно зберігати власні палітри або імпортувати готові, щоб забезпечити однаковість стилю у різних сценах.

MagicaVoxel також підтримує вибір кольору за допомогою клавіші Alt – слід натиснути її і клацнути на будь-який воксель у сцені, щоб автоматично вибрати його колір у палітрі. Це дуже зручно, коли користувач хоче швидко підібрати вже використаний колір без пошуку вручну.



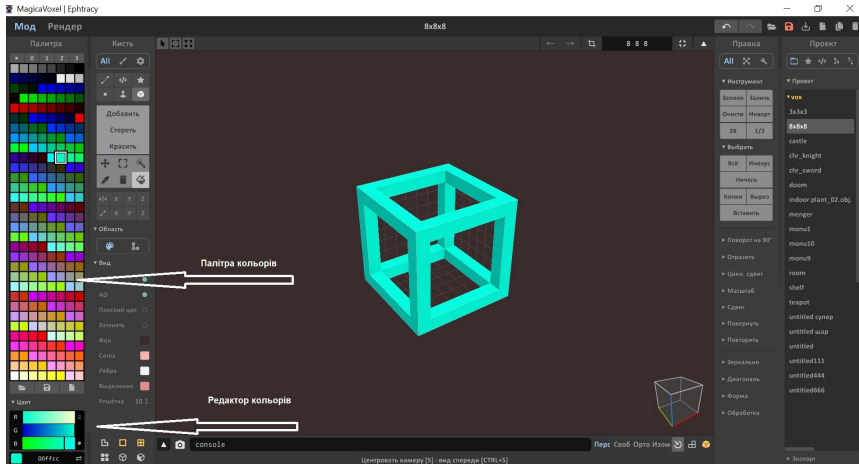


Рисунок 2.16

2.12 Панель «Вид» (View) – налаштування вигляду сцени (рис. 2.17)

Панель «Вид» (View) у MagicaVoxel дозволяє змінювати зовнішній вигляд сцени під час роботи (рис. 2.18). Вона розташована справа від палітри кольорів і знизу під панеллю інструментів. Вона не редагує саму модель, а лише впливає на те, як саме її бачить користувач програми.

Правильне налаштування вигляду допомагає краще оцінити деталі, співвідношення об'єктів, кольори й тіні, а також забезпечує комфорт для очей під час тривалої роботи.

Інструменти панелі «Вид» (рис. 2.18) включають кілька груп параметрів:

HDPI:

- вмикає або вимикає високу чіткість відображення інтерфейсу;
- коли увімкнено (зелена крапка), елементи сцени та інтерфейсу виглядають згладженими, особливо на моніторах із високою роздільністю;
- якщо у вас спостерігається затримка у роботі



програми, HDPI можна вимкнути для збільшення швидкодії.

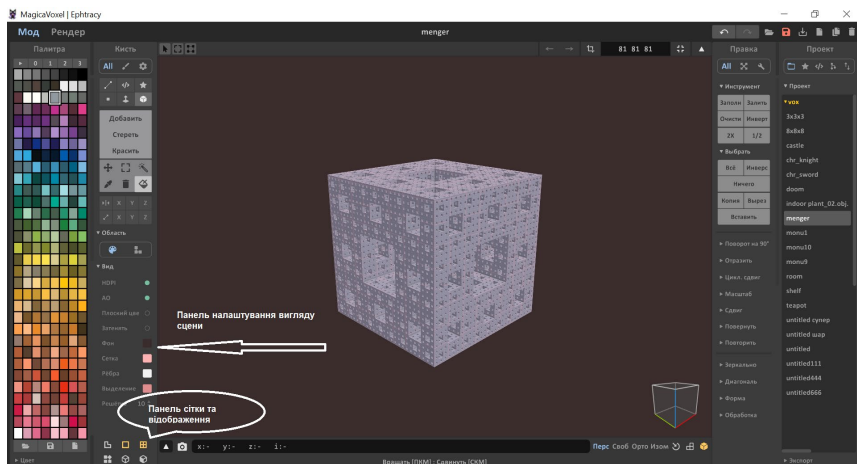


Рисунок 2.17

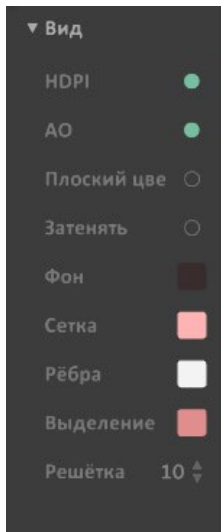


Рисунок 2.18

AO (Ambient Occlusion):

– увімкнення/вимкнення оклюзії навколишнього освітлення;



– додає легкі тіні у кутах і зонах зіткнення об'єктів – це створює візуальну глибину;

– увімкнено (зелена крапка) – модель виглядає реалістичніше навіть у режимі перегляду.

Плоский колір (Flat Color):

– коли увімкнено, відображає кольори без освітлення і тіней – як у піксельному редакторі;

– усі вокселі виглядають рівно і чисто, без затемнень;

– цей режим зручно вмикати, коли потрібно бачити справжній колір вокселя без спотворень освітлення.

Затіняти (Shade):

– дає змогу увімкнути чи вимкнути затінення поверхонь (тобто роботу базового освітлення);

– коли «Shade» вимкнено – сцена виглядає плоскою;

– «Shade» + «АО» разом створюють приємне м'яке освітлення для огляду моделей.

Фон (Background):

– визначає колір заднього фону сцени;

– натискання на квадратик відкриває палітру, де можна обрати будь-який колір;

– якщо модель погано контрастує з фоном – можна змінити фон на темний або світлий для кращого перегляду.

Сітка (Grid):

– колір координатної сітки, яка відображається на сцені;

– допомагає орієнтуватися у просторі, особливо при точному моделюванні.

Ребра (Edges):

– колір контурів вокселів – ті самі «шви» між кубиками;

– їх можна зробити білими, сірими або темними для візуального стилю «піксельної сітки».

Виділення (Selection):

– колір, яким буде виділена активна область або



виділені вокселі;

– виділення відображається при роботі з інструментом «Select».

Решітка (Grid Size):

– встановлює розмір модульної сітки (за замовчуванням – 10);

– сітка допомагає в позиціонуванні об'єктів та точному розміщенні елементів;

– значення можна збільшити або зменшити залежно від розміру сцени чи масштабу об'єкта.

Поради із використання налаштувань вигляду:

1. Якщо працюєте з яскравими об'єктами – слід використовувати темний фон.

2. Якщо сцена виглядає «занадто плоскою» – спробуйте увімкнути «АО» та «Shade».

3. Для точного редагування вмикайте «Grid» та встановлюйте відповідну «Grid Size».

4. Якщо кольори нечіткі – вимкніть «Shade» і «АО», щоб побачити справжній вигляд палітри.

2.13 Панель сітки та режимів відображення – керування виглядом моделі (рис. 2.19)

У нижній частині вікна MagicaVoxel знаходиться панель сітки та візуалізації об'єкта – набір із 6 кнопок, які дозволяють перемикає спосіб показу моделі та її структури на сцені.

Вони не змінюють саму модель, а лише впливають на те, як вона виглядає під час редагування. Це зручно, коли потрібно перевірити внутрішню структуру моделі, вирівняти об'єкти, побачити межі сітки або перемикає між «суцільним» і «каркасним» переглядом.



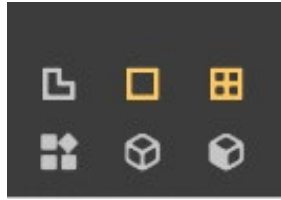


Рисунок 2.19

Верхній ряд (3 кнопки – рис. 2.19) – режими показу сітки сцени:

1. Кутова сітка (Corner Grid). Відображає лише кути сітки або її основні межі. Дає загальне уявлення про розміщення об'єкта у сцені. Це найменш нав'язливий варіант – добре підходить для візуалізації великих моделей.

2. Повна сітка (Full Grid) – (активна на зображенні). Показує повну прямокутну сітку, у якій видно кожен клітинку (воксельний блок). Корисно, коли потрібно чітко позиціонувати кубики вручну. Рекомендується при роботі з дрібними деталями.

3. Коміркова сітка (Dense Grid) – (сітка поділена на квадрати). Показує дрібні елементи всередині кожного блоку – особливо при масштабуванні. Дає можливість оцінити геометричну точність. Ідеально підходить для технічних моделей або при симетричному проектуванні.

Нижній ряд (3 кнопки – рис. 2.19) – режими візуалізації об'єкта:

1. Каркасний режим (Wireframe). Відображає модель як кубічний каркас, показуючи лише контури кожного вокселя. Внутрішня частина сцени стає прозорою. Допомогає бачити, що всередині моделі, не розбираючи її. Корисно при роботі з багатошаровими або порожнистими об'єктами.

2. Напівпрозорий режим (Shaded Cubes) – напівпрозорий куб. Відображає модель із м'якими краями та прозорими ребрами. Створює ефект «об'ємної сітки» – видно форму та частково вміст. Дає відчуття «легкості» та структури моделі одночасно.



2.14.1 Блок «Інструмент» – комплексні дії зі сценою

Цей блок (рис. 2.21, а) містить основні команди для глобального впливу на всю сцену або активну область. Вони дають змогу масово змінювати колір, масштаб чи структуру вокселів.

- «Заповнити»: заповнює весь простір сцени активним кольором. Це буде корисним, якщо потрібно створити суцільний фон або кубічну базу для моделі.

- «Залити»: перефарбовує всі вже заповнені вокселі у вибраній колір. Структура залишається незмінною, змінюється тільки палітра.

- «Очисти»: видаляє всі вокселі зі сцени. Ця дія необоротна (варто зберігати резервну копію).

- «Інверт»: перетворює порожній простір на заповнений, а заповнений – на порожній. Це створює ніби «негатив» моделі.

- «2X» / «1/2»: масштабування всієї моделі вдвічі більше або вдвічі менше. Зберігається форма, але змінюється розмір кожного вокселя. Корисно для переходу між дрібними та великими сценами.



Рисунок 2.21

2.14.2 Блок «Вибрати» – керування виділенням

Даний блок (рис. 2.21б) дозволяє працювати з виділенням: обирати, інвертувати, копіювати або вставляти області сцени.



- «Все»: виділити всі вокселі у шарі чи поточному кадрі.
- «Інверс»: інвертувати виділення – вибрані вокселі стають невибраними, і навпаки.
- «Нічого»: зняти будь-яке поточне виділення.
- «Копія»: копіювати обрану область у буфер обміну.
- «Виріз»: копіювати і видалити область одночасно.
- «Вставити»: вставити із буфера копійовану область.

Можна вставити кілька разів у різні частини сцени.

Ці дії використовуються разом з інструментом «Select» на лівій панелі. Наприклад, можна виділити область рамкою, вирізати її та вставити як окремих елемент.

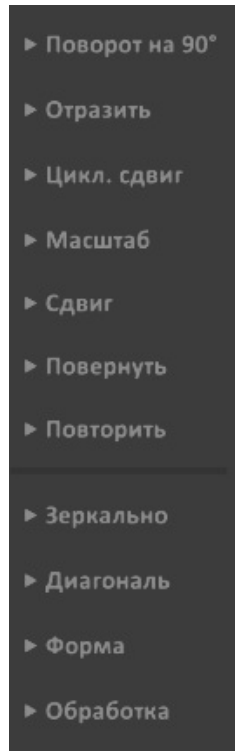


Рисунок 2.22



2.14.3 Блок «Поворот на 90°» – швидке обертання

Цей блок (рис. 2.22) розкриває параметри повороту обраної області на 90 градусів по осях X, Y або Z. У більшості випадків це використовується для швидкого обертання об'єктів, наприклад:

- повернути вікно в стіні;
- розвернути дах будинку;
- орієнтувати меблі.

Кожен напрямок обертання можна задавати окремо, що дозволяє гнучко керувати геометрією моделі.

2.14.4 Блок «Відобразити» – дзеркальне відображення

Цей блок (рис. 2.22) дозволяє віддзеркалити вибрану частину моделі по осі X, Y або Z. Це дуже корисно при створенні симетричних моделей:

- дві однакові частини будівлі;
- симетричне обличчя персонажа;
- дзеркальне розміщення декору.

Дія застосовується до виділеної області або до всієї сцени, якщо нічого не виділено.

2.14.5 Блок «Цикл. зсув» – циклічний зсув

Цей інструмент (рис. 2.22) пересуває воксели в певному напрямку, при цьому все, що виходить за межі, повертається з іншого боку (ефект обгортання).

Наприклад:

- модель зсувається праворуч;
- крайні воксели з'являються зліва.

Це дуже корисно для створення плиткових текстур, швів, безшовних візерунків або рухомих деталей.

2.14.6 Блок «Масштаб» — зміна розмірів об'єкта

Інструмент «Масштаб» (рис. 2.22) дозволяє змінювати розміри виділеної області моделі по одній або кількох координатних осях (X, Y, Z). На відміну від глобального



масштабування ($2X / 1/2$), цей інструмент працює лише з вибраною частиною сцени. Можна:

- збільшити об'єкт у ширину, зберігаючи висоту;
- зменшити модель у висоту, зберігаючи форму основи;
- розтягнути чи стиснути воксельний фрагмент у потрібному напрямку.

Це дає змогу швидко створити новий варіант моделі, не переробляючи її повністю. Наприклад: в будинку є прямокутне вікно, і ви хочете зробити його вищим – достатньо виділити і масштабувати лише по осі Y.

2.14.7 Блок «Зсув» – пряме зміщення області

«Зсув» – це інструмент для переміщення виділеної області у просторі без копіювання (рис. 2.22). Він дозволяє зсунути групу вокселів у будь-якому напрямку:

- вгору або вниз;
- вперед або назад;
- вліво або вправо.

Це корисно для точного розташування частин моделі. На відміну від трансформацій у режимі «Model», цей зсув не рухає всю модель, а лише вміст виділення. Наприклад, можна виділити двері в будинку й пересунути їх трохи правіше, не змінюючи решту сцени.

2.14.8 Блок «Повернути» – обертання на довільний кут

Цей інструмент (рис. 2.22) дає змогу обертати об'єкт не лише на 90° , а на будь-який вказаний кут (наприклад, 30° , 45° , 120°). Обертання можна здійснити навколо осей X, Y або Z.

Інструмент підтримує більш гнучку трансформацію і дозволяє створювати діагональні або нестандартні форми.

У разі повороту на неградуйований кут (наприклад, 17°) можливе незначне спотворення, оскільки вокселі – це цілі кубики.



2.14.9 Блок «Повторити» – повтор останньої дії

«Повторити» виконує останню застосовану операцію ще раз (рис. 2.22). Наприклад, якщо користувач щойно застосував інструмент «Масштаб» або «Зсув», натискання кнопки «Повторити» виконає цю дію повторно.

Це економить час при серійному редагуванні, коли одна й та сама дія потрібна кілька разів поспіль. Наприклад, якщо здійснюється копіювання частини огорожі й кілька разів вона зсувається вправо – достатньо один раз виконати дію, а потім просто натискати «Повторити».

2.14.10 Блок «Дзеркально» – точне керування дзеркальністю

Цей блок (рис. 2.22) надає детальніші параметри для дзеркального редагування. На відміну від простого «Відобразити», тут можна задати вісь симетрії, її напрямок і спосіб віддзеркалення:

- за позицією центру сцени;
- за краєм області;
- по відношенню до об'єкта або всього шару.

Це важливо, коли потрібно досягнути ідеальної симетрії для складної фігури (наприклад, крил, архітектурних деталей тощо).

2.14.11 Блок «Діагональ» – діагональні перетворення

Цей інструмент (рис. 2.22) дозволяє виконувати скошування, зрізи або перетини під кутом. Ця функція використовується не часто, але вона корисна для створення дахів, клиноподібних об'єктів, криволінійних форм.

Можна задати напрямок та силу діагонального зсуву, що формує унікальні форми з вокселів. Наприклад, використати цей інструмент для створення нахилу покрівлі або сходів.

2.14.12 Блок «Форма» – зміна геометрії області

Блок «Форма» (рис. 2.22) містить дії, які впливають на



геометричну структуру виділення:

- обрізання країв;
- вирівнювання поверхонь;
- округлення форм;
- згладжування нерівностей.

Це дозволяє переходити від «грубих» блокових форм до м'якших або рівномірних структур, що виглядають природніше.

2.14.13 Блок «Обробка» – очищення та фільтри (рис. 2.22)

Останній блок – це набір допоміжних фільтрів і функцій обробки сцени:

- видалення порожніх або зайвих вокселів;
- автоматичне вирівнювання висоти;
- злиття кольорів;
- сортування шарів або кадрів.

Ці функції часто використовуються на фінальному етапі роботи – для підготовки моделі до рендеру чи експорту.

Взагалі панель «Правка» – це інструментарій професійного рівня, що відкриває безліч можливостей навіть для простих сцен. Опанування цієї панелі дозволяє працювати з MagicaVoxel значно швидше, гнучкіше та точніше.

2.15 Панель «Консоль» – керування камерою, виглядами і рендером (рис. 2.23)

Панель «Консоль» (Console) – один із потужних, але маловідомих інструментів MagicaVoxel, який дозволяє керувати параметрами камери, перегляду, положенням сцени, а також фіксувати вигляди для візуалізації (рендерів).

Цей блок (рис. 2.23) зазвичай розташований у нижній частині вікна, і він дає точний контроль над тим, як користувач бачить свою модель. Панель «Console» – це набір параметрів для налаштування камери вручну, вибору типу проєкції,



фіксації положення моделі у просторі та створення унікальних видів сцени. Вона особливо корисна на етапі рендерингу, коли потрібно досягти стабільного, професійного ракурсу.

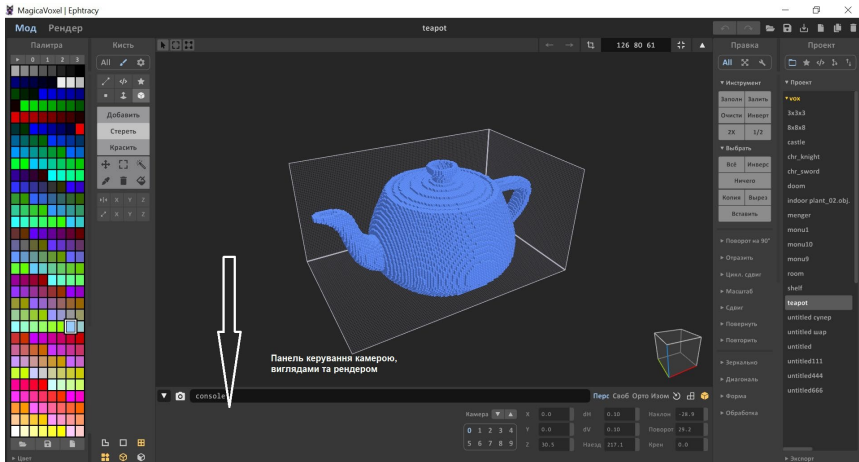


Рисунок 2.23

У лівій частині (рис. 2.24) – поле, де можна вводити текстові команди. В основному воно використовується розробниками. Для новачка достатньо знати, що це поле можна залишити порожнім – більшість параметрів можна налаштувати праворуч.

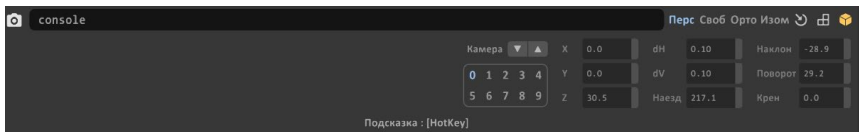


Рисунок 2.24

Камера (блок керування). Камера: [0–9]

Дає змогу вибрати один із десяти слотів камери (від 0 до 9). Кожен слот – це позиція й орієнтація камери, яку можна зберегти або викликати. Наприклад, у слоті 0 можна зберегти вигляд зверху, а в 1 – вигляд під кутом.

Якщо створюється кілька знімків однієї сцени з різних



кутів – це незамінний інструмент.

X / Y / Z

- положення камери у просторі;
- дозволяє вручну виставити координати положення;
- зручно для фіксованих сцен або точного повторення позиції.

dH / dV (delta Horizontal / delta Vertical)

Це чутливість руху камери при ручному обертанні:

- dH – горизонтальна швидкість;
- dV – вертикальна.

Якщо сцена надто швидко обертається – слід зменшити значення. Якщо повільно – збільшити.

Нахил (Tilt)

Відповідає за нахил камери вгору або вниз.

Дозволяє подивитися на об'єкт не прямо, а під кутом.

Поворот (Yaw)

Поворот камери вліво або вправо навколо вертикальної осі. Це найчастіше використовуваний параметр – саме він змінює «кут погляду» по горизонталі.

Крен (Roll)

Обертання камери навколо власної осі (як у літака).

Застосовується для кінематографічних ефектів, коли зображення трохи «нахилене».

Наїзд (Zoom / Distance)

Це відстань від камери до сцени:

- збільшуючи значення – користувач віддаляє камеру;
- зменшуючи – наближає.

У режимі ортографії числове значення працює як масштаб.



Тип проєкції (кнопки праворуч угорі (рис. 2.24)):

- «Перспектива» – стандартна 3D-проєкція, з глибиною і спотворенням масштабу;
- «Вільна» – камера без обмежень, для довільного руху;
- «Ортографічна» – плоска проєкція, без перспективи. Проєкція добре підходить для технічних або стилізованих знімків;
- «Ізометрія» – жорстко фіксована проєкція, де кути між осями рівні. Ідеально підходить для рендерів у стилі ігор.

Інші кнопки (рис. 2.24, лівий край):

- іконка камери – створення скріншоту або знімка сцени;
- «Стрілка» – меню з додатковими налаштуваннями або вибір параметрів.

Практичні поради для користувачів:

- для знімків із однакових кутів – використовуйте слоти камери (0–9);
- для коміксів або покрокової анімації – зберігайте кожен кадр з однаковою камерою;
- використовуйте «Орто» або «Ізом», якщо ваша модель має чітку геометрію (будівлі, техніка);
- високі значення крену створюють ефект нестабільної камери – використовуйте обережно.

Практична вправа: Створення сцени із чотирма камерами (ракурсами). Кроки:**1. Встановлення першого ракурсу**

Наведи камеру на сцену з вигляду спереду. Відрегулюй масштаб (поле «Наїзд»), щоб об'єкт було добре видно. У полі Камера клацни на 0 – вибрано слот 0. Натисни іконку дискетки або комбінацію клавіш Ctrl+0 – камера збережена. Дана позиція збережена в слоті 0. Її можна викликати в будь-який



момент.

2. Створення другого ракурсу – зліва

Повернути сцену так, щоб об'єкт було видно зліва. Обрати слот 1 у полі Камера. Натиснути Ctrl+1 – ракурс збережено.

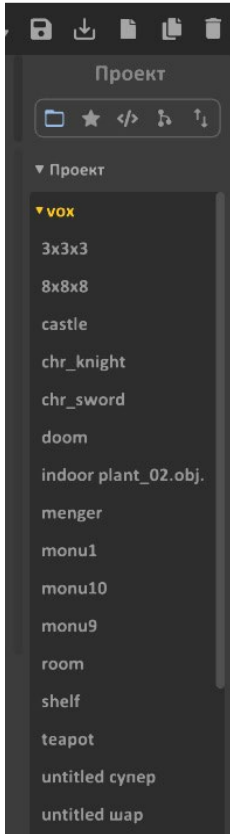


Рисунок 2.25

Вгорі екрану розташована панель проекту (рис. 2.25), яка дозволяє створювати нові сцени, відкривати існуючі, зберігати моделі та експортувати їх у потрібних форматах. Тут також можна дати назву своєму файлу, скопіювати модель, очистити сцену або навіть змінити розмір 3D-середовища, в якому працює

3. Створення третього ракурсу – вигляд зверху

Перетягнути камеру так, щоб дивитися згори вниз. Обрати слот 2. Натиснути Ctrl + 2 – збережено.

4. Створення ізометричного вигляду

В полі проєкції вибрати режим «Ізом». Повернути сцену під кутом 45° (поворот $\approx -135^\circ$, нахил $\approx -30^\circ$). Обрати слот 3. Натиснути Ctrl+3.

Результат: Отримаємо 4 чітко зафіксовані камери, до яких можна повертатися у будь-який момент. Це дуже зручно для:

- презентацій;
- створення коміксів;
- анімації;
- порівняння змін у моделі.

2.16 Панель «Проект» – управління файлами, моделями та сценами (рис. 2.25)



користувач.

Коли користувач натискає кнопку «New», створюється чиста сцена з порожнім простором. Кнопка «Open» відкриває вікно, де можна обрати раніше збережений файл у форматі .vox. Щоб не втратити свої моделі, варто регулярно натискати «Save» – це збереже ваші зміни у файл із такою ж назвою, яка вказана.

Формат .vox є рідним для MagicaVoxel – у ньому зберігається вся інформація про кольори, форми та позиції вокселів. Однак пізніше користувач зможе експортувати моделі у інші формати. Про це буде йти мова в розділах, присвячених візуалізації та експорту.

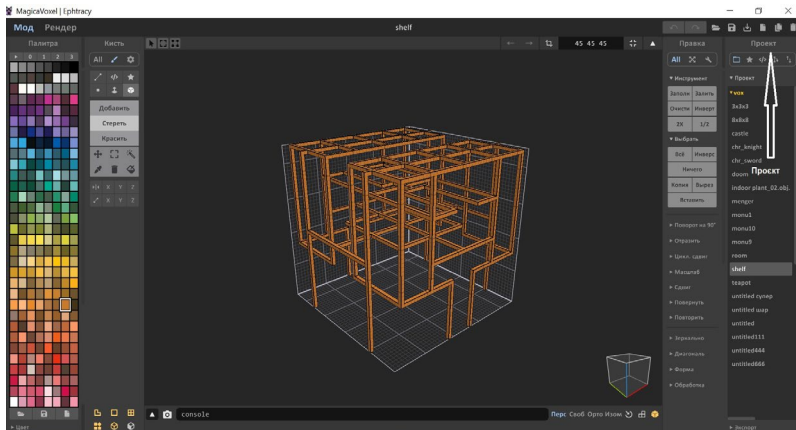


Рисунок 2.26

Загальні функції панелі «Проект» (рис. 2.26)

Панель «Проект» – це вбудований файловий менеджер, який дозволяє:

- переглядати всі наявні проекти;
- відкривати, створювати й перейменовувати сцени;
- експортувати модель у зовнішні формати (OBJ, VOX, PNG тощо).

Панель розташована праворуч від головного вікна (рис. 28) і за замовчуванням відкрита при старті програми.



Вона складається з двох основних розділів:

- «Проект» (Project) – робота з поточними файлами .vox.
- «Експорт» (Export) – збереження у різних форматах для подальшої роботи в інших програмах.

Блок «Інструменти проекту» (верхній ряд).

У верхній частині панелі «Проект» (рис. 2.27) – набір кнопок-іконок для роботи з файлами:

- «Папка» – відкрити директорію з моделями (vox), перейти в іншу папку;

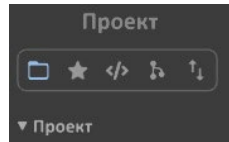


Рисунок 2.27

- «Зірочка» (Патерн) – позначити обраний файл як улюблений або виділити;
- «Код» (Shader) – відкрити кодову/текстову частину моделі або додаткові параметри;
- «Ланцюг» – зв'язати або від'єднати пов'язані об'єкти (групування моделей);
- «Стрілки» – змінити порядок проектів у списку (перетягування моделей).

Панель «Патерн» (Pattern) дозволяє створювати, зберігати й вставляти повторювані фрагменти воксельної моделі – шаблони. Це як конструктор: один раз створив – і можеш вставляти його в різні сцени, або багато разів у ту саму.

Панель «Shader» – це генератор об'єктів. Вона дозволяє створювати складні моделі за допомогою математичних функцій або автоматизованих шаблонів, це такі об'єкти як сфери, тори, шестерні, багатокутники та інші.

Список доступних генераторів (шейдерів):

- gear – шестерня;
- mandelbulb – 3D-фрактал;



- poly – багатокутник;
- round – округла форма;
- sphere – сфера;
- stair – сходи;
- torus – тор (бублик).

Практичні поради для користувачів щодо «Pattern» та «Shader»:

- «Pattern» – це як будівельні блоки, які ти створюєш сам;
- «Shader» – це автоматичні шаблони, які тільки користувач параметризує.

Обидва блоки можна комбінувати: згенеруй модель у Shader. Збережи її у Pattern і використовуй повторно.

2.17 Швидкі клавіші та робочі прийоми

MagicaVoxel – це програма, де клавіатурні скорочення (гарячі клавіші) можуть прискорити роботу у кілька разів. Хоча всі функції доступні через панелі інтерфейсу, використання комбінацій клавіш дозволяє моделювати без постійних перемикань мишкою (табл. 2.1).

Швидкі прийоми моделювання

- Блоковий підхід (Blockout): Починати з великих форм (куб, прямокутник), а потім деталізувати дрібними вокселями. Приклад: спочатку – стіни будинку, потім – вікна, двері, декор.

- Використання симетрії: Увімкнути Mirror (M), щоб працювати лише на половині моделі – інша половина буде повторюватися автоматично.

- Шаблони та копіювання: Створити об'єкт, зберегти в «Pattern», вставити скільки завгодно разів.

- Фіксація камери: Зберегти потрібний ракурс у слоті камери (Camera Slot) через панель Console (save c1) і потім швидко повернутись (load c1).



Таблиця 2.1 – Базові клавіші редагування

Клавіша / Комбінація	Дія
T	Додати (Attach) – розміщує воксель
R	Видалити (Erase) – стирає воксель
G	Фарбування (Paint) – змінює колір вокселя
M	Вибір (Select) – виділення області
Ctrl + Z	Скасувати дію (Undo)
Ctrl + Y	Повторити дію (Redo)
Ctrl + S	Зберегти сцену
ПКМ	Обертати камеру
Середня кнопка	Панорамування
Колесо миші	Зум
Shift + ПКМ	Швидке обертання на 45°
Tab	Перемикання між режимом редагування та режимом переміщення об'єкта (Global/Edit Mode)
Alt + Колесо миші	Плавний зум

Поради користувачу

Не намагайтесь відразу працювати тільки на клавіатурі – поєднувати клавіші з панелями, поки не запам'ятаються комбінації.

Для складних сцен камеру краще фіксувати – це економить час при рендері.

Вправа 1. Створення власної бібліотеки шаблонів (Pattern)

1. Створення об'єкта:



- відкрити нову сцену;
 - змодельовати просту форму: наприклад, стілець, колону або вікно;
 - виділити цей об'єкт за допомогою інструменту «Select» (використати «Панель інструментів» – «Вибір області»).
2. Збереження об'єкта як «Pattern»:
 - перейти на панель «Pattern»;
 - натиснути кнопку «+» додати новий шаблон;
 - об'єкт збережеться як новий шаблон у списку нижче.
 3. Використання шаблону:
 - очистити сцену або створити нову;
 - клікнути по новому шаблону у списку – він з'явиться в сцені;
 - повторно вставити шаблон у різні місця – можна його обертати, масштабувати, фарбувати.

Вправа 2. Генерація 3D-фігур за допомогою «Shader»

1. Вибір типу генерації:
 - перейти на панель «Shader»;
 - вибрати один із генераторів, наприклад: sphere – для сфери, gear – для шестерні, torus – для тора.
2. Налаштування параметрів:
 - в полі «Side» ввести розмір;
 - в полі «Angle» залишити «0», або спробувати «45» чи «90» для симетричних форм;
 - натиснути кнопку – генерація моделі.
3. Використання моделі:
 - модель з'явиться у сцені;
 - перевірити її – можна змінити розмір або перемістити;
 - натиснути «+» на панелі «Pattern», щоб зберегти її як шаблон;
 - повторити для ще 2 інших типів генераторів.



4. Блок «Експорт» (нижня частина екрану)

Блок «Експорт» (Export) дозволяє зберегти вашу воксельну модель у різні формати (рис. 2.28):

- 3D-моделі для подальшої обробки у Blender, Unity, Maya тощо;
- двовимірні зображення для рендерингу, презентацій, концептів;
- воксельні набори даних для програмного використання або моддингу.

Експорт – це фінальний етап роботи над проектом. Через нього модель стає доступною для інших програм чи кінцевого використання.



Рисунок 2.28

Структура панелі «Експорт»

Після розгортання цієї панелі (рис. 2.28), з'явиться список форматів, кожен із яких має свої особливості. Основними із них є:

1. **.VOX**. Це внутрішній формат MagicaVoxel. При натисканні на кнопку експорту в .vox користувач може:

- зберегти поточну модель у вибране місце;
- використовувати її повторно в інших проектах MagicaVoxel;
- поширювати як шаблон або частину бібліотеки.

2. **.OBJ** – класична 3D-модель. Формат .obj – це універсальний стандарт для обміну 3D-моделями. Підтримується: Blender, Unity, Unreal Engine, Sketchfab, Maya та інші. Тут екпортується: геометрія моделі, координати



вокселів, UV-мапи та текстури (якщо увімкнено). Супутні файли: `.obj` – геометрія, `.mtl` – матеріали, `.png` – текстура (якщо обрано).

3. **.PLY** – формат для 3D-сканування. PLY (Polygon File Format) часто використовується для: 3D-сканів, збереження кольору вокселів, наукових візуалізацій. Експортується як «хмара точок» або воксельна геометрія.

4. **.QB** – формат Qubicle. Це специфічний формат для програми Qubicle – альтернативного редактора вокселів. Потрібен, якщо ти плануєш: перейти з MagicaVoxel до Qubicle, використовувати Qubicle для анімацій чи експорту до Unity з LOD-системами.

5. **.PNG** – рендер/знімок сцени. Цей формат експортує зображення, а не 3D-модель. Можна експортувати: статичний рендер сцени (вигляд з камери), текстуру об'єкта, карти (наприклад, картографічні UV-мапи).

6. **.SLAB** – низькорівневий формат вокселів. Формат `.slab` використовується переважно у воксельних іграх або двигунах, які потребують точної структурованої інформації про блоки. Підходить для програмістів.

7. **.TXT** – текстовий дамп сцени. Це файл із текстовим описом об'єкта: координати вокселів, кольори, ідентифікатори. Підходить для: програмного аналізу сцени, вивчення структури.

8. **.CSV** – таблична структура. Екпортує воксельну модель як таблицю (використовується рідко, але може бути корисним для статистики або обробки даних).

Як експортувати модель.

- Завершити редагування сцени.
- Перейти на панель «Export».
- Обрати потрібний формат (наприклад, `.obj`).
- Вказати шлях для збереження (діалогове вікно).
- Натиснути «Export» – модель буде збережена разом із усіма додатковими файлами.



3 РЕЖИМИ ПЕРЕГЛЯДУ І КАМЕРИ У MAGICAVOXEL

3.1 Камера в 3D-сцені

У MagicaVoxel, як і в інших 3D-редакторах, сцена сприймається ніби через об'єктив камери. Проте ця камера не є фізичною. Вона визначає, звідки, під яким кутом і з яким спотворенням видно об'єкт. Саме камера визначає:

- який ракурс буде на фінальному зображенні;
- чи буде об'єм виглядати плоско чи глибоко;
- чи буде модель більше схожа на креслення, чи на реальний предмет.

MagicaVoxel має кілька основних режимів камери, які можна легко перемикає за потреби.

3.2 «Перспективна» камера (Perspective View)

Це стандартний режим перегляду (рис. 3.1). Його особливість у тому, що:

- об'єкти, які ближчі до камери, здаються більшими;
- ті, що далі – меншими (так, як бачимо у реальному житті).

Краї предметів можуть сходитися до точки на горизонті (лінії перспективи).

Цей режим чудово підходить для емоційного, «живого» зображення. Він додає глибини, об'єми виглядають масивно і виразно.

3.3 «Ізометрична» камера (Isometric View)

У цьому режимі (рис. 3.2):

- усі паралельні лінії залишаються паралельними;
- немає перспективних спотворень;
- кути між осями завжди однакові (найчасте 30° або 45°).



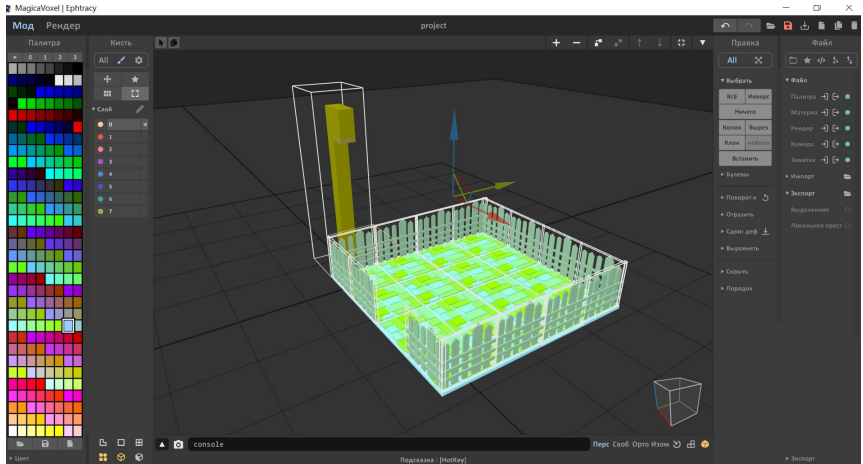


Рисунок 3.1

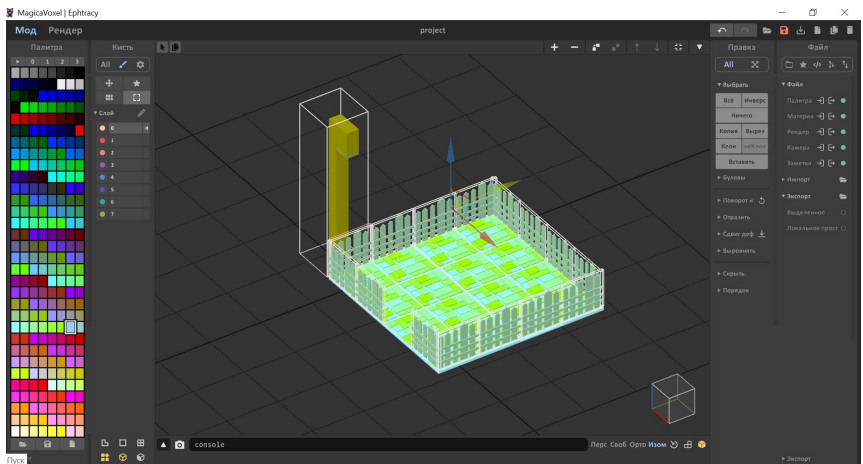


Рисунок 3.2

Модель ніби плаває у повітрі, її усі частини однаково помітні. Ізометрія дуже популярна в:

- 8-бітних іграх;
- мапах міст, дашбордах, діаграмах;
- мистецтві у стилі піксель-арт або кібертехнологіях.

У MagicaVoxel цей режим дозволяє:



- показати усю модель одразу;
- створити технічно коректне зображення;
- надати сцені стилізованого вигляду.

3.4 Ортогональна камера (Orthographic View)

Ортографічний режим – це як інженерне креслення (рис. 3.3). Тут:

- немає жодних спотворень;
- прямі залишаються прямими;
- об'єкти виглядають пласко, але чітко.

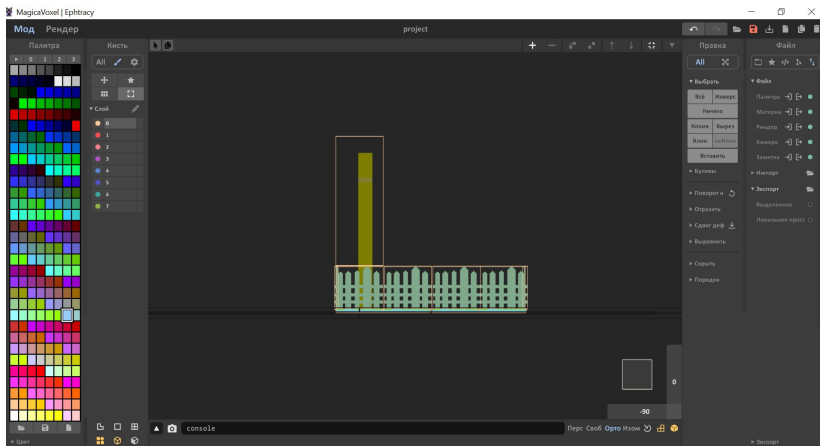


Рисунок 3.3

Цей режим ідеальний для:

- технічного моделювання;
- роботи з сіткою;
- анімацій у вигляді спрайтів;
- виводу схематичних зображень для документації.

У MagicaVoxel можна легко переключитись на ортографічний режим, коли потрібно:

- вирівняти частини моделі;
- перевірити симетрію;
- створити плитковий спрайт або текстуру.



4 ПАНЕЛІ ВІЗУАЛІЗАЦІЇ, РЕНДЕРИНГ У MAGICAVOXEL

Коли користувач вже створив свою модель у MagicaVoxel – чи то простий кубик, абстрактну форму чи модель будиночка – настає момент, коли хочеться побачити, як вона виглядатиме у «справжньому світі». Тут починається робота із візуалізацією. Візуалізація або рендеринг – це процес створення фотореалістичного зображення з тривимірної сцени.

У лівій, верхній частині вікна програми знаходиться кнопка «Render» – вона відкриває спеціальний режим перегляду, де об'єкт освітлюється світлом, на нього падають тіні, а матеріали (наприклад, скло або метал) набувають своїх характерних властивостей (рис. 4.1).

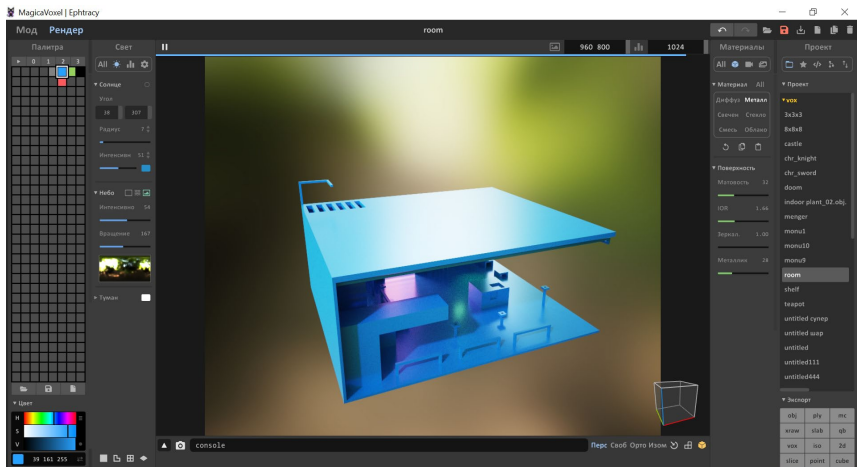


Рисунок 4.1

Важливо зазначити, що рендеринг у MagicaVoxel дуже швидкий і не потребує складного налаштування графіки чи підключення до потужних серверів – усе працює локально на комп'ютері.

У режимі рендерингу з'являються нові панелі. Перша з



них – це панель освітлення (рис. 4.2). Тут можна змінити кут нахилу джерела світла, його інтенсивність, колір, додати або прибрати тіні.

4.1 Панель «Світло»

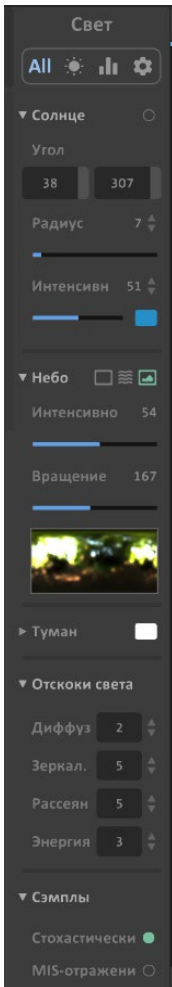


Рисунок 4.2

Світло – один із ключових параметрів у рендері, адже воно створює глибину, об’єм і настрої сцени. У MagicaVoxel є кілька джерел освітлення, які можна вмикати, вимикати й змінювати їхню інтенсивність і розташування. Основні параметри (рис. 4.2):

4.1.1 «Сонце» (Sun)

Імітує головне джерело світла – сонячне або штучне з направленим пучком.

Параметри:

– «Кут» (Angle) – два значення (азимут і висота) задають напрямок світла. Лівий повзунок – обертання по горизонталі (0–360°). Правий – кут підйому (0° – горизонт, 90° – прямо зверху).

– «Радіус» (Radius) – розмір джерела світла. Малий радіус – різкі тіні. Великий – м’які, розмиті тіні.

– «Інтенсивн.» (Intensity) – яскравість сонця.

– «Колір» – клік по квадратику дозволяє вибрати відтінок світла (наприклад, теплий жовтий або холодний блакитний). Порада користувачу: для ранкового світла встановіть малий кут



підйому (20–30°) і теплий колір.

4.1.2 «Небо» (Sky)

Додає глобальне фонове світло від небосхилу.

Параметри:

- «Інтенсивн.» (Intensity) – сила освітлення від неба;
- «Колір» – визначає колір атмосферного підсвічування (блакитний для денного світла, помаранчевий для заходу);
- «Іконка хмари» – перемикає між одноколірним фоном та градієнтним.

Порада користувачу: якщо ввімкнути «Небо», сцена матиме більш м'які тіні і не буде занадто темною у затінених місцях.

4.1.3 «Туман» (Fog)

Імітує атмосферний серпанок або туман (рис. 4.3).

Параметри:

- «Перемикач» – увімкнення/вимкнення;
- «Колір» – відтінок туману;
- «Intensity» (у заданому меню регулюється автоматично через відстань) – чим більша щільність, тим сильніше об'єкти зливаються з фоном на відстані.

Порада користувачу: використовуйте туман для сцен з великою глибиною (наприклад, ландшафти або вулиці).

4.1.4 «Відскоки світла» (Light Bounces)

Цей блок керує тим, як світло відбивається і розсіюється між об'єктами.

- «Дифуз» (Diff) – кількість м'яких розсіювань світла від поверхонь. Збільшення робить освітлення більш реалістичним, але збільшує час рендера.
- «Дзеркал.» (Spec) – інтенсивність дзеркальних відблисків.
- «Розсіюван.» (Scatter) – глибоке розсіювання (імітує



проходження світла крізь напівпрозорі об'єкти).

– «Енергія» (Energy) – загальна потужність відбитого світла.

Порада користувачу: для реалістичних сцен тримати «Diff» та «Spec» на середньому рівні, «Scatter» використовувати для рослин і тканин.

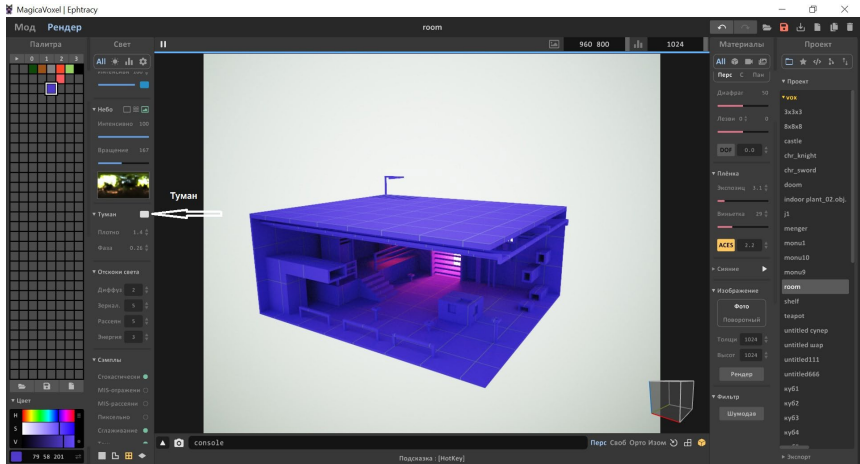


Рисунок 4.3

4.1.5 «Семпли» (Samples)

Визначає спосіб, яким MagicaVoxel обчислює освітлення (рис. 4.4):

- «Стохастично» – додає невеликий шум, але прискорює рендер;
- «MIS-відбиття» – точніше обчислює багатократні відбиття світла;
- «MIS-розсіювання» – покращує обчислення розсіювання;
- «Пікельно» – зменшує якість, але значно прискорює прев'ю;
- «Згладжування» – згладжує краї об'єктів у пікельному рендері;

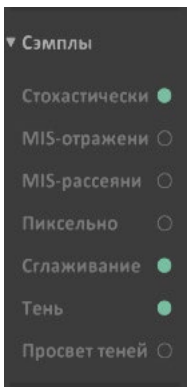


Рисунок 4.4



- «Тінь» – вмикає/вимикає тіні.

Порада користувачу: для фінального рендеру вмикати всі опції, для швидкої перевірки сцени – лише необхідні.

4.2 Панель «Матеріали»

Кожен воксель у MagicaVoxel має колір, але в режимі рендеру він також може мати матеріал – тобто відображати властивості поверхні: металевість, прозорість, скло, світіння тощо. Матеріали призначаються через палітру (рис. 4.5). Кожен колір у палітрі має окремі налаштування:

Типи матеріалів:

4.2.1 Дифуз (Diffuse) – базовий матовий матеріал

Це стандартний тип матеріалу, який рівномірно відбиває світло у всі сторони. Не має прозорості чи металевого блиску. Найкраще підходить для об'єктів, які повинні виглядати природно і не відбивати світло занадто інтенсивно.

Параметри:

- «Diffuse» = 1.0 (за замовчуванням);
- «Roughness» – визначає, наскільки м'якими будуть відблиски (для Diffuse зазвичай високе значення);
- «Specular» – мінімальне або нульове.

Приклади застосування: камінь, цегла, бетон, дерев'яні дошки, тканина, пластик з матовою поверхнею.

4.2.2 Метал (Metal) – матеріал з металевим блиском

Метал відбиває світло, зберігаючи власний колір відблиску. Чим вищий параметр Metal, тим менше матеріал виглядає як фарба і більше – як полірований метал.

Параметри:

- «Metal» = 1.0 (повністю металевий);
- «Specular» – регулює інтенсивність блиску;



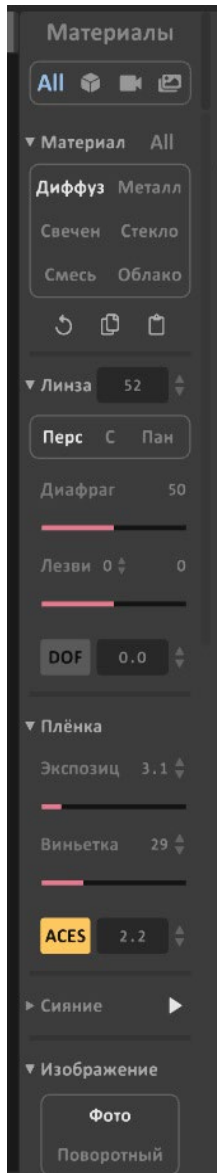


Рисунок 4.5

– «Roughness» – чим менше значення, тим гладший і дзеркальніший метал. Приклади застосування: золото (жовтий колір + Metal 1.0 + Roughness 0.05), сталь (сірий колір + Metal 1.0 + Roughness 0.2), мідь або бронза (коричнево-червоний + Metal 1.0 + трохи вищий Roughness).

4.2.3 Світіння (Emission) – матеріал, що світиться

Цей матеріал сам випромінює світло. Він не потребує додаткових джерел освітлення для створення яскравих зон у рендері. Яскравість залежить від «Emission» і може підсвічувати інші об'єкти при налаштованих відбиттях.

Параметри:

- «Emission» – сила світіння (чим більше, тим яскравіше);
- «Bloom» у рендері – підсилює ефект саява;
- колір у палітрі – визначає колір світла.

Приклади застосування: неонові вивіски, лампи, ліхтарі, футуристичні екрани або індикатори, містичні кристали чи магичні сфери.

4.2.4 Скло (Glass) – прозорий матеріал із заломленням світла

Імітує скло, лід, кришталь. Має параметри прозорості, коефіцієнта заломлення та шорсткості. При низькій



шорсткості (Roughness = 0) виглядає як ідеально чисте скло, при високій – як матове або «молочне».

Параметри:

- «Transparency» – ступінь прозорості (0.0 = непрозорий, 1.0 = повністю прозорий);
- «Refraction» – коефіцієнт заломлення (1.0 для повітря, ~1.5 для скла, >2 для дорогоцінного каміння);
- «Roughness» – матовість поверхні;
- «Specular» – інтенсивність відблисків.

Приклади застосування: віконне скло, лід, кришталеві келихи, прозорі кулі з магічними ефектами.

4.2.5 Суміш (Mix) – комбінування властивостей

Дозволяє змішати властивості двох матеріалів. Наприклад, можна зробити напівпрозорий метал або скло з емісивним ефектом.

Параметри:

- комбінується «Transparency», «Metal», «Emission»;
- можна створювати унікальні ефекти, недоступні в інших режимах.

Приклади застосування: скло з металевим покриттям, напівпрозорі кристали з внутрішнім світінням, лаковані металеві поверхні.

4.2.6 Хмара (Cloud) – об'ємний розсіюний матеріал

Імітує об'ємні ефекти: туман, хмари, дим, м'які частинки. Не є повністю твердим тілом, світло проходить крізь нього та розсіюється.

Параметри:

- «Scattering» – сила розсіювання світла всередині об'єму;
- «Absorption» – колір та інтенсивність поглинання світла;
- «Transparency» – прозорість;
- «Emission» (за бажанням) – внутрішнє світіння



(наприклад, сяючі хмари).

Приклади застосування: хмари у небі, туман у лісі, дим з вогню, магичні газові ефекти.

4.3 Панель «Ефекти»

Режим «Render» у MagicaVoxel дозволяє не тільки налаштовувати світло та матеріали, але й застосовувати різноманітні ефекти для створення особливого настрою сцени. Ці інструменти допомагають імітувати атмосферу, додають глибину і навіть дозволяють робити художні кадри, схожі на ілюстрації.

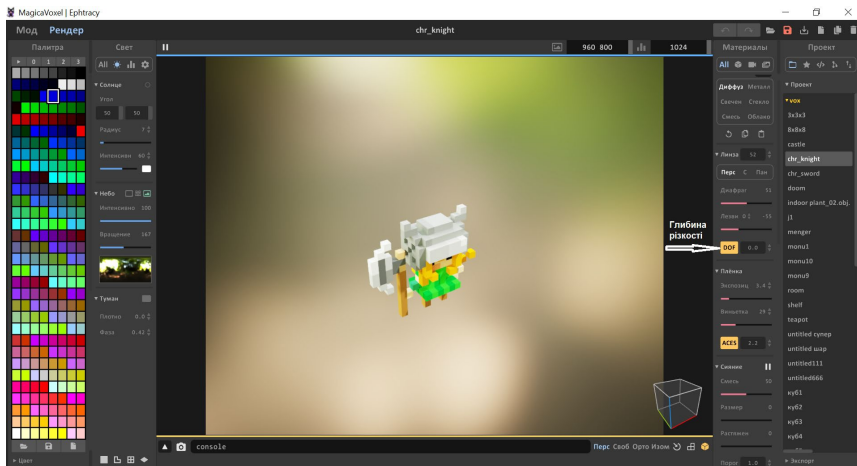


Рисунок 4.6

4.3.1 «Лінза» (Lens)

Керує тим, як камера «бачить» сцену:

- «Число» (наприклад, 52) – це фокусна відстань камери (Field of View). Мале значення (20–30) – широкий кут. Велике значення (70–100) – вузький кут, сильний зум;
- «Перс» (Pers) – перспектива. Класичний 3D-ефект із зменшенням об'єктів у даліні;
- «С» (C) – ортографічний режим. Без перспективних



спотворень (усі паралельні лінії лишаються паралельними);

- «Пан» (Pan) – панорамний режим камери;
- «Діафрагма» (Aperture) – впливає на глибину різкості

(DOF). Чим більше значення, тим сильніше розмиття у зоні поза фокусом;

- «Лезо» (Blade) – кількість пелюсток діафрагми, впливає на форму світлових плям (боке).

DOF – сила розмиття об'єктів поза фокусом (рис. 4.6).

4.3.2 «Плівка» (Film)

Імітує ефекти фотокамери:

- «Експозиція» (Exposure) – регулює яскравість кадру.

Збільшення – кадр світліший. Зменшення – темніший;

- «Віньетка» (Vignette) – затемнює кути зображення, фокусуючи увагу в центрі (рис. 4.7).

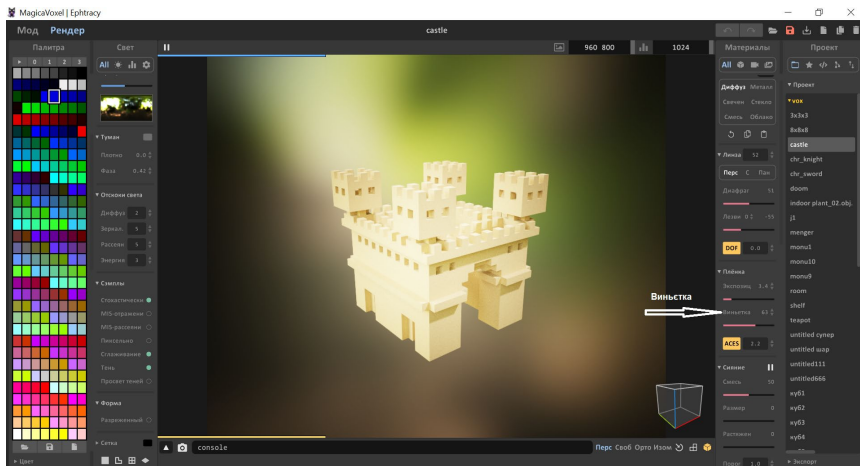


Рисунок 4.7

4.3.3 «ACES»

Професійна система тональної корекції (tone mapping) (рис. 4.8):



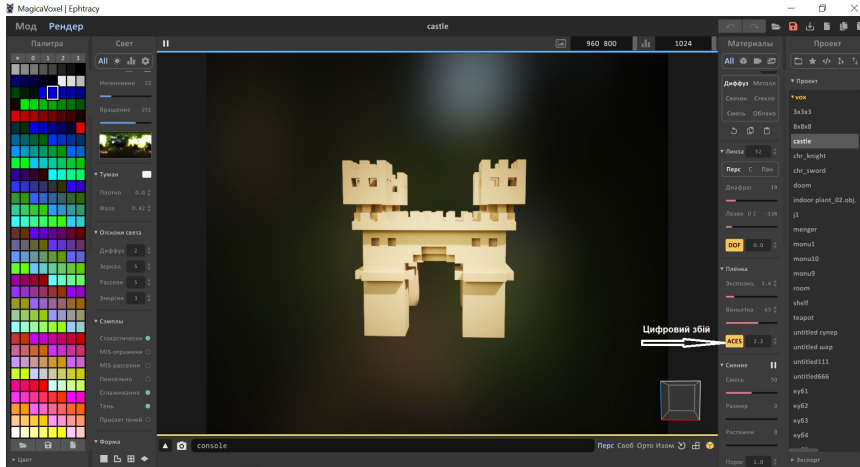


Рисунок 4.8

- «Число» (наприклад, 2.2) – регулює контраст і насиченість, зберігаючи кольори реалістичними при яскравому світлі;
- при низьких значеннях картинка стає тьмянішою, при високих – контрастною та соковитою.

4.3.4 «Світіння» (Bloom)

- Додає м’який ореол навколо яскравих ділянок (рис. 4.9):
- інтенсивність «Bloom» залежить від налаштування в самому ефекті та від яскравості матеріалу (Emission);
- «Bloom» виглядає особливо ефектно на неоні, лампах, світлових мечях, вогні.

4.3.5 «Зображення» (Image)

- «Фото» (Photo) – збереження поточного кадру у форматі PNG.
- «Поворотний» (Turntable) – автоматичне створення серії знімків, де об’єкт повільно обертається.



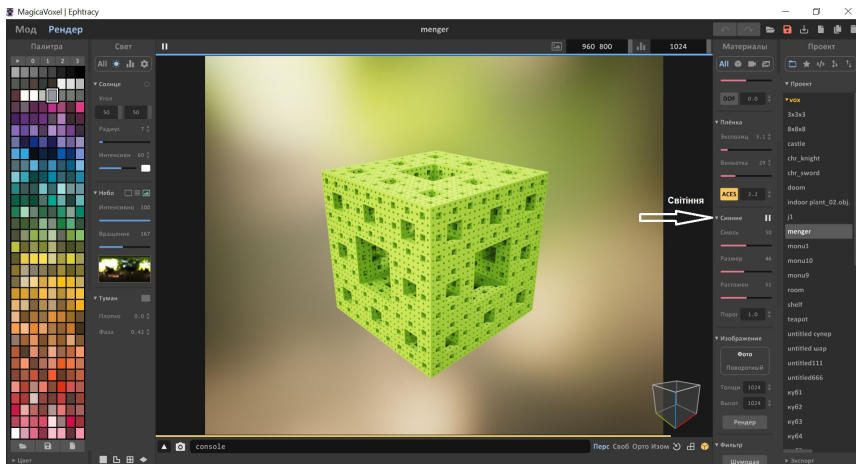


Рисунок 4.9

Ефекти застосовуються в реальному часі до рендеру й не змінюють саму модель. Вони потрібні лише для фінального знімка. Розмір зображення, яке буде збережене, також можна змінювати. За замовчуванням рендер відбувається у квадратному форматі 1024×1024 пікселі, але ви можете вибрати будь-яку іншу роздільність, наприклад, 1920×1080 – стандарт для широкоформатного зображення.

Після завершення рендерингу зображення можна зберегти. Для цього потрібно натиснути кнопку з дискеткою (або піктограму камери) та обрати формат збереження. Найчастіше використовується PNG, оскільки цей формат підтримує прозорий фон і не стискає зображення, на відміну від JPEG. PNG найкраще підходить для подальшого редагування в графічних редакторах або публікації в інтернеті.

Дуже важливо зберігати не тільки зображення, а й сам проєкт. Для цього у верхній частині інтерфейсу існує кнопка «Save». Натиснувши її, користувач збереже свій файл у форматі .vox. Усі об'єкти, їх кольори, положення в сцені та навіть палітра – все залишиться таким, яким його зберегли.

Надалі є можливість відкрити цей файл, змінити модель



або зробити новий рендер з іншими параметрами.

Також варто знати, що можна зберегти не лише саму сцену, а й експортувати її як 3D-модель у форматі .obj. Цей формат підтримується у більшості 3D-редакторів – наприклад, Blender або Unity. Таким чином, модель може бути використана у грі, віртуальній сцені, презентації або навіть надрукована на 3D-принтері, якщо її правильно підготувати.

У MagicaVoxel передбачено також автозбереження. Якщо випадково закрити програму або щось пішло не так, при наступному відкритті MagicaVoxel запропонує відновити попередню сесію – це дуже корисна функція, особливо коли працюєш з великими моделями.

Завершуючи цей розділ, варто наголосити: ніколи не забувати регулярно зберігати виконану роботу. У цифровому середовищі це – запорука спокою і впевненості, що модель не зникне через випадкову помилку чи збій.

Рендеринг у програмі – це не тільки про гарні картинки, а й про можливість побачити свою ідею «наживо», перевірити пропорції, матеріали, освітлення і, можливо, знайти щось нове, що можна додати до свого воксельного світу.



5 ЕКСПОРТ ТА ІНТЕГРАЦІЯ У ЗОВНІШНІХ ПРОГРАМАХ

5.1 Навіщо потрібен експорт

MagicaVoxel добре підходить для створення воксельних моделей, але часто вони є лише одним етапом у великому творчому процесі.

Завдяки експорту можна:

- Використати модель у Blender для складнішого моделювання, анімацій та фотореалістичного рендеру.
- Перенести модель у Unity або Unreal Engine для інтерактивних сцен та ігор.
- Підготувати файл для 3D-друку або AR/VR.
- Використати модель у графічних редакторах для ілюстрацій та відео.

Експорт – це перехід від внутрішнього формату MagicaVoxel до універсальних форматів, які розуміють інші програми.

5.2 Формати експорту MagicaVoxel

У меню «Export» (клавіша F6) можна знайти кілька форматів (таблиця 5.1). Кожен має свої переваги та обмеження:

5.3 Підготовка моделі до експорту

Перед збереженням моделі у зовнішній формат слід виконати кілька кроків оптимізації:

- прибрати зайві вокселі. Переконайтесь, що в сцені немає об'єктів за межами камери або прихованих всередині. Використати інструменти «Erase» або послідовно застосувати команди «Select» і «Delete»;
- об'єднати шари (Layers). Якщо модель складається з



багатьох частин, але не планується рухати їх окремо – об’єднати для зменшення кількості об’єктів.

Таблиця 5.1

Формат	Призначення	Переваги	Обмеження
.vox	Рідний формат MagicaVoxel	Повністю зберігає палітру, шари, матеріали	Використовується лише у MagicaVoxel
.obj	Універсальний формат для 3D	Підтримується Blender, Unity, Maya, Cinema4D	Генерує 3 файли: .obj, .mtl, .png
.ply	Формат для наукових звітів і 3D-друку	Зберігає колір кожного вокселя	Не завжди підтримує матеріали
.qb	Формат Qubicle	Сумісний з іншими воксельними редакторами	Не підтримує всіх можливостей MagicaVoxel
.png	2D-рендер сцени	Підходить для прев’ю або текстур	Не містить 3D-даних
.iso / .slab	Ізометричні спрайти	Використовується у 2D-іграх	Тільки картинка

- оптимізувати палітру. Видали невикористані кольори. Збережи палітру як окремий файл на випадок змін;
- перевірити масштаб. У MagicaVoxel 1 воксель – це умовна одиниця. У Blender або Unity він може мати інший розмір, тому масштаб треба буде налаштувати;
- орієнтація моделі. У MagicaVoxel вісь Y – вертикальна. У Blender вісь Z – вертикальна, тому при імпорті модель може «лежати».



5.4 Експорт у формат .OBJ (рекомендований)

1. Відкрити модель у MagicaVoxel.
2. Натиснути F6 або кнопку Export.
3. У вікні вибору формату обрати OBJ.
4. Вказати папку для збереження.
5. MagicaVoxel створить: model.obj – геометрія, model.mtl – матеріали, model.png – текстура палітри.
6. Переконайтесь, що всі 3 файли залишилися в одній папці.

5.5 Імпорт у Blender

1. Відкрити Blender.
 2. Видалити стандартний куб (X – Delete).
 3. Перейти у «File» – «Import» – «Wavefront» (.obj).
 4. Знайти свій model.obj і вибери його.
 5. У вікні імпорту встановити: Forward: – Z Forward, Up: Y Up.
 6. Натиснути «Import OBJ».
- Якщо текстури не з'явилися: перейти у вкладку «Shading», знайти вузол «Image Texture» та вибрати model.png.

5.6 Приклад – інтеграція в Unity

1. Створити новий проект.
2. Скопіювати model.obj, model.mtl і model.png у «Assets».
3. Перетягнути модель на сцену.
4. Якщо кольори розмиті:
 - Виділити model.png.
 - У «Inspector» змінити «Filter Mode» – «Point» (no filter).
 - «Compression» – «None».



6 СИМУЛЯЦІЯ РУХУ ТА АНІМАЦІЯ У MAGICAVOXEL

На відміну від класичних 3D-редакторів, MagicaVoxel не має повноцінного анімаційного таймлайну, як у Blender чи Maya. Але це не означає, що анімацію тут неможливо виконати. Просто вона має інший принцип: покадрову анімацію через дублювання сцен, об'єктів і їхніх положень.

У цьому випадку створюється не модель, яка рухається, а серія моделей, кожна з яких є окремим кадром руху.

Цей підхід, хоча і спрощений, однак дає певну творчу свободу і дозволяє створювати ефекти руху, трансформації, живих сцен і навіть короткі мультки.

6.1 Підготовка кадрів у MagicaVoxel

Один кадр анімації – це окрема модель або сцена. Користувач може зробити кілька версій тієї самої сцени з невеликими змінами, і потім рендерити їх одну за одною. Кожна – це зупинка в часі. Послідовність дій:

1. Створи кожен кадр анімації окремо (у вигляді «Pattern» або як окремі сцени у «Project Panel»).
2. Усі кадри повинні мати: однаковий розмір сцени (наприклад, 40×40×40); однакове положення об'єкта відносно центру; однаковий ракурс камери (фіксуй його в «Camera Slots» через «Console», наприклад save c1, потім load c1). Якщо камера навіть трохи зміститься – GIF буде «смикатися».

6.2 Експорт окремих PNG для GIF

Кроки:

1. Увійти в «Render Mode».
2. Завантажити перший кадр.
3. Перевірити ракурс камери (повернути через load c1 у консолі, якщо потрібно).



4. Натиснути F6 або клікнути на іконці фотоапарата у верхньому меню.
5. Зберегти з назвою j_01.png.
6. Повторити для всіх кадрів (j_02.png, j_03.png і т. д.).
7. Переконаватися, що всі PNG мають однакові розміри (наприклад, 512×512 пікселів).

6.3 Створення GIF з PNG-файлів (Easy GIF Animator)

1. Запустити Easy GIF Animator.
2. Оберати «Create New Animation».
3. Додати усі PNG кадри (Add Image Files, виділити j_01.png ... j_06.png).
4. Виділити всі кадри, правий клік, «Set Delay» (100–150 мс для плавного руху).
5. Увімкнути «Loop» (зациклення).
6. Натиснути «Preview», щоб перевірити рух.
7. Зберегти як .gif.

6.4 Приклад створення покадрової анімації

Створення ефекту «Стрибок кубика» – класичний приклад руху із виразною пластикою у MagicaVoxel. Мета – створити анімацію з 6 кадрів, у якій кубик:

1. Стоїть нерухомо на землі;
2. Стискається, ніби набирає силу для стрибка;
3. Стрибає вгору;
4. Досягає піку;
5. Падає вниз;
6. Знову приземляється, стискається й трохи підстрибує.

Це дозволить відчувати ритм, вагу, зміну форми, швидкість – усе, що важливо для живого анімаційного ефекту.



1. Створення першого кадру: «спокійний кубик».

Відкрити MagicaVoxel, створити новий проєкт. Задати розмір сцени: $X = 20, Y = 20, Z = 20$;

Створити «нижню основу»: інструмент: «Додати Vox»; колір – сірий; координати: $X = 0...19, Y = 0, Z = 0...19$.

Створити кубик: «Додати Vox»; розміри: $4 \times 4 \times 4$; розмістіть на землі, центр – $X = 8, Y = 1, Z = 8$; Колір: червоний або жовтий (щоб виділявся);

Зберегти сцену як перший кадр (рис. 6.1): «Зберегти» куб1.vox.

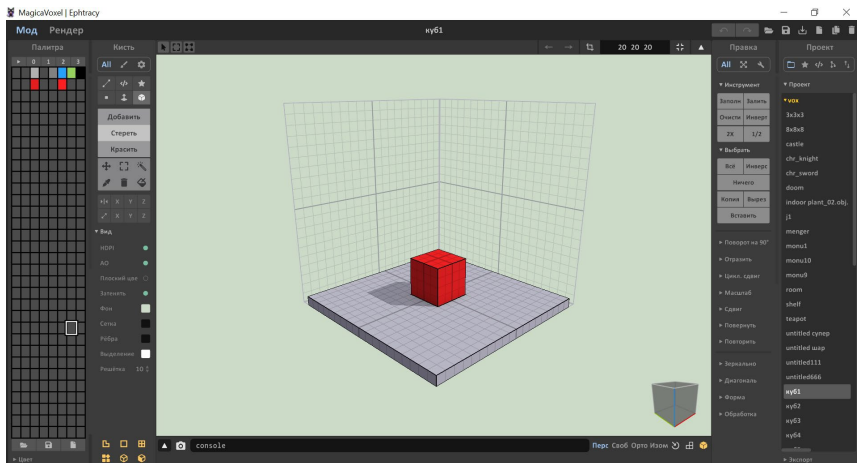


Рисунок 6.1

2. Кадр 2: Стискання.

Кубик має «стиснутись» перед стрибком, як м'ячик.

Відкрити куб1.vox і виконати «Зберегти» куб2.vox.

Видалити кубик: «Видалити»; побудувати новий зменшений по висоті, але розширений по ширині ($5 \times 2 \times 5$); Це імітує «напруження»; залишити його на тому самому місці (центр на $X = 8, Z = 8$) (рис. 6.2).

За бажанням – трохи затемнити колір: він «нагрівається» перед стрибком.



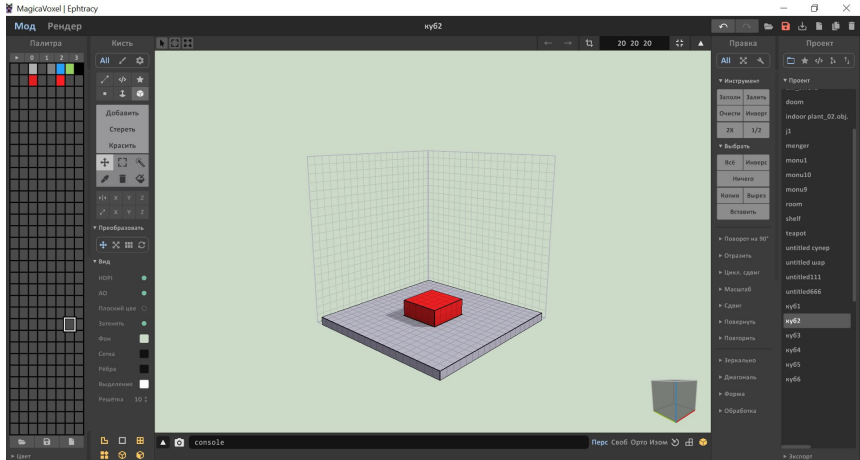


Рисунок 6.2

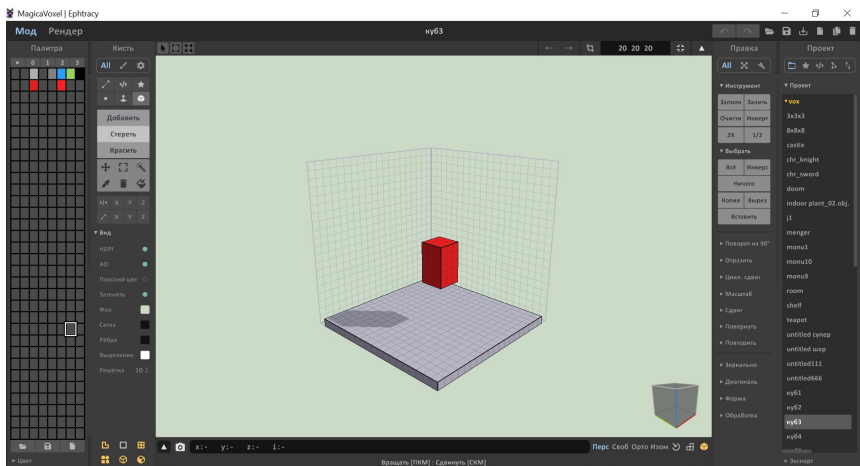


Рисунок 6.3

3. Кадр 3: Відрив від основи.

Кубик відривається і летить угору.

Відкрити куб2.vox і виконати «Зберегти» куб3.vox.; видалити старий кубик; побудувати новий, трохи витягнутий по вертикалі: розміри: $3 \times 5 \times 3$; центр підняти по Y: $Y = 5$; це створює ефект «пострілу» – кубик різко летить. Знову зберегти



сцену (рис. 6.3).

4. Кадр 4: Пік стрибка

Кубик у повітрі, стабільна форма. Відкрити куб1.vox.

Зберегти як куб4.vox.; побудувати ідеально рівний кубик $4 \times 4 \times 4$; центр задати по $Y = 8$ (максимальна висота); зробити колір трохи світлішим (ефект освітлення вгори). Перезберегти сцену (рис. 6.4).

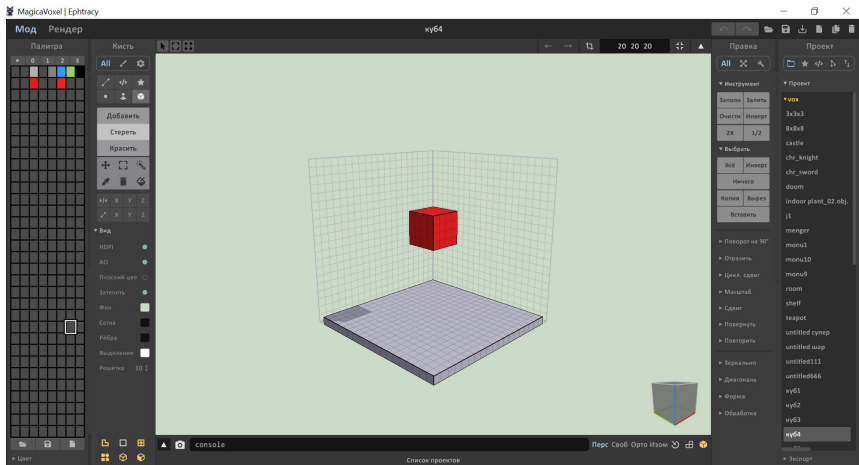


Рисунок 6.4

5. Кадр 5: Падіння

Кубик падає, починає стискатись.

Відкрити куб2.vox і виконати «Зберегти» куб5.vox.; розмістити кубик на $Y = 3$; Форма: $5 \times 2 \times 5$ (знову стискається ніби під силою тяжіння); Колір – теплий, можливо, додати сяйва («Emission» = 0.2) (рис. 6.5).



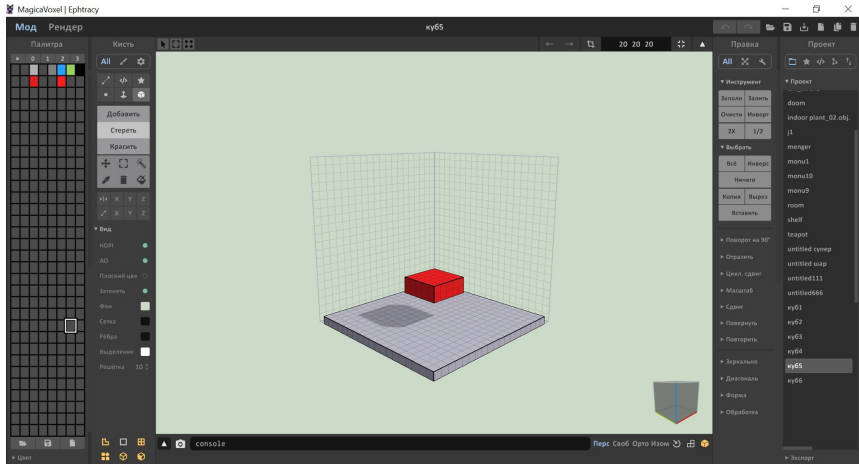


Рисунок 6.5

6. Кадр 6: Відскок

Кубик трохи підстрибує і стабілізується.

Відкрити `куб1.vox` і виконати «Зберегти» `куб6.vox`.; форма: $4 \times 3 \times 4$; розміщення: $Y = 2$. Це фінальна амортизація куба після падіння (рис. 6.6).

Рендер усіх кадрів

1. Відкривати по черзі кожен файл окремо.
2. Встановити однакові для всіх сцен: камеру («Isometric» або «Front»); освітлення («Sun Dir», «Intensity»); Фон (BG Color = білий або прозорий).
3. Рендер PNG: «Зберегти» `куб1.png`. – `куб6.png`.



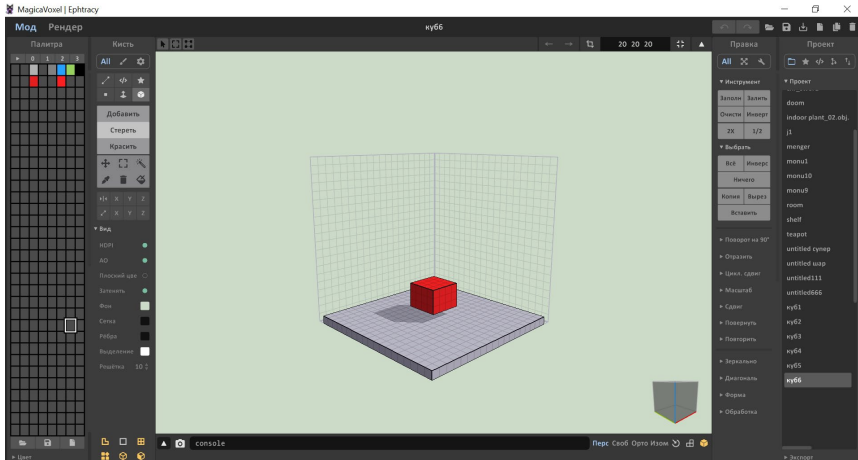


Рисунок 6.6

Збірка GIF-анімації

Варіант 1 – онлайн:

1. Перейти на <https://ezgif.com/maker>;
2. Завантажити всі зображення;
3. Встановити час: 100–150 мс/кадр;
4. Створити GIF та зберегти.

Варіант 2 – у відеоредакторі:

1. Скачати будь-який безкоштовний редактор GIF-анімації;
2. Завантажити у нього всі зображення;
3. Створити GIF та зберегти.



7 ОСНОВИ ПАРАМЕТРИЧНОГО ПРОЄКТУВАННЯ

7.1 Поняття параметра, параметризації та параметричного проєктування

У сучасному цифровому проєктуванні форма розглядається не лише як завершений геометричний результат, а як наслідок дії певної системи залежностей, правил і змінних. Одним із ключових понять такого підходу є параметр.

Параметр – це змінна величина або характеристика об'єкта, зміна якої впливає на його форму, розміри, структуру, положення елементів, пропорції чи інші властивості (рис. 7.1). У цифровому моделюванні параметрами можуть бути лінійні розміри, кути, радіуси, кількість повторів елементів, відстані між модулями, товщина оболонки, висота об'єкта, щільність структури, орієнтація компонентів тощо. У параметричних середовищах параметри можуть доповнюватися обмеженнями, формулами, змінними та логічними зв'язками між елементами моделі.

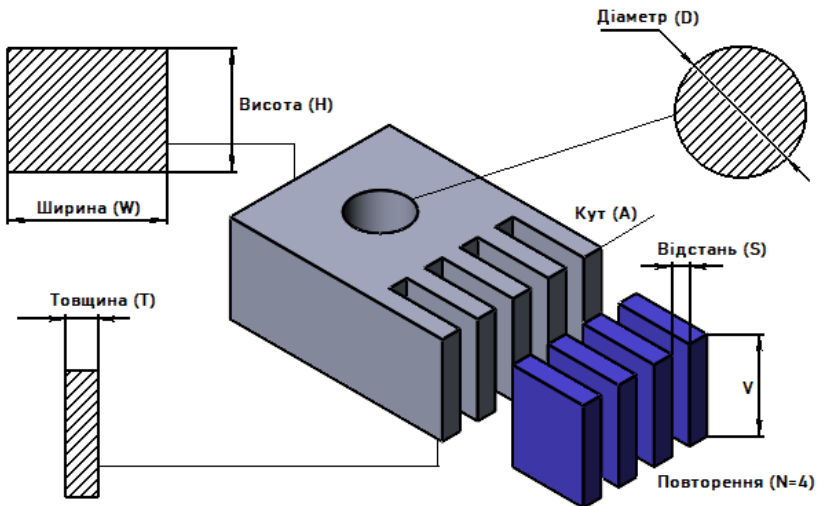


Рисунок 7.1



Параметризація є процесом задання параметрів і встановлення залежностей між ними. Її зміст полягає в тому, що проєктувальник не просто формує окрему геометричну конфігурацію, а створює систему керування формою, у межах якої об'єкт може змінюватися без повного перемодельовання. Отже, у параметризованій моделі визначається не лише зовнішній вигляд об'єкта, а й правила його трансформації.

На відміну від традиційного геометричного моделювання, де зміна одного елемента часто потребує ручного коригування інших частин моделі, параметричне проєктування передбачає наявність зв'язків між елементами (рис. 7.2).

Перехід від ручного до параметричного проєктування

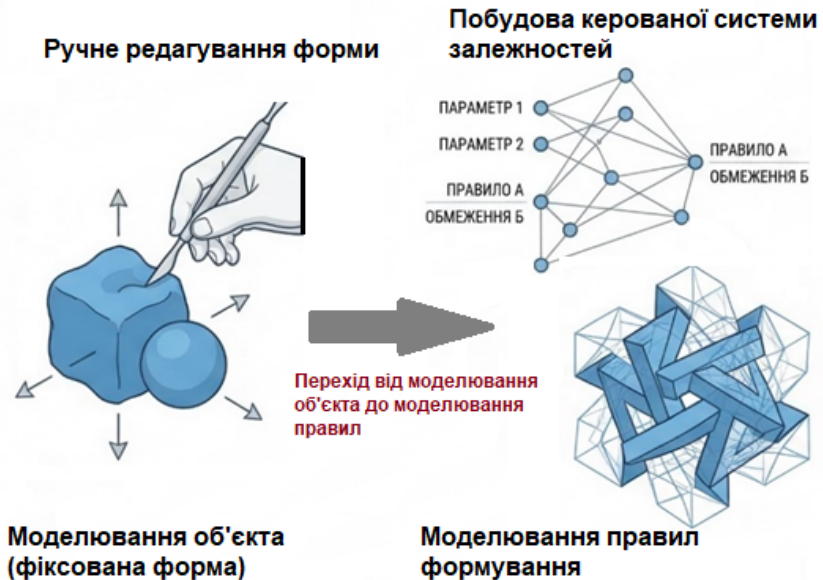


Рисунок 7.2

У загальному вигляді параметричне проєктування можна визначити як метод цифрового формоутворення, за якого



об'єкт описується через сукупність параметрів, зв'язків, обмежень і правил перетворення, а зміна вхідних даних спричиняє автоматичне оновлення моделі (рис. 7.3). Такий підхід дозволяє створювати не одну фіксовану форму, а множину можливих варіантів, що особливо важливо у випадках, коли потрібно швидко досліджувати альтернативи.

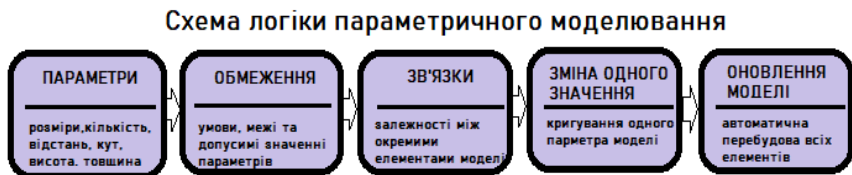


Рисунок 7.3

З позицій навчального проєктування параметричний підхід важливий не лише як технічний інструмент, а і як особливий тип мислення. Користувач у такому випадку працює не тільки з формою як результатом, а й з логікою її побудови: формулює умови, встановлює залежності між частинами моделі, визначає головні та другорядні змінні, аналізує, як зміна одного параметра впливає на композицію, конструкцію чи структуру об'єкта.

Разом із тим сутність параметричного підходу не зводиться до використання певного програмного забезпечення. Його основою є перехід від ручного редагування форми до побудови керованої системи залежностей. Тому параметричне проєктування доцільно розглядати як перехід від моделювання окремого об'єкта до моделювання правил його формування.

Важливо також підкреслити, що параметричний підхід може реалізовуватися як у неперервному геометричному середовищі, так і в дискретному (рис. 7.4). У першому випадку зміни параметрів переважно призводять до плавної трансформації поверхонь або об'ємів. У другому – до перебудови модульної, коміркової або воксельної структури.



Саме ця теза є принципово важливою для подальшого переходу до дискретно-параметричного та воксельного проєктування.

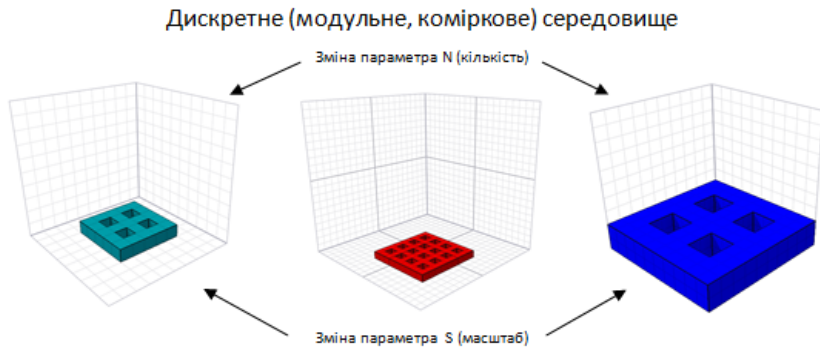
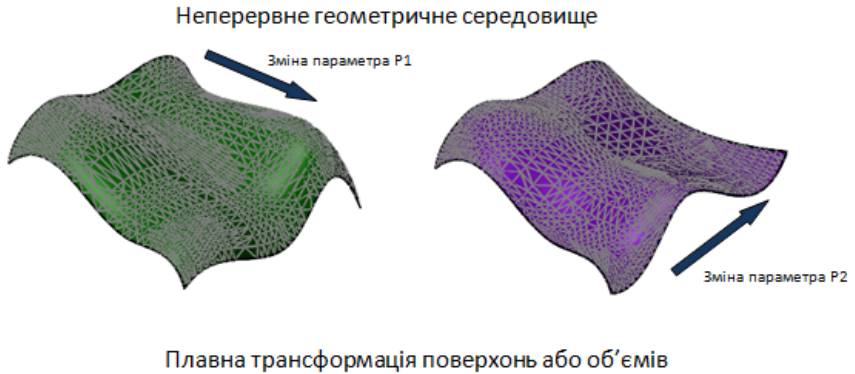


Рисунок 7.4

Таким чином, параметр, параметризація та параметричне проєктування утворюють послідовний понятійний ланцюг: параметр є окремою змінною, параметризація – процесом організації змінних і зв'язків, а параметричне проєктування – способом формоутворення на основі таких залежностей. Саме



ця логіка є вихідною для подальшого розгляду воксельного моделювання та параметричних можливостей середовища MagicaVoxel.

7.2 Історія виникнення і розвитку параметричного проєктування

Історія параметричного проєктування пов'язана з розвитком цифрових методів опису форми, комп'ютерної графіки, автоматизованого проєктування та алгоритмічного мислення. Його поява не була одномоментною подією, а стала результатом поступового переходу від ручного креслення і статичного геометричного моделювання до систем, у яких форма почала розглядатися як наслідок змінних величин, обмежень і взаємозв'язків між елементами моделі.

Передумови параметричного проєктування простежуються ще у ранніх практиках формоутворення, де геометрія об'єкта задавалася не лише безпосереднім кресленням, а через правила побудови, пропорційні співвідношення, модульні сітки або фізичні моделі. У цьому контексті показовими є експерименти Антоніо Гауді з ланцюговими моделями, у яких форма визначалася системою силових і геометричних залежностей (рис. 7.5). Такі приклади ще не є параметричним проєктуванням у сучасному цифровому розумінні, проте демонструють важливу ідею: форма може бути не довільно намальованою, а керовано сформованою на основі умов і правил.

Технологічний фундамент майбутнього параметричного проєктування почав формуватися в середині ХХ століття разом із розвитком комп'ютерної графіки. Однією з найважливіших віх стала система Sketchpad, створена Іваном Сазерлендом у 1963 році в Массачусетському технологічному інституті.





Рисунок 7.5

У цій системі користувач міг взаємодіяти з графічними об'єктами на екрані в режимі реального часу, використовувати світлове перо, задавати геометричні співвідношення і повторювати елементи (рис. 7.6). Саме Sketchpad часто розглядають як один із витоків комп'ютерного креслення і важливий крок до систем, у яких форма вже не просто зображується, а описується через керовані залежності.

У другій половині XX століття розвиток CAD-систем зосереджувався насамперед на автоматизації креслення, геометричного моделювання і технічного конструювання. На початкових етапах більшість таких систем були орієнтовані переважно на створення та редагування геометрії, однак поступово в них почали з'являтися механізми фіксації логічних і розмірних зв'язків між елементами. Це стало важливим кроком до параметризації, оскільки дозволило перейти від окремих графічних операцій до узгодженого керування моделлю.





Рисунок 7.6

Подальший розвиток параметричного підходу був пов'язаний із тим, що модель почали розуміти не лише як форму, а як систему параметрів, обмежень, змінних та правил зміни. У такому трактуванні геометрія задається параметрами й взаємозалежностями, а зміна одного значення може автоматично оновлювати пов'язані частини моделі.

Особливо важливим параметричне проектування стало для архітектури, дизайну та інженерії наприкінці ХХ – на початку ХХІ століття, коли цифрові технології дали змогу працювати зі складною геометрією, адаптивними системами, варіативними структурами і великою кількістю альтернативних рішень. У цей період почали активно розвиватися програмні середовища та надбудови, орієнтовані не лише на моделювання форми, а й на побудову алгоритму її створення. Одним із найвідоміших напрямів стало поширення візуального алгоритмічного моделювання, зокрема через Grasshopper для Rhino, а також параметрично-асоціативних систем, таких, як GenerativeComponents (рис. 7.7).



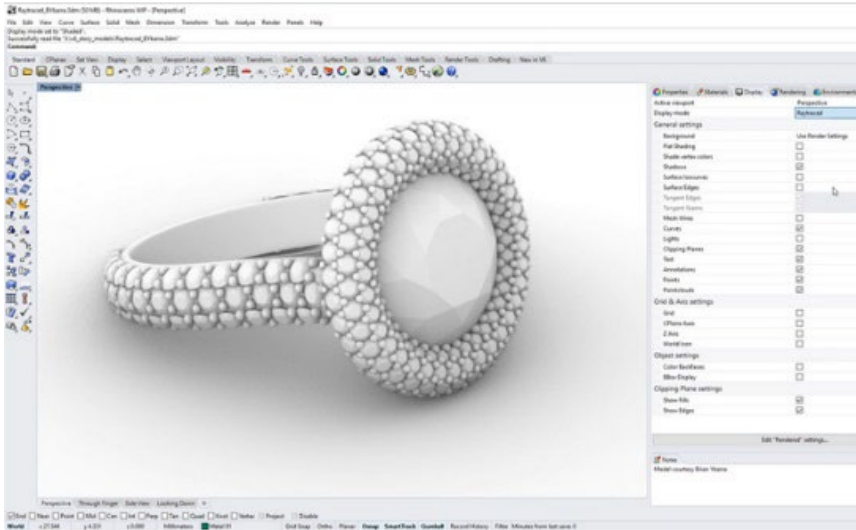


Рисунок 7.7

На цьому етапі параметричне проєктування почало сприйматися вже не просто як технічний інструмент редагування форми, а як окрема стратегія цифрового формоутворення. У центрі уваги опинився не одиничний геометричний результат, а система правил, що може породжувати множину варіантів. Це суттєво змінило логіку проєктної діяльності: замість ручного коригування кожного елемента проєктувальник отримав можливість керувати загальною моделлю через зміну ключових параметрів.

У сфері архітектури та дизайну це сприяло формуванню нових підходів до роботи з формою, структурою та адаптивністю об'єктів. Параметричні методи почали застосовувати для створення оболонок складної геометрії, фасадних систем, повторюваних модульних композицій, варіативних просторових структур, елементів інтер'єру, меблів і цифрових арт-об'єктів. Важливим стало й те, що параметричне проєктування виявилось корисним не лише для отримання виразних пластичних форм, а й для оптимізації, серійної варіативності, автоматизації змін та узгодження



складних залежностей між частинами моделі.

У сучасному розумінні історія параметричного проектування є історією поступового переходу від статичної геометрії до динамічно керованої моделі. Якщо на ранніх етапах цифрове проектування було зосереджене на відтворенні форми, то надалі більшого значення набували зв'язки, правила, обмеження, сценарії трансформації та алгоритми побудови. Саме ця еволюція створила підґрунтя не лише для класичного параметричного моделювання в CAD/BIM-середовищах, а й для розвитку дискретних, модульних і воксельних форм параметризації.

На рис. 7.8 наведено приклад архітектурно-дизайнерського об'єкта, створеного із застосуванням параметричного підходу, що ілюструє перехід параметричного мислення від інструментального рівня до повноцінної проєктної практики.



Рисунок 7.8

Отже, історія розвитку параметричного проектування демонструє, що його становлення відбувалося на перетині кількох ліній розвитку: математичного опису форми,



комп'ютерної графіки, автоматизованого креслення, асоціативного моделювання та алгоритмічного проектування. Параметричний підхід поступово вийшов за межі суто неперервного геометричного моделювання і став основою для нових способів цифрового формоутворення, зокрема дискретно-параметричного та воксельного.

7.3 Основні підходи та методи параметричного моделювання

Параметричне моделювання охоплює не один універсальний спосіб побудови форми, а кілька підходів, які відрізняються способом задання змінних, логікою зв'язків між елементами та характером керування геометрією. У практиці цифрового проектування найчастіше використовують моделі, засновані на розмірних і геометричних обмеженнях, змінних і формулах, ієрархії параметрів у сімействах або компонентах, конфігураціях і варіантах виконання, а також на візуальному алгоритмічному моделюванні.

Одним із базових підходів є параметризація на основі розмірних і геометричних обмежень. У такому випадку керування формою здійснюється через задання розмірів, паралельності, перпендикулярності, концентричності, фіксованих відстаней, кутів та інших умов, які визначають допустиму геометричну поведінку елементів (рис. 7.9).

Іншим важливим методом є параметризація через змінні, формули та рівняння. У цьому підході модель визначається не тільки геометричними залежностями, а й математичними виразами, які задають числові співвідношення між її параметрами.



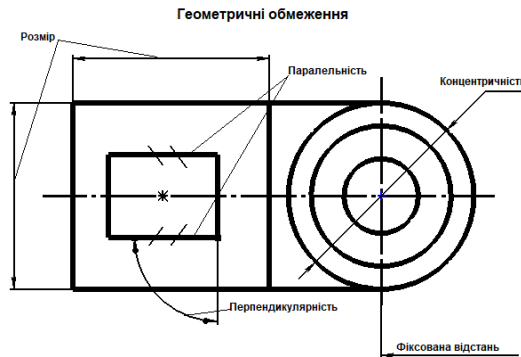


Рисунок 7.9

Наприклад, висота може визначатися як функція ширини, крок розташування елементів – як частка від загальної довжини, а допоміжні параметри можуть обчислюватися автоматично на основі введених значень (рис. 7.10).

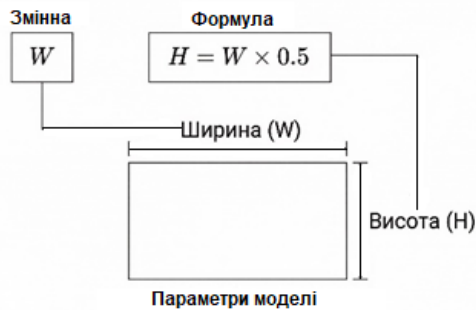


Рисунок 7.10

Для об'єктно-орієнтованого моделювання характерним є ієрархічний підхід, за якого параметри організуються в межах компонентів, сімейств або типів. У такій моделі частина параметрів є базовими, інші – похідними; одні керують габаритами, інші – видимістю, конфігурацією, матеріалом чи поведінкою окремих елементів. Перевага цього підходу полягає в тому, що параметрична логіка закладається не лише



в окрему форму, а в цілий клас об'єктів, які можуть мати багато варіантів виконання.

Окрему групу становить конфігураційне параметричне моделювання, коли користувач не змінює модель довільно, а обирає один із наперед передбачених наборів параметрів або варіантів. Такий підхід особливо корисний для виробів, модульних систем, повторюваних елементів і типових деталей, де потрібно швидко отримувати різні версії одного об'єкта без побудови кожної моделі з нуля.

Найбільш гнучким і водночас концептуально важливим напрямом є візуальне алгоритмічне параметричне моделювання, у якому геометрія створюється через граф зв'язків між вузлами, операціями, списками даних і параметрами (рис. 7.11).

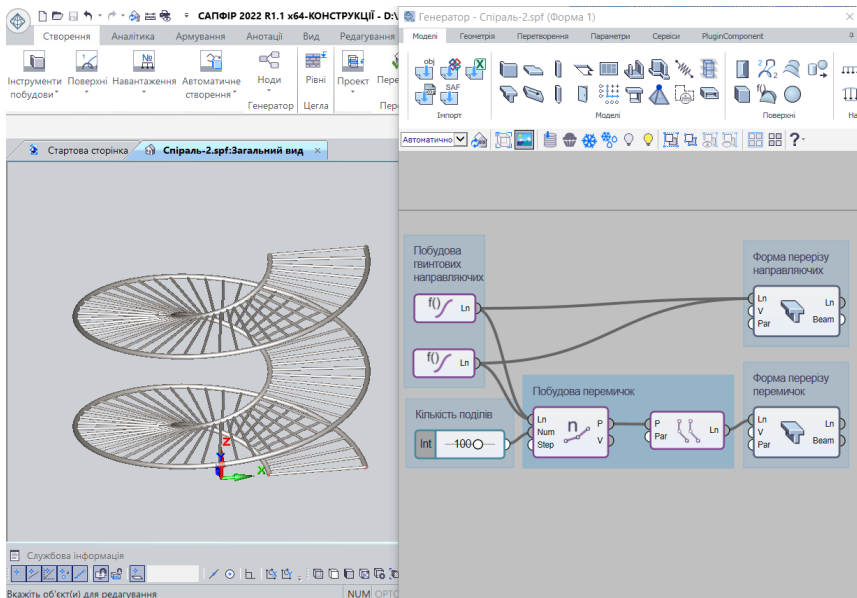


Рисунок 7.11

У такому середовищі проєктувальник задає не готову форму, а послідовність дій і залежностей, що породжують



форму. Це робить алгоритмічний підхід особливо придатним для складних поверхонь, варіативних структур, повторюваних систем та експериментального цифрового формоутворення.

У практиці проектування наведені підходи рідко існують ізольовано. Часто вони поєднуються: базова геометрія може керуватися через обмеження, окремі розміри – через формули, типологічні варіанти – через конфігурації, а складні залежності між частинами моделі – через візуальний алгоритм. Тому доцільно говорити не про один метод параметричного моделювання, а про систему взаємодоповнювальних методів, вибір яких залежить від характеру задачі, рівня складності форми, потреби у варіативності та середовища моделювання.

Разом із розширенням параметричних можливостей зростає і складність моделі. Велика кількість змінних, формул, обмежень, конфігурацій і похідних залежностей потребує чіткої організації параметричної структури. Тому ефективність параметричного моделювання визначається не лише інструментами середовища, а й продуманістю логіки побудови моделі, вибором базових параметрів і послідовністю формування залежностей.

Засвоєння основ параметричного моделювання має відбуватися послідовно: спочатку – через розуміння параметрів і обмежень, далі – через формули та залежності, потім – через варіативність і конфігурації, і лише після цього – через алгоритмічні схеми. Така логіка є корисною для подальшого переходу до дискретно-параметричного та воксельного моделювання, де параметризація також ґрунтується на правилах, але реалізується вже в іншому, дискретному середовищі.

7.4 Сфери застосування параметричного підходу

Параметричний підхід сьогодні належить до найважливіших методологій цифрового проектування, оскільки дає змогу описувати форму не лише як завершений



геометричний результат, а як наслідок дії змінних, обмежень, залежностей і правил побудови. Саме тому сфери його застосування є надзвичайно широкими: від архітектурного формоутворення і дизайну до інженерного моделювання, цифрового виробництва та міжплатформної інтеграції проєктних середовищ.

Однією з найважливіших сфер застосування параметричного підходу є архітектурне формоутворення. У цій сфері параметричне моделювання використовується для створення складних геометричних оболонок, варіативних об'ємів, пластичних поверхонь, просторових конструкцій і пов'язаних систем елементів. Його перевага полягає в тому, що архітектор може керувати не лише зовнішнім виглядом форми, а й логікою її зміни, адаптуючи модель до функціональних, композиційних або конструктивних вимог (рис. 7.12).



Рисунок 7.12

Окрему групу становлять адаптивні фасадні системи та оболонки, у яких параметричний підхід поєднується з кліматичними, світлотехнічними й енергетичними чинниками.



У таких випадках параметричне моделювання використовується не лише для побудови форми, а й для опису реакції елементів на зовнішні умови. Показовим прикладом є фасадна система Al Bahar Towers, у якій параметричний опис геометрії рухомих панелей застосовувався для моделювання їхньої роботи залежно від сонячного опромінення і кутів падіння світла протягом року (рис. 7.13).



Рисунок 7.13

Ще однією важливою сферою є проектування повторюваних, модульних і варіативних систем. Параметричний підхід особливо ефективний тоді, коли потрібно керувати великою кількістю подібних елементів, змінювати їхній крок, кількість, масштаб, орієнтацію або конфігурацію без ручного редагування кожного компонента. Саме тому він широко використовується для фасадних сіток, структурних оболонок, просторових каркасів, інтер'єрних систем, меблів, декоративних елементів і цифрових прототипів (рис. 7.14).





Рисунок 7.14

У сфері інженерного та міждисциплінарного проектування параметричний підхід особливо цінний тоді, коли потрібно пов'язати форму з розрахунком, категоризацією елементів, подальшим інформаційним моделюванням або документацією.

Параметричне проектування також активно застосовується у цифровому виробництві та виготовленні. Коли форма задана через параметри й правила, її значно простіше модифікувати під конкретні технологічні умови, матеріал, спосіб виготовлення чи обмеження обладнання. У таких випадках параметрична модель може бути проміжною ланкою між задумом, аналізом, оптимізацією та виробничою реалізацією.

У дизайні параметричний підхід використовується для варіативного формоутворення виробів, інтер'єрних елементів, меблів, освітлювальних об'єктів і цифрових арт-композицій. Його цінність полягає в можливості швидко переходити від однієї версії об'єкта до іншої, не втрачаючи логіки побудови моделі. Це дозволяє досліджувати серії альтернатив, добирати пропорції, ритміку, густоту структури та характер членування форми.



Поширення параметричного підходу в зазначених сферах пояснюється тим, що він забезпечує керованість змін, підтримку варіативності, узгодженість взаємопов'язаних елементів і зручність інтеграції з аналізом, інформаційним моделюванням та цифровим виробництвом. Саме тому параметричне проєктування стало не лише способом формоутворення, а й важливим елементом сучасного цифрового проєктного процесу.

Важливо також зазначити, що сфери застосування параметричного підходу не обмежуються лише середовищами неперервного геометричного моделювання. Хоча класичні приклади найчастіше пов'язані з гладкими поверхнями, фасадами або адаптивними оболонками, сама логіка параметризації – тобто опис форми через змінні, зв'язки і правила – може бути перенесена і в дискретне, модульне або воксельне середовище.

7.5 Параметричне, алгоритмічне та генеративне проєктування

Параметричне, алгоритмічне та генеративне проєктування є спорідненими, але не тотожними підходами до цифрового формоутворення. Їх об'єднує те, що форма в усіх трьох випадках розглядається не як одноразово побудований об'єкт, а як результат дії змінних, правил, зв'язків і процедур. Водночас між ними існують суттєві відмінності, пов'язані з роллю проєктувальника, способом задання логіки моделі та ступенем автоматизації пошуку варіантів.

Параметричне проєктування є найзагальнішим із цих підходів. У ньому форма описується через параметри, обмеження, формули та залежності, а зміна одного або кількох вхідних значень викликає перебудову моделі відповідно до наперед визначеної логіки. У такому випадку проєктувальник задає систему керування формою, а не лише її кінцевий вигляд.

Алгоритмічне проєктування можна розглядати як



розвиток параметричного підходу, у якому центральною стає не лише система параметрів, а й послідовність дій, що породжує форму. Якщо в класичному параметричному моделюванні основний акцент робиться на змінних і залежностях між елементами, то в алгоритмічному підході форма описується як результат виконання певної процедури або набору операцій. У середовищі САПФІР-3D така логіка реалізується через нодову схему побудови, у якій задані напрямні та твірні використовуються для формування кінематичної поверхні як результату послідовності параметрично пов'язаних операцій (рис. 7.15).

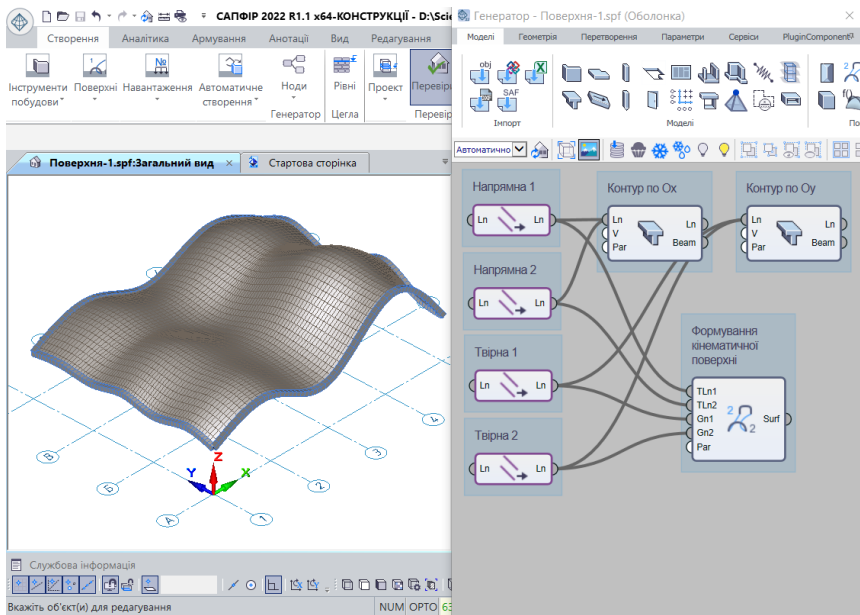


Рисунок 7.15

Таким чином, параметричне й алгоритмічне проєктування не є взаємовиключними. Навпаки, алгоритмічний підхід часто виступає способом реалізації параметричного мислення в тих випадках, коли модель є складною, багатокроковою або залежить від опрацювання



масивів даних, повторюваних структур чи варіативних процедур. Іншими словами, параметричне проектування відповідає на питання, які змінні керують формою, а алгоритмічне – яким способом ця форма послідовно утворюється.

Ще вищий рівень автоматизації властивий генеративному проектуванню. У цьому підході проектувальник задає не конкретну форму і навіть не лише алгоритм її побудови, а насамперед вихідні умови, критерії, обмеження та цілі, після чого система автоматично генерує велику кількість можливих рішень (рис. 7.16). Якщо параметричне моделювання дозволяє опрацьовувати варіанти, то генеративне проектування спеціально орієнтоване на масове автоматизоване породження й порівняння альтернатив.

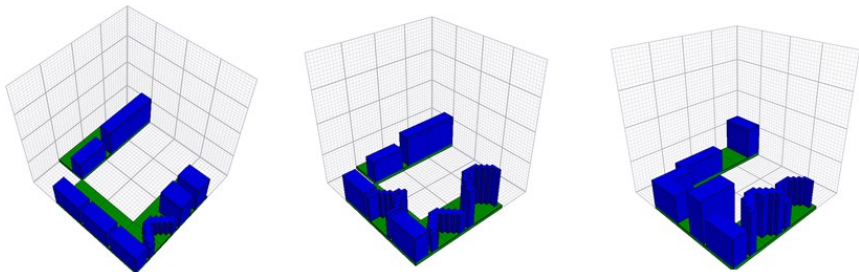


Рисунок 7.16

Принципова відмінність генеративного проектування від параметричного полягає в тому, що в параметричному підході проектувальник зазвичай сам керує змінами моделі, послідовно змінюючи параметри або правила, тоді як у генеративному підході система сама створює множину альтернатив у межах заданих умов. Саме тому генеративний підхід часто використовується тоді, коли потрібно не просто змінити форму, а знайти ефективні варіанти за багатьма критеріями одночасно.

Разом із тим ці три підходи доцільно розглядати не як ізольовані категорії, а як послідовний спектр методів



цифрового проєктування. Параметричне проєктування створює базову логіку залежностей; алгоритмічне проєктування розширює її, перетворюючи модель на послідовність процедур і операцій; генеративне проєктування піднімає процес ще на один рівень, де головним стає автоматичне породження й відбір варіантів за заданими критеріями.

Саме обмеження класичного параметричного моделювання – зокрема залежність від попередньо вибудованої логіки, історії побудови та складних ієрархій залежностей – частково пояснюють розвиток алгоритмічних і генеративних підходів, які дають інші механізми організації й опрацювання форми. У реальній практиці ці підходи часто поєднуються: параметрична модель може будуватися алгоритмічно, а генеративний процес – використовувати вже створені параметричні та алгоритмічні залежності як основу для автоматизованого пошуку рішень.

Принципово важливо, що всі три підходи базуються на спільній ідеї: форма має бути не лише змодельованою, а й описаною через правила її утворення. Саме ця логіка є ключовою для подальшого переходу до дискретно-параметричного та воксельного проєктування. Хоча воксельне середовище відрізняється від класичних систем неперервного моделювання, сама ідея керування формою через параметри, алгоритми та правила залишається спільною.



8 ВОКСЕЛЬНЕ МОДЕЛЮВАННЯ ЯК ОСНОВА ДИСКРЕТНО-ПАРАМЕТРИЧНОГО ДИЗАЙНУ

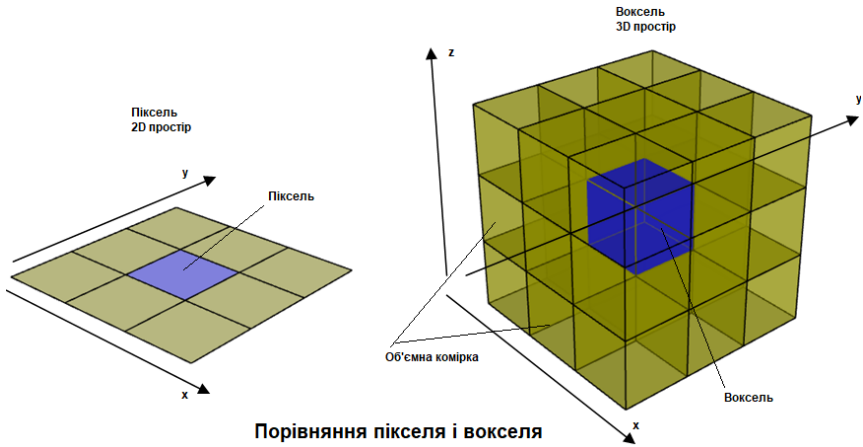
8.1 Воксель як елементарна одиниця дискретного цифрового простору

У контексті дискретно-параметричного дизайну вихідним поняттям є воксель – елементарна об’ємна одиниця цифрового простору, що займає певне положення у регулярній тривимірній сітці. Якщо піксель є мінімальним елементом двовимірного растрового зображення, то воксель можна розглядати як його просторовий аналог, тобто як об’ємний піксель, пов’язаний уже не з площиною, а з об’ємом. Саме через вокселі цифровий простір набуває властивостей подільності, адресованості та структурованості.

На відміну від пікселя, який визначається положенням у площині за координатами x та y , воксель існує у системі координат $x-y-z$, що дає змогу репрезентувати не лише контур або поверхню, а й глибину, внутрішню будову та просторове положення елементів у межах об’єкта (рис. 8.1). У цьому сенсі воксель є не просто засобом візуального подання, а коміркою цифрового середовища, якій може бути приписане певне значення: колір, щільність, матеріал, температура, прозорість або інший параметр.

У більш загальному розумінні воксель слід розглядати як комірку просторової дискретизації, яка може належати об’єкту або не належати йому, а також містити множину атрибутів. Саме тому воксельне представлення застосовується там, де необхідно працювати не тільки з зовнішньою межею форми, а з її повним об’ємом: у медичній томографії, науковій візуалізації, цифровому аналізі середовища, об’ємному моделюванні та у воксельному дизайні.





Порівняння пікселя і вокселя

Рисунок 8.1

Принциповою є також та обставина, що воксель не існує ізольовано. Його положення визначається місцем у регулярній тривимірній сітці, а отже, будь-яка воксельна модель є впорядкованою множиною просторових комірок, пов'язаних координатними відношеннями (рис. 8.2). Це робить воксельне середовище придатним до алгоритмізації, параметричного керування, підрахунку, модифікації та просторового варіювання.

Суттєвою властивістю воксельного простору є його дискретність. Якщо у неперервному геометричному моделюванні форма описується через аналітичні поверхні, сплайни або полігональні оболонки, то у воксельному середовищі простір попередньо поділяється на комірки наперед заданого розміру. Унаслідок цього будь-яка форма розглядається як результат заповнення або незаповнення окремих елементів просторової сітки.

У теоретичному сенсі воксель поєднує геометричний і інформаційний аспекти. Геометричний аспект пов'язаний із просторовим положенням елемента, а інформаційний – із набором властивостей, що з ним асоціюються. Саме тому воксель можна інтерпретувати як найменшу керовану



одиницю цифрового простору, до якої потенційно може бути прив'язаний параметр. У подальшому це набуває особливого значення для дискретно-параметричного дизайну, де не лише вся форма, а й окремі просторові комірки можуть ставати носіями змінних, умов і правил.

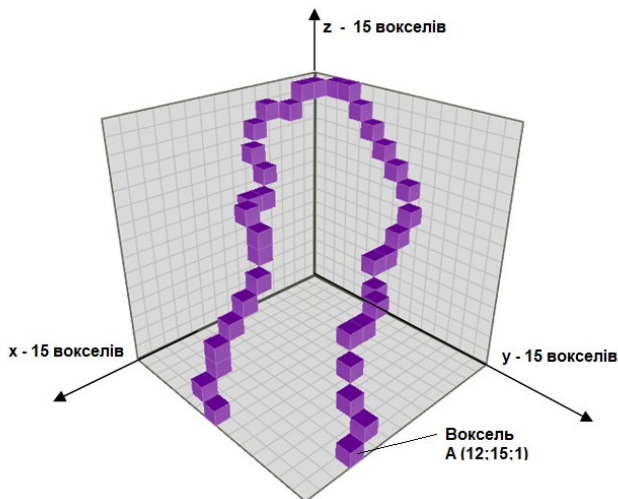


Рисунок 8.2

Ще однією важливою властивістю вокселя є масштабна залежність. Розмір комірки визначає точність відтворення форми: чим менший воксель, тим вищий рівень деталізації, але водночас тим більшими стають кількість елементів, обсяг даних і складність обробки. Отже, воксельне моделювання завжди передбачає компроміс між детальністю, наочністю та обчислювальною ефективністю.

Таким чином, воксель слід розглядати як елементарну дискретну комірку тривимірного цифрового середовища, що поєднує просторову локалізацію, інформаційне наповнення та потенціал параметричного керування. Саме через воксель цифровий простір стає структурованим, модульним і алгоритмічно керованим, а тому він є фундаментальною основою дискретно-параметричного дизайну.



8.2 Воксельне представлення форми та його відмінність від полігонального моделювання

Воксельне представлення форми ґрунтується на поданні об'єкта не через вершини, ребра і грані, а через множину елементарних об'ємних комірок, розмішених у регулярній тривимірній сітці. У такому підході форма описується як заповнена або частково заповнена область дискретного простору. Кожен воксель фіксує належність певної комірки до об'єкта та, за потреби, містить додаткові атрибути – колір, матеріал, щільність, прозорість або інші властивості.

На відміну від цього, полігональне моделювання подає об'єкт переважно через його зовнішню оболонку, сформовану множиною поверхневих елементів, найчастіше трикутників або чотирикутників. Геометрія такої моделі визначається координатами вершин і топологічними зв'язками між ними. Саме тому полігональне подання є особливо ефективним для опису гладких поверхонь, контурів і криволінійних оболонок (рис. 8.3).

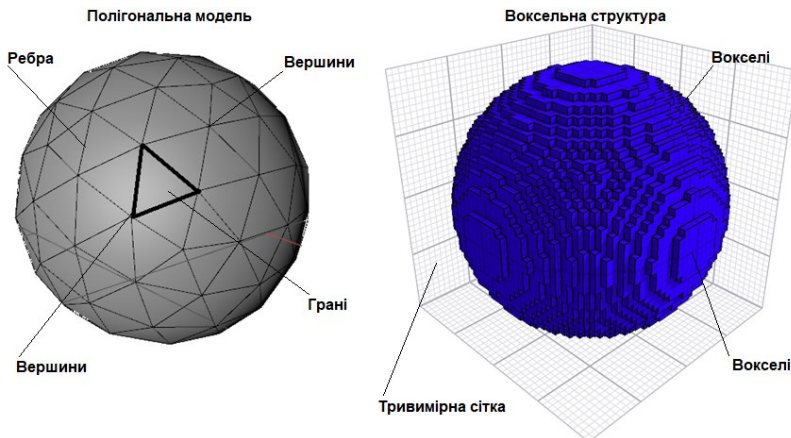


Рисунок 8.3

Полігональна модель історично стала основою більшості комп'ютерних систем проектування, оскільки вона дає змогу



відносно економно описувати візуально складні поверхні за допомогою сітки граней. Водночас така модель переважно репрезентує саме межу форми, а не її внутрішній об'єм. Якщо ж необхідно аналізувати внутрішню структуру об'єкта, неоднорідності матеріалу, порожнини, просторові шари або виконувати об'ємні обчислювальні операції, полігональне подання потребує додаткових процедур.

У воксельному середовищі ситуація інша. Тут форма від початку подається як об'ємна присутність у сітці комірок, а отже, зручно працювати не лише із зовнішньою поверхнею, а й із внутрішнім наповненням. Це відкриває можливість аналізувати локальні властивості середовища, проводити відбір окремих зон, формувати складні внутрішні структури, пористість, розподіл щільності та інші параметри, які у поверхневих моделях подаються опосередковано.

Однією з головних переваг полігонального моделювання є ефективність опису гладких і геометрично складних оболонок. За відносно невеликої кількості граней така модель може бути достатньо точною, добре піддається згладжуванню, текстуруванню, візуалізації та інженерному редагуванню. Саме тому полігональна геометрія зручна там, де пріоритетом є безперервність поверхні та візуальна плавність форми.

Натомість воксельне представлення має перевагу там, де важливими є структурність, модульність і керованість просторового заповнення. Завдяки регулярності сітки кожен елемент форми має чітку адресу, а тому може бути включений у процедури підрахунку, фільтрації, заміни, росту, ерозії, кластеризації чи параметричного варіювання. Форма в такому підході сприймається не як оболонка, а як організована множина дискретних одиниць.

Водночас воксельне моделювання має і свої обмеження. Через дискретний характер простору межі форми часто набувають ступінчастого характеру, особливо за великого розміру комірки. Підвищення точності можливе шляхом



зменшення розміру вокселя, але це призводить до зростання кількості елементів, обсягу даних і навантаження на систему.

Цей аспект ілюструє приклад вокселізації полігональної форми (рис. 8.4), у якому показано, як гладка оболонка переходить у дискретне об'ємне подання.

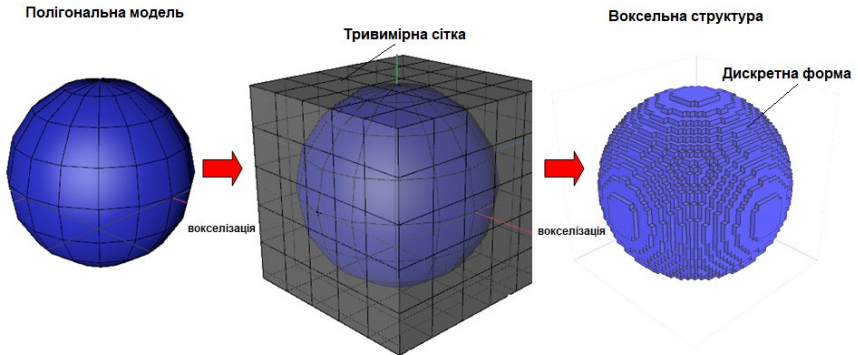


Рисунок 8.4

У сучасних цифрових практиках воксельне і полігональне подання не є взаємовиключаючими. Навпаки, між ними часто існує перехідний зв'язок: полігональні моделі можуть вокселізуватися для подальшого аналізу, симуляції чи дискретного формоутворення, а воксельні дані можуть перетворюватися назад у поверхневі оболонки для візуалізації або виробництва. Це означає, що воксельне й полігональне подання слід розглядати не лише як альтернативи, а і як взаємодоповнювальні режими цифрового моделювання.

Таким чином, воксельне представлення форми принципово відрізняється від полігонального тим, що описує об'єкт не через поверхневу межу, а через дискретно організований об'єм. Якщо полігональна модель є ефективнішою для точного подання гладких оболонок, то воксельна створює кращі умови для роботи з внутрішньою структурою, просторовою модульністю та алгоритмічним керуванням формою.



8.3 Дискретність як принцип цифрового формоутворення

У цифровому проектуванні дискретність слід розуміти не лише як технічну властивість подання даних, а як принцип організації форми, за якого складний об'єкт утворюється з множини окремих, просторово локалізованих елементів. У воксельному середовищі така логіка реалізується особливо послідовно, оскільки простір від початку поділено на регулярні комірки, а форма виникає як результат їх заповнення, вилучення, групування, нашарування або алгоритмічного відбору.

На відміну від неперервної геометрії, де форма зазвичай виглядає як плавна поверхня, що може змінюватися безперервною деформацією, у дискретному середовищі зміна форми відбувається покрово. Кожна модифікація реалізується через додавання, переміщення, заміну або вилучення окремих елементів чи їхніх груп. Унаслідок цього форма набуває властивостей модульності, адресованості, комбінаторності та алгоритмічної керованості.

У дискретному формоутворенні особливого значення набуває операційна логіка роботи з формою. Якщо в неперервному моделюванні проектувальник здебільшого працює з вершинами, ребрами, контрольними точками або поверхнями, то у воксельному середовищі основними діями стають заповнення комірок, просторовий ріст, відбір, кластеризація, модульне комбінування та структурні перетворення. У цьому значенні форма постає як результат алгоритму, який діє на множину дискретних елементів.

З позицій теорії формоутворення дискретність змінює сам характер побудови форми. Вона переводить увагу з безперервного моделювання оболонки на структурне збирання форми з елементів, що мають фіксоване положення, розмір і потенційний набір параметрів. Це означає, що виразність об'єкта може визначатися не лише контуром, а й щільністю



заповнення, частотою повторення, ритмом модулів, характером переходів між зонами та ступенем неоднорідності структури.

Однією з ключових переваг дискретного підходу є можливість безпосередньо інтегрувати у форму дані та правила. Оскільки кожна комірка має точну просторову адресу, до неї може бути прив'язане значення, пов'язане з матеріалом, функцією, щільністю, умовами середовища, конструктивними чи екологічними обмеженнями. У такому разі форма не просто моделюється, а обчислюється як результат застосування критеріїв до просторової матриці.

Це особливо наочно проявляється у випадках, коли об'єм генерується шляхом вилучення або відбору комірок за певними умовами (рис. 8.5).

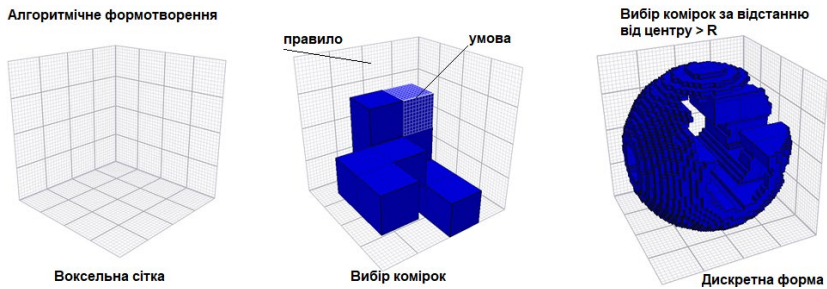


Рисунок 8.5

Ще одним важливим наслідком дискретного підходу є те, що він природно підтримує серійність і варіативність. Оскільки форма складається з керованих одиниць, зміна одного параметра або правила може призвести до появи нової конфігурації без повної перебудови моделі. Це дає змогу легко отримувати сімейства варіантів, досліджувати просторові переходи та формувати поля можливих станів.

Водночас дискретність не слід зводити лише до ступінчастості чи кубічності. Її сутність полягає у тому, що форма описується як скінченна множина елементів із чіткими адресами й станами, а тому підлягає лічбі, порівнянню,



автоматизованому редагуванню та правилам композиційної організації. Навіть коли на основі воксельних даних пізніше відновлюється гладка поверхня, первинна логіка формоутворення все одно залишається дискретною.

Отже, у воксельному моделюванні дискретність виступає фундаментальним принципом цифрового формоутворення, оскільки саме вона задає спосіб поділу простору, характер побудови форми, логіку її модифікації та можливість параметричного керування окремими елементами.

8.4 Воксельне представлення параметричних моделей: від неперервної до дискретної параметрики

Параметричне моделювання у класичному розумінні ґрунтується на тому, що форма визначається системою змінних параметрів і залежностей між геометричними елементами. У такому підході зміна значення параметра впливає на положення контрольних точок, контурів, поверхонь або вузлів моделі, але сама геометрія зазвичай зберігає неперервний характер. Воксельне представлення параметричних моделей переводить цю логіку в іншу площину: параметри починають впливати не лише на оболонку форми, а на склад, конфігурацію, щільність і розподіл дискретних просторових комірок.

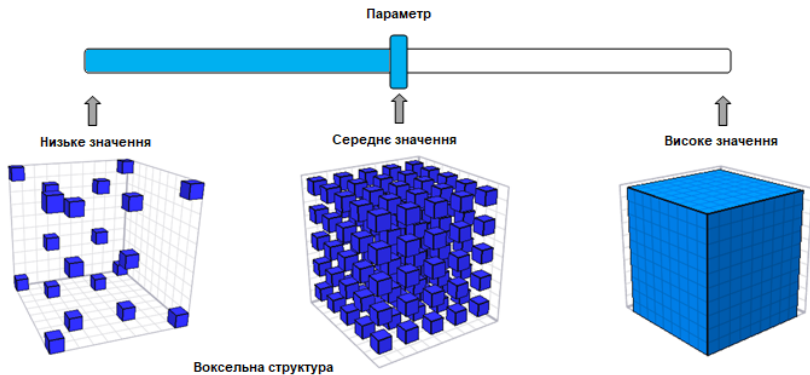
У неперервній параметриці форма переважно описується через математично безперервні об'єкти – криві, поверхні, сплайни, NURBS-оболонки або полігональні сітки, що апроксимують гладку геометрію. У дискретній параметриці простір поділяється на скінченну множину комірок, а параметр керує тим, які з них активуються, як групуються, де повторюються, де зникають і як змінюють свою локальну конфігурацію.

У практиці цифрового проектування перехід до воксельного представлення може здійснюватися кількома шляхами. Перший – вокселізація вже створеної параметричної



моделі, коли поверхнева геометрія перетворюється на просторову сітку заповнених комірок. Другий – первинне конструювання форми у воксельному середовищі, де параметри від початку керують дискретною структурою. Третій – гібридний підхід, за якого параметрична система задає загальні правила побудови, а воксельне середовище використовується для дискретизації, відбору, фільтрації та структурного варіювання.

Особливо показовою є ідея параметричного керування воксельною геометрією, у якій параметри впливають не на суцільну поверхню, а на поведінку воксельної системи – її ріст, локальні перетворення, закономірності групування і просторову організацію форми. У такому випадку геометрія формується як наслідок дії алгоритмів над дискретним середовищем. Варіювання воксельної структури під дією параметрів показано на рис. 8.6.



Параметричне варіювання

Рисунок 8.6

Порівнюючи неперервну і дискретну параметрику, важливо підкреслити, що вони відрізняються не лише технікою моделювання, а й типом змін, які допускає модель. У неперервному середовищі параметр зазвичай впливає на гладкі переходи, кривизну, контури, розміри та кути. У



дискретному середовищі параметр частіше пов'язаний із порогами, повторенням, щільністю, заповненням, кількістю активних комірок, шаблонами розміщення та правилами побудови.

У воксельному середовищі параметризувати можна не лише форму як таку, а й локальні властивості її елементів. Кожна комірка потенційно може містити значення, пов'язане з матеріалом, кольором, щільністю, функціональним статусом, конструктивною роллю або екологічним індикатором. Завдяки цьому параметрична модель у воксельному поданні може одночасно бути і геометричною, і інформаційною.

Ще одна суттєва риса дискретної параметрики полягає в тому, що вона природно підтримує серійне варіювання. Зміна одного параметра або набору правил може породжувати не один варіант форми, а ціле сімейство конфігурацій, які зберігають спільну логіку побудови, але відрізняються щільністю, масштабом, пористістю, членуванням або композиційним ритмом.

При цьому перехід від неперервної до дискретної параметрики не означає повної відмови від поверхневої геометрії. На практиці між цими режимами часто існує двосторонній обмін: неперервна модель вокселізується для аналізу або структурного формоутворення, а дискретне поле згодом може бути знову перетворене на поверхню. Відтворення поверхні з воксельного поля показано на рис. 8.7.

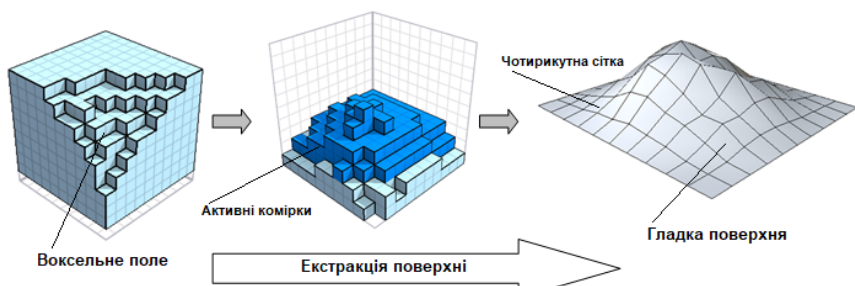


Рисунок 8.7



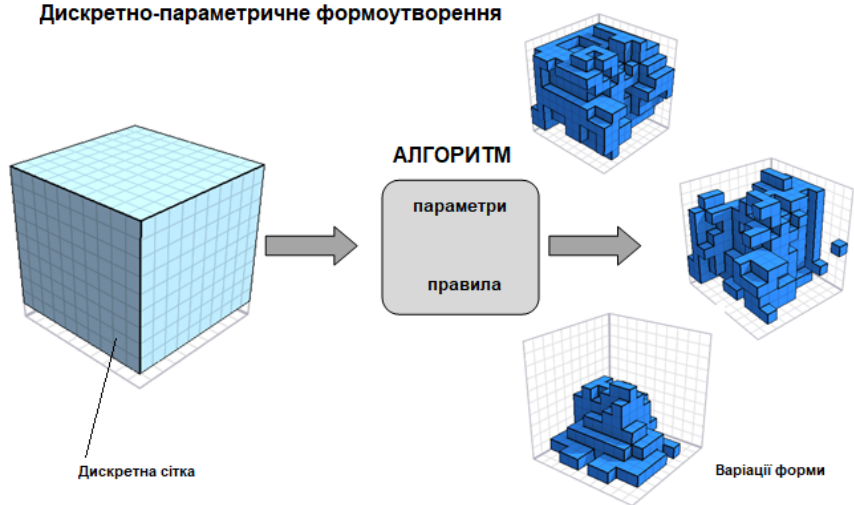
У цьому контексті воксельне представлення параметричних моделей є формою переходу від класичної неперервної параметрики до дискретної параметрики, у якій параметри керують не лише геометричною оболонкою, а структурою просторових комірок, їхніми властивостями та правилами організації. Саме ця логіка становить теоретичне підґрунтя для подальшого розгляду MagicaVoxel як інструмента дискретно-параметричного дизайну.

8.5 Дискретно-параметричний дизайн: сутність, ознаки та сфери застосування

Дискретно-параметричний дизайн доцільно розглядати як такий підхід до цифрового формоутворення, у якому параметричне керування поєднується з дискретною просторовою організацією форми. Якщо в класичному параметричному моделюванні зміна параметрів переважно впливає на неперервну геометрію – контури, поверхні, кривину, положення вузлів або контрольних точок, то в дискретно-параметричному підході параметри визначають склад, розподіл, щільність, повторення, комбінацію та трансформацію окремих елементарних комірок або модулів.

У межах цього підходу форма представляється не як наперед задана гладка оболонка, а як керована множина просторових одиниць, що можуть змінювати свій стан відповідно до параметрів, обмежень і алгоритмів. Це дає змогу працювати з формою на кількох рівнях одночасно: на рівні загальної конфігурації, на рівні локального компонента і на рівні внутрішньої просторової структури. Поєднання дискретної сітки, параметричних правил і варіативних результатів формоутворення подано на рис. 8.8.





Сутність дискретно-параметричного дизайну найкраще розкривається через його характерні ознаки. По-перше, це дискретність просторової організації, тобто подання форми як множини комірок, блоків або інших модульних елементів. По-друге, це параметрична керованість, коли структура змінюється не вручну, а через систему змінних, залежностей і правил. По-третє, це алгоритмічність, бо форма генерується або трансформується шляхом виконання процедур – повторення, порогового відбору, зростання, вилучення та об'єднання. По-четверте, це варіативність, оскільки одна й та сама система правил може породжувати множину допустимих конфігурацій. По-п'яте, це інформаційна насиченість, коли окрема комірка може містити не лише факт заповнення, а й значення матеріалу, щільності, функції чи іншого параметра.

У цьому значенні дискретно-параметричний дизайн займає проміжне, але самостійне місце між класичним параметричним моделюванням і суто воксельним або модульним моделюванням. Від параметризму він успадковує ідею керування формою через змінні та залежності. Від



дискретного моделювання – логіку побудови форми з окремих просторових елементів. Але результатом є не проста сума двох підходів, а нова проєктна парадигма, де дискретність стає середовищем реалізації параметрики, а параметрика – механізмом керування дискретною структурою.

Серед основних сфер застосування дискретно-параметричного дизайну передусім варто назвати архітектурне концептуальне моделювання. Тут дискретна просторово-параметрична логіка зручна для пошуку об'ємно-просторових рішень, варіювання забудови, роботи з обмеженнями ділянки, інсоляцією, висотними регламентами, пористістю об'єму та адаптивним членуванням фасадів. Воксельна репрезентація дозволяє швидко перебирати варіанти та оцінювати їх за набором параметрів.

Другою важливою сферою є урбаністика і просторове планування, де воксельні моделі використовуються для представлення тривимірних міських даних, аналізу компактності, багатозарових характеристик території, екологічних показників і сценаріїв розвитку середовища.

Третьою сферою є цифрове та генеративне мистецтво, ігрове середовище, воксельна стилістика і навчальний дизайн-експеримент, де особливого значення набувають ритм, модульність, серійність і композиційна варіативність.

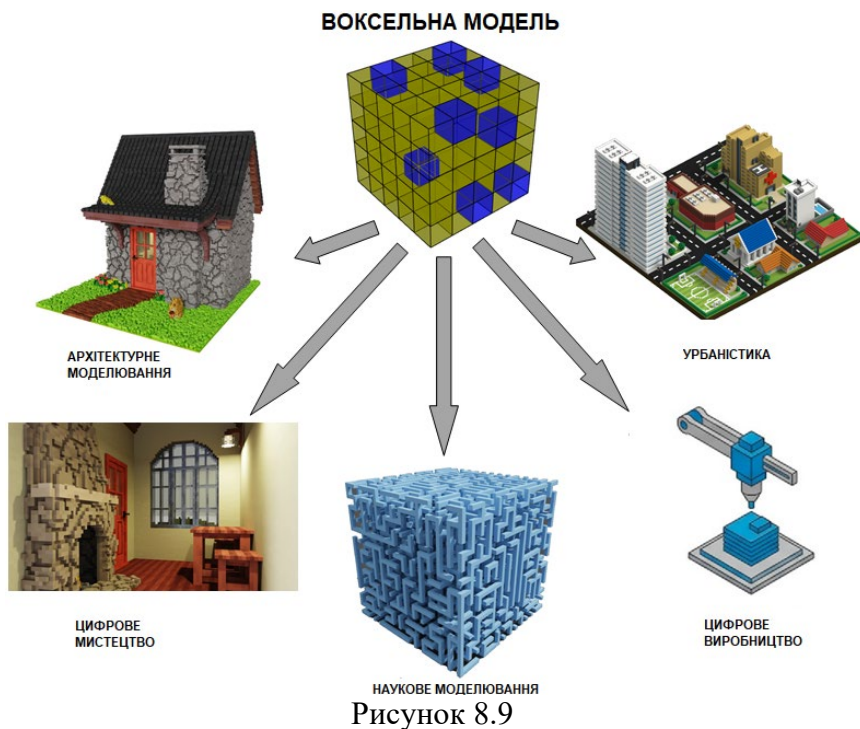
Четвертою сферою є цифрове виготовлення, модульні конструктивні системи та виробничі стратегії, де воксельна логіка може бути пов'язана з багаторазовими модулями, роботизованою збіркою та адитивними технологіями.

П'ятою – наукове та інженерне моделювання, де воксельні структури використовуються для зберігання просторових даних, симуляцій, аналізу фізичних полів і властивостей матеріалу. Широту сфер застосування воксельних моделей узагальнено показано на рис. 8.9.

Окремо слід наголосити, що для навчального процесу дискретно-параметричний дизайн має особливу цінність. Він



дозволяє здобувачам одночасно освоювати просторове мислення, логіку модульної побудови, базові принципи параметризації, алгоритмічне мислення та навички варіативного формотворення. На відміну від складних систем класичного параметризму, воксельне середовище часто дає більш наочний і швидкий зв'язок між дією, правилом і просторовим результатом.



Отже, дискретно-параметричний дизайн можна визначити як підхід до цифрового формоутворення, у якому параметричне керування реалізується через дискретні просторові елементи, а форма виникає як результат взаємодії комірок, параметрів, правил і даних. Його ключовими ознаками є дискретність, параметричність, алгоритмічність, варіативність, модульність та інформаційна насиченість.



9 ДИСКРЕТНО-ВОКСЕЛЬНИЙ ПАРАМЕТРИЧНИЙ ДИЗАЙН У СЕРЕДОВИЩІ MAGICAVOXEL

Дискретно-воксельний параметричний дизайн як новий підхід до дискретно-параметричного формоутворення дозволяє створювати різні варіанти об'єктів шляхом керованої трансформації їхньої форми.

Воксель як мінімальна об'ємна одиниця є принципово дискретним елементом, що відрізняє його від полігональних та NURBS-моделей. Як зазначалося у вступі, MagicaVoxel це програмний продукт, орієнтований на роботу із воксельними структурами. Він для більшості користувачів сприймається як інструмент для ігрової або ілюстративної графіки. Проте внутрішня логіка програми, дискретна природа та елементи алгоритмічної генерації форм роблять його придатним для дослідження додаткових можливостей параметричного дизайну у дискретному просторі.

Воксельний простір є тривимірною дискретною ґраткою, у якій кожен елемент має фіксовані координати та обмежений набір властивостей. На відміну від неперервних середовищ, воксельний простір не допускає нескінченно малих змін у геометрії елементів, а всі трансформації відбуваються кроками, кратними розміру вокселя. Це зумовлює специфічну естетику та особливу логіку процесів формоутворення. Кожна модель у MagicaVoxel є скінченною множиною елементів цього простору. Таким чином, будь-яка форма є результатом вибору певних координат відповідно до заданих правил або алгоритмів.

Дискретність воксельного простору має принципове значення для параметричного дизайну, оскільки параметри тут також мають дискретний характер. Це означає, що зміна параметра призводить не до плавної деформації, а до структурної перебудови форми, що відкриває нові можливості для аналізу та контролю геометрії об'єктів.

Параметр у воксельному дизайні можна визначити як



змінну, що впливає на:

- кількість вокселів у певному напрямку;
- конфігурацію просторової структури;
- щільність або пористість моделі;
- закономірності повторення та симетрії.

Таким чином, параметричний підхід у MagicaVoxel полягає у формалізації логіки побудови моделі, де кінцева форма є результатом дії набору правил, а не безпосереднього ручного ліплення.

MagicaVoxel не надає класичного інтерфейсу параметричного моделювання із пов'язаними змінними та залежностями. Однак параметричність у цьому середовищі можна реалізувати опосередковано – через спосіб мислення та організації роботи з моделлю.

Параметричне мислення у MagicaVoxel проявляється у використанні модульності, повторюваних операцій, симетрії, трансформації, масштабованості і т. і. Користувач фактично може проектувати не окрему форму, а алгоритм її створення, навіть якщо цей алгоритм реалізується вручну.

Важливим аспектом є також можливість серійного моделювання, коли одна і та сама логіка використовується для створення різних варіантів об'єкта шляхом зміни кількох ключових параметрів.

Теоретичні підстави дискретно-воксельного параметричного дизайну ґрунтуються на сукупності концепцій, пов'язаних із дискретними системами, алгоритмічним описом форми та формалізацією процесів формоутворення. На відміну від класичних геометричних підходів, де форма описується як неперервний об'єкт, воксельна параметрика оперує множинами, правилами та операторами.

Воксельну модель у MagicaVoxel можна інтерпретувати як скінченну множину елементів $V \subset \mathbb{Z}^3$, де кожен елемент має певний стан. Формоутворення у цьому випадку є процесом



визначення правил включення або виключення окремих елементів із цієї множини. Кожен воксель можна розглядати як просторову клітину, стан якої залежить від координат, глобальних параметрів або сусідніх клітин.

Алгоритмічне мислення користувача є ще однією фундаментальною складовою дискретно-воксельного параметричного проектування. Параметрична модель у MagicaVoxel фактично є алгоритмом, де параметри виконують роль вхідних змінних, а результатом є просторово організована воксельна структура. У цьому сенсі дизайнер, або просто користувач працює не з формою як такою, а з процедурою її побудови.

І як підсумок, дискретна параметрика підкреслює ідею того, що форма є результатом системи правил, а не унікальним об'єктом. Це зближує воксельний дизайн із системним підходом та сучасними теоріями генеративного мистецтва.

Використання параметричних функцій у MagicaVoxel відкриває широкі можливості для створення складних структур. Однією із ключових переваг є варіативність, що дозволяє отримувати серії моделей з єдиної логіки побудови. Параметричний підхід може забезпечити:

- систематизацію процесу моделювання;
- скорочення часу на внесення змін;
- підвищення концептуальної цілісності моделей;
- переходу від інтуїтивного до аналітичного

формоутворення.

Попри значний потенціал, параметричний дизайн у MagicaVoxel має низку обмежень. Відсутність вбудованих параметричних інтерфейсів ускладнює інтерактивну роботу з параметрами. Крім того, дискретна природа вокселя накладає обмеження на точність та гладкість форм. Разом із тим ці обмеження можуть розглядатися не лише як недолік, а як специфічна умова, що формує унікальну мову воксельного дизайну.



Для досягнення наукової обґрунтованості, параметричний дизайн у MagicaVoxel доцільно описувати через формалізовані функції. Узагальнену параметричну воксельну модель можна представити у вигляді функції:

$$F(x, y, z, p_1, p_2, p_3, \dots, p_n) \rightarrow (0, 1), \quad (9.1)$$

де (x, y, z) – координати вокселя,

(p_1, p_2, \dots, p_n) – набір параметрів, що визначають правила формоутворення.

Значення функції (9.1) визначає наявність або відсутність вокселя у заданій позиції. Такий підхід дозволяє описувати форму не як геометричний об'єкт, а як логічну умову. Наприклад, параметр може визначати радіус, висоту, щільність або періодичність структури. Зміна параметра призводить до перебудови всієї множини, а не до її локальної деформації.

Застосування параметричних принципів у MagicaVoxel охоплює широкий спектр завдань – від архітектурного концептуального моделювання до художніх і наукових експериментів. У кожному випадку параметри визначають логіку організації воксельної структури. Архітектурні моделі часто базуються на параметрах масштабу, ритму та повторення. Кількість поверхів, крок колон або розмір модульної сітки можуть задаватися як змінні, що дозволяє швидко генерувати серії просторових проектних рішень.

Модульні структури у MagicaVoxel використовують параметри повторення, масштабування, віддзеркалення кольору, трансляції та інше. Простий воксельний модуль, заданий один раз, може формувати складні решітки або просторові каркаси шляхом параметричного тиражування.

Органічні та ландшафтні форми базуються на параметричних функціях шуму. Параметри кольору, частоти та масштабу визначають характер рельєфу або рослинної структури, що робить можливим дослідження природоподібних форм у дискретному середовищі.



Фрактальні структури є окремим класом параметричних моделей. Рекурсивні правила дозволяють створювати складні ієрархічні форми, де параметри визначають глибину рекурсії та масштаб елементів.

Порівняльний аналіз дискретної параметрики, реалізованої у воксельному середовищі MagicaVoxel, та неперервної параметрики, характерної для систем, наприклад SolidWorks або Houdini, дозволяє чітко окреслити методологічні відмінності цих підходів.

Неперервні системи ґрунтуються на аналітичних кривих, поверхнях і чисельних методах апроксимації, що забезпечує високу геометричну точність і плавність форм. Натомість дискретна параметрика у MagicaVoxel працює із множинами, логічними умовами та цілими значеннями параметрів. Тут форма не апроксимується, а конструюється безпосередньо із елементів просторової ґратки. Це забезпечує повний контроль над структурою та внутрішньою організацією об'єкта, але водночас накладає обмеження на гладкість і масштаб деталізації.

Обидва підходи не слід розглядати як взаємовиключні. Вони швидше доповнюють один одного, формуючи різні стратегії цифрового формоутворення.

Естетика воксельної параметрики ґрунтується на прийнятті дискретності як художньої якості. Ступінчастість, модульність і ритмічна повторюваність воксельних структур формують особливу візуальну мову, відмінну від гладкої естетики неперервних поверхонь.

Параметричний підхід дозволяє працювати з цією естетикою свідомо, перетворюючи обмеження воксельного простору на засіб художнього вираження. Серійність, варіативність і структурна логіка стають основними композиційними інструментами.

Таким чином, воксельна параметрика може розглядатися не лише як технічний метод, а як окрема естетична парадигма



цифрового мистецтва і дизайну у MagicaVoxel. На початкових етапах функціонал параметрики у програмі можна подати наступними складовими:

1. Моделювання по сітці з умовами. Наприклад, можна створювати дискретну структуру, де кожен воксель ставиться або не ставиться в залежності від положення (x, y, z) за певним правилом (парність координат, чи відстань від центра). Це вже можна вважати дискретним параметричним підходом.

2. Використання зовнішніх скриптів (шейдерів) або плагінів. MagicaVoxel підтримує імпорт моделей у форматі .vox, який можна генерувати поза програмою, наприклад, скриптом на Python. Там можна створювати параметричні сітки вокселів, а потім імпортувати їх у MagicaVoxel для візуалізації.

3. Модульні структури. Якщо зробити набір базових «модулів» із вокселів, а потім збирати з них складніші композиції, змінюючи параметри (розмір, кількість повторень, розташування), це теж свого роду параметричне дискретне моделювання.

Відповідно наведених вище принципів, сформуємо у MagicaVoxel початковий «псевдо-параметричний» конвеєр – використовуючи лише вбудовані інструменти та логіку, без зовнішнього софту. Мова йде про те, щоб розглядати воксельний простір як дискретну сітку з певними математичними правилами, і використовувати інструменти MagicaVoxel (Pattern, Brush, Attach/Erase, Repeat, Symmetry, Transform) як «параметричні перемикачі».

1. Патерни (Pattern) як параметри

MagicaVoxel дозволяє задавати форму інструмента «Brush» через «Pattern» (P). Можна вважати ці патерни «параметричними змінними», бо вони визначають, що і де малюється або стирається. Інструментами можуть бути:

- клітинки через 1 – це параметр, період сітки;
- градієнтні переходи по кольору чи висоті – це



параметр: поріг кольору або координата;

– циліндричні чи сферичні зони – це параметр: радіус або центр.

2. Генерація через симетрію

Інструмент «Symmetry» (x, y, z , Mirror) можна трактувати як параметр, що задає кількість та розташування копій елементів. Інструментами є:

– створення одного елемента, а симетрія відтворює його в потрібній кількості. Параметри: кількість осей симетрії, напрям, відстань між копіями;

– комбінування симетрії з Pattern – генерація складних фрактальних візерунків.

3. Repeat та Transform як «масштабні параметри»

MagicaVoxel включає «Repeat» (R) і «Transform» (T) – фактично інструменти повтору та масштабування. За допомогою них можливо:

– задавати базовий модуль (наприклад, куб $5 \times 5 \times 5$) і повторювати його по сітці. Параметр: кількість повторів по кожній осі;

– масштабування моделей як параметр зміни дискретного розміру.

4. Параметризація через палітру

Колір у вокселях – теж параметр, який можна використати в алгоритмах побудови. Інструменти дають можливість:

– прив'язати колір до висоти (наприклад, чим вище, тим світліше);

– використовувати палітру для позначення «шарів» або «зон» моделі, щоб легше змінювати їх потім.

5. Алгоритмічне видалення та побудова

Комбінація «Attach» + «Erase» за патернами може виконувати роль умовного алгоритму. Шлях параметричного моделювання:

– спочатку заповнити об'єм, потім «висвердли»



форму за маскою (Pattern);

– створювати порожнини по певних координатах (наприклад, тільки там, де $(x+y)$ – парне число).

Систему «псевдо-параметричних» сценаріїв можна подати базовим набором дій:

1. Створюється базова форма (модуль).
2. Визначається набір керуючих параметрів:
 - розмір модуля;
 - кількість повторів;
 - симетрія по осях;
 - тип патерну для заповнення/видалення;
 - прив'язка кольору до координат.
3. Змінюються наведені параметри і отримуємо нові варіанти форми.

Система визначає наступні базові парадигми:

1. Чітке визначення параметрики як логіки, а не інтерфейсу. Вводиться поняття параметричного мислення.

2. Акцент на дискретності як цінності. Дискретний параметр змінює множину вокселів, а не деформує форму. Це фундаментальна відмінність від NURBS-параметрики.

3. Формалізація через $F(x, y, z, p_1, p_2, p_3, \dots, p_n)$. Це дозволяє:

- пов'язати художню практику із математикою;
- легко створювати різновидності моделей типу «умова-форма».

Наведемо декілька прикладів використання можливостей дискретно-воксельного параметричного дизайну у MagicaVoxel.

Приклад 1. Вертикальна воксельна структура із параметрами висоти та щільності

У даному прикладі форма інтерпретується як результат дії двох параметрів: висоти та щільності заповнення. Зміна висоти реалізується через масштабування по осі Z , тоді як щільність структури контролюється через патерни (маску)



вибіркового заповнення або видалення вокселів. Внаслідок цього зміна параметрів призводить не до деформації об'єкта, а до повної перебудови його внутрішньої структури.

Параметри;

- P_1 – висота модуля (H);
- P_2 – крок заповнення (D) через патерн.

Крок 1. Базовий модуль

Створити нову сцену: «File» – «New». Встановити розмір сцени, наприклад: $32 \times 32 \times 64$. Інструмент «Attach», «Brush» = $1 \times 1 \times 1$. Створити один вертикальний стовп: координати – $X=16, Y=16, Z=40$ вокселів. Це базовий модуль, а не форма (рис. 9.1).

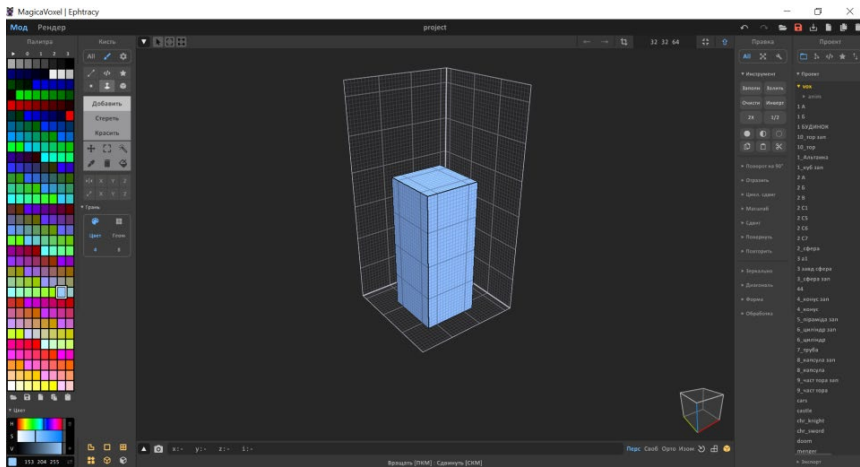


Рисунок 9.1

Крок 2. Параметр висоти (H)

Виділити модуль (Ctrl+A або Region Select). Використати інструмент «Transform» (T): «Scale» по осі Z. Змінити масштаб: 0.5 – низький модуль, 1.5 – високий модуль (рис. 9.2). Форма не змінюється вручну – змінюється значення параметра висоти H.



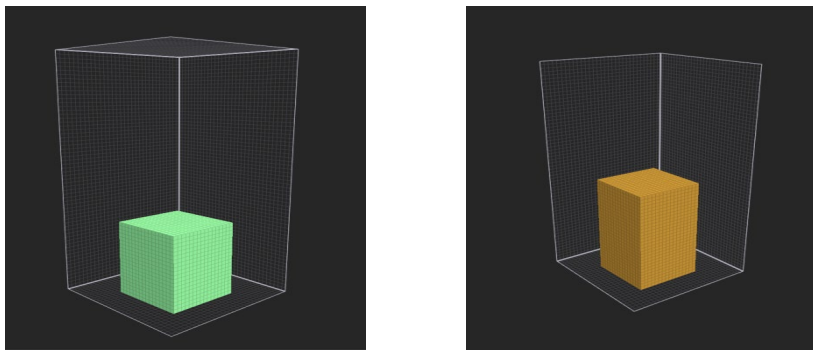


Рисунок 9.2

Крок 3. Параметр щільності D через Pattern

Вибрати патерни, наприклад, «шаховий» або «сферично-шаховий». Патерни із заданими параметрами клітин – створені заздалегідь. Включити інструмент «Erase», а патернами провести по висоті створених параметричних варіацій модулів.

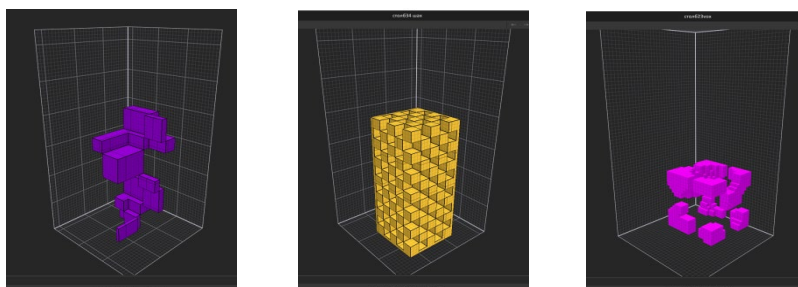


Рисунок 9.3

Результат: при виборі одного модуля – одна структура, при виборі наступного – пориста «решітка» з іншими геометричними параметрами. Це дискретна параметрика: параметр – період виключення вокселів, зміни проходять не плавно, а змінюють структурну будову моделей.

Отримаємо серію моделей за спільною логікою (рис. 9.3). Форма описується як $F(z, p_1, p_2)$, де p_1 – висота модулів, p_2 –



структурована щільність.

Приклад 2. Модульна фасадна структура з параметрами повторення та симетрії

Даний приклад демонструє параметричний підхід через модульність. Базовий фасадний елемент тиражується у просторі за допомогою інструмента «Repeat», де кількість повторів по кожній осі виступає параметром. Додатково використовується симетрія, яка задає правила розташування елементів у композиції. Таким чином фасад розглядається як система, а не як статичний об'єкт.

Параметри:

- P_1 – розмір модуля;
- P_2 – кількість повторів;
- P_3 – симетрія.

Крок 1. Створення фасадного модуля

Створити куб $10 \times 1 \times 19$. Всередині: інструментом «Erase» – сформувані отвори певної конфігурації. Це фасадна «клітинка», яка є алгоритмічним елементом.

Крок 2. «Repeat» як параметр

Виділити модуль. Інструментом «Repeat» (R) – виконати повтори: $X=6, Y=1, Z=6$. Змінити параметри: $X=2, Y=1, Z=2$ – щільний фасад, $X=20, Y=1, Z=10$ – протяжна структура. «Repeat» – параметр кількості модулів (рис. 9.4).

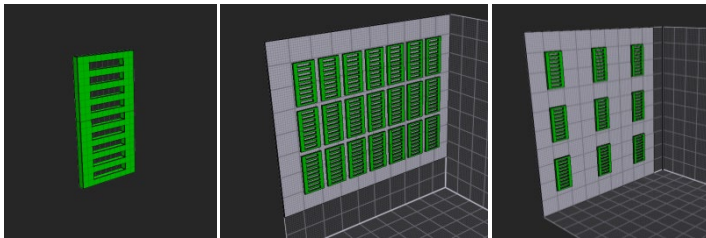


Рисунок 9.4

Крок 3. Параметр ритму

Увімкнути «Pattern». Обрати: «шаховий», видалити (Erase) деякі модулі. Результатом є: змінюється ритм фасаду



без ручного редагування.

Крок 4. Симетрія як параметр композиції

Увімкнути «Symmetry» X або Y. Додати один елемент (балкон або просто виступ). Він автоматично буде дублюватися (рис. 9.5). Тут симетрія – параметр розташування форми. Результатом є: один і той самий модуль дає можливість створити десятки фасадів. Змінюються числа, а не процедура формування варіантів «вручну».

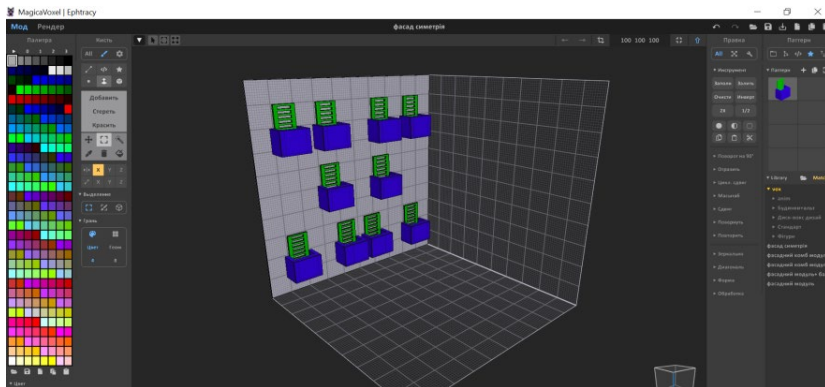


Рисунок 9.5

Приклад 3. Органічний рельєф як порогова параметрична структура

У даному прикладі реалізується ідея порогової параметрики. Початкова форма задається як суцільний об'єм, з якого шляхом алгоритмічного видалення вокселів формується рельєф. Параметрами виступають максимальна висота та поріг видалення, що визначає, які воксели залишаються у структурі. Такий підхід дозволяє створювати органічні форми у межах суворо дискретної логіки.

Цей приклад безпосередньо пов'язаний із формулою $F(x, y, z, p_1, p_2, p_3, \dots, p_n) \rightarrow (0, 1)$ і добре ілюструє поняття порогової параметрики.

Форма інтерпретується як дискретний ландшафт, де:

- X і Y – координати площини;



- Z – висота;
- параметр визначає, до якої висоти воксели зберігаються, а вище – видаляються.

Органічність досягається не складністю інструментів, а накладанням патернів, шейдерів або порогів.

Параметри:

- P_1 – максимальна висота рельєфу (H);
- P_2 – поріг видалення (T);
- P_3 – характер патерну (шум/ритм).

Крок 1. Створення суцільного об'єму

Створити нову сцену: «File» – «New». Вибрати розмір сцени, наприклад: $64 \times 64 \times 32$. Інструментом «Attach» заповнити повний паралелепіпед (суцільний блок).

Початкова форма – це повна множина V , з якої потрібна форма буде отримана шляхом видалення.

Крок 2. Формування «висотного поля»

Увімкнути шейдер, наприклад «градієнтний». Налаштуваннями шейдера слід поступово видаляти воксели (рис. 9.6).

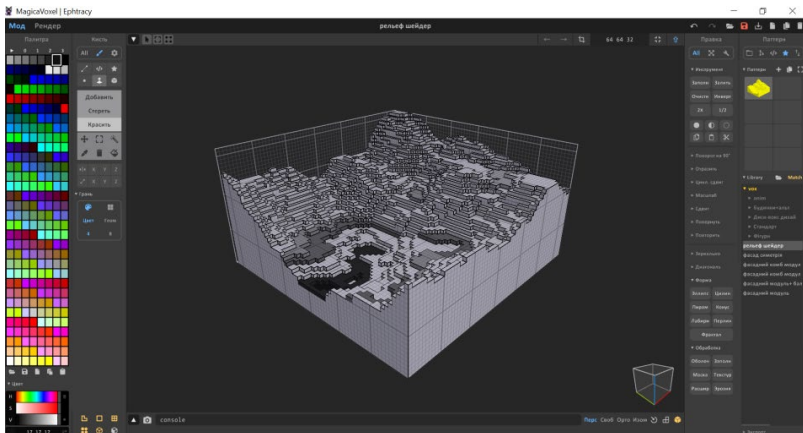


Рисунок 9.6

Результат: поверхня набуває нерівного, органічного характеру, але логіка побудови залишається параметричною.



Крок 3. Параметр порогу (Т)

Змінити інтенсивність шейдера, або кількість проходів «Erase». Спостерігається: що при низькому порозі – пологий рельєф, при високому – різкий, скелястий.

Дана Органічна форма у воксельному середовищі виникає як результат порогового відбору елементів, а не як наслідок ручного скульптингу. Таким чином, навіть ландшафт може бути параметрично описаний у дискретному просторі.

Можна піти складнішим шляхом. Створити параметричну модель, наприклад в SolidWorks (або будь-якій іншій CAD-системі з параметричним моделюванням) з усіма потрібними параметрами. Потім експортувати модель у формат, який можна перетворити у воксельний формат (або безпосередньо у вокселі), і завантажити її у MagicaVoxel для воксельного рендеру, редагування, додавання деталей тощо.

Основні етапи даного процесу складаються із наступних дій:

1. Параметрична модель у SolidWorks. Створюється модель із всіма необхідними параметрами: розмірами, кутами, кількістю повторів, складними поверхнями (рис. 9.7).

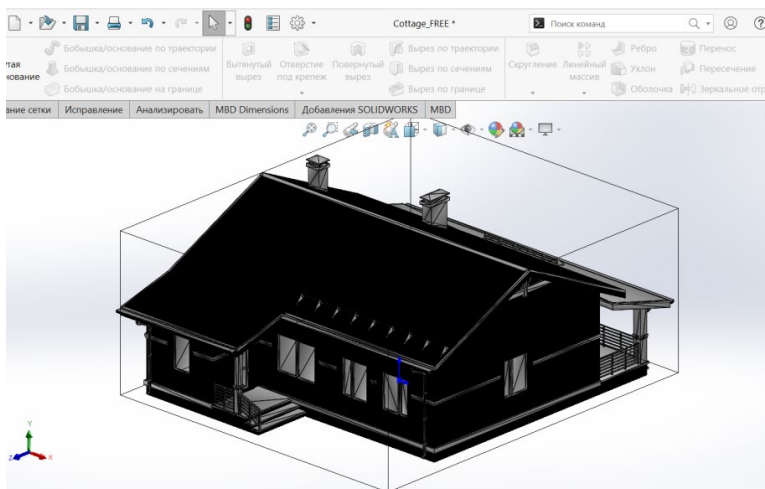


Рисунок 9.7



Експорт у STL або OBJ.

SolidWorks підтримує експорт у популярні формати для 3D-моделювання: STL, OBJ, 3MF.

2. Конвертація у вокселі.

MagicaVoxel не імпортує STL/OBJ напряму, але можна скористатися спеціальними конверторами (наприклад, Voxelizer, Qubicle, Binvox) (рис. 9.8).

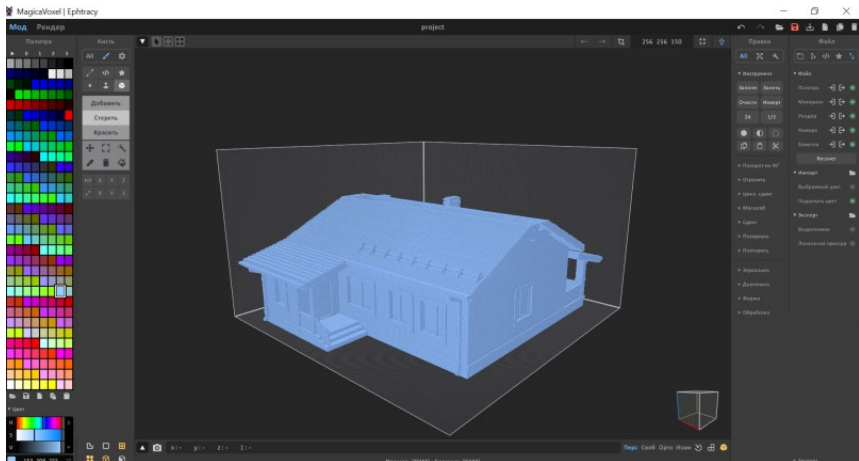


Рисунок 9.8

Ці програми чи утиліти перетворюють 3D-модель з триангульованою сіткою (mesh) у воксельний формат.

3. Імпорт у MagicaVoxel. Завантажте воксельну модель, яку можна редагувати, текстурувати, анімувати (рис. 9.9).

Перевагами такого підходу є:

- повноцінне параметричне проектування, яке дає точний контроль (рис. 9.10);
- воксельна візуалізація з можливістю додати шар художнього стилю, піксельного шарму;
- можливість швидко змінювати параметри в SolidWorks і генерувати нові версії в MagicaVoxel;



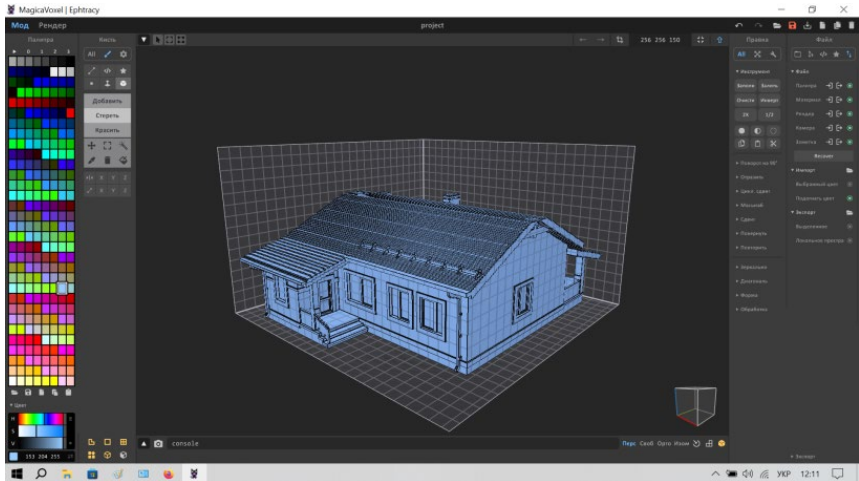


Рисунок 9.9

- створення параметричних архітектурних моделей із подальшим воксельним рендером для презентації (рис. 9.11);
- можливість проєктування ігрових сценаріїв, які виконуються в CAD-системах і потім адаптуються у воксельний стиль.

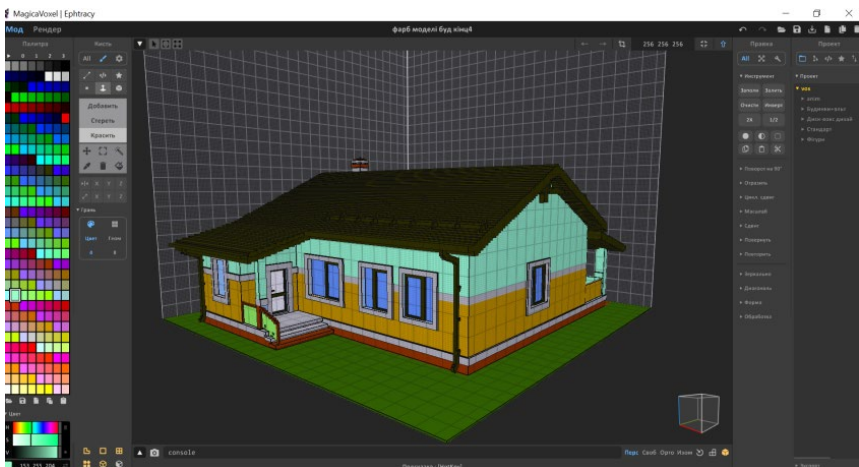


Рисунок 9.10



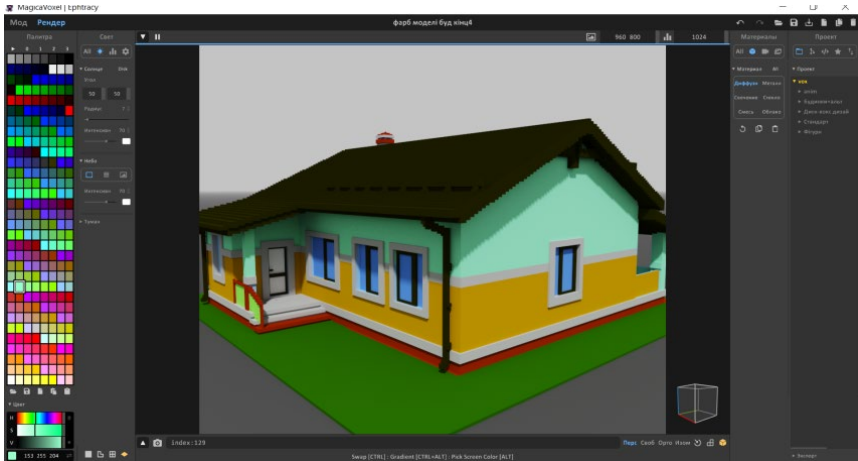


Рисунок 9.11

Подальший розвиток дискретно-воксельного параметричного дизайну пов'язаний із розширенням міждисциплінарних зв'язків. Він може ефективно використовуватися у таких сферах:

1. Комп'ютерна томографія (СТ) та магнітно-резонансна томографія (MRI). Тут медичні зображення формуються у вигляді воксельних сіток, що дозволяє будувати тривимірні моделі внутрішніх органів. Це надає оптимально можливості для: аналізу та візуалізації медичних сканів, виконання 3D-реконструкції органів для хірургічного планування, проведення автоматичної сегментації та ідентифікації аномалій (пухлин, переломів).

2. Обчислювальна механіка та симуляції. Воксельна графіка використовується для дискретизації континуальних середовищ, що дозволяє моделювати фізичні процеси в матеріалах. Відкриває можливості для моделювання деформацій твердих тіл, аналізу проходження рідин у пористих структурах (гідродинаміка, нафтогазова промисловість), воксельного представлення матеріалів для їх розрахунків методом кінцевих елементів.

3. Комп'ютерна графіка та ігрова індустрія. Воксельні



моделі застосовуються для створення складних середовищ і об'єктів у реальному часі. Це актуально для воксельних ігор, генерації проєктних світів, реалістичного руйнування об'єктів в процесі гри і т. д.

4. 3D-друк та адитивне виробництво. Воксельні моделі використовуються для точного представлення внутрішньої структури матеріалів. Це дозволяє: створювати складні пористі структури з оптимальними механічними властивостями, виконувати біодрук тканин та органів, проводити оптимізацію параметрів 3D-друку для різних матеріалів.

5. Гідродинамічне та газодинамічне моделювання. Воксельне представлення моделей дозволяє ефективно проводити чисельні симуляції потоків, аналізувати аеродинаміку автомобілів та літаків, вивчати процеси горіння у двигунах внутрішнього згорання.

6. Геологія та геофізика. Воксельна графіка використовується для представлення геологічних структур у вигляді воксельних сіток, для побудови підземних моделей із видобутку корисних копалин, аналізувати розповсюдження хвиль в сейсмології, моніторити зміни у ґрунтових структурах.

7. Робототехніка та навігація. Воксельні карти дозволяють ефективно моделювати середовище для створення карт руху автономних роботів, для аналізу траєкторій руху безпілотних апаратів.

8. Квантові обчислення та симуляція наноструктур. Воксельне представлення використовується для дискретизації квантових систем, моделювання поведінки електронів у наноструктурах, аналізу магнітних полів у квантових комп'ютерах.

У всіх, наведених вище практичних застосунках, ефективно можуть бути застосовані елементи дискретно-воксельного параметричного проєктування, у тому числі в середовищі MagicaVoxel.



Даний підхід активно формується як самостійний напрям параметричного цифрового формоутворення, як інструмент візуалізації, як повноцінна платформа для досліджень у галузі цифрового дизайну та дискретного моделювання. Параметричні функції у дискретному середовищі дозволяють описувати форму як результат системи правил, забезпечуючи варіативність, відтворюваність і аналітичну чіткість процесу проектування.



ПРАКТИКУМ



Практичне завдання 1

Тема: Знайомство з інтерфейсом MagicaVoxel та базовими інструментами «Attach», «Erase», «Paint».

Мета:

1. Засвоїти базові інструменти редагування:
 - «Attach» (A) – додавання вокселів;
 - «Erase» (E) – видалення вокселів;
 - «Paint» (P) – фарбування вокселів.
2. Створити серію «плоских» геометричних об'єктів.

Підготовка

1. Відкрити програму MagicaVoxel.
2. Створити нову сцену (меню «File» – «New» або Ctrl+N).
3. Робочий розмір сцени: 15×15×15.

Покрокова інструкція для роботи

1. Орієнтація у просторі (рис. 1.1).

- Обертання сцени – правою кнопкою миші.
- Наближати/віддаляти камеру слід колесиком миші.
- Панорамування (зсув сцени): середня кнопка миші або Shift+ПКМ.

2. Додавання вокселів (Attach) (рис. 1.2).

- Активувати інструмент «Attach» (A).
- У лівій панелі обрати форму (Brush): «Box».
- У палітрі кольорів обрати будь-який колір (наприклад блакитний).
 - Намалювати заданий об'єкт розміром 15×15×1 вокселів у центрі сцени (тягнути лівою кнопкою миші).



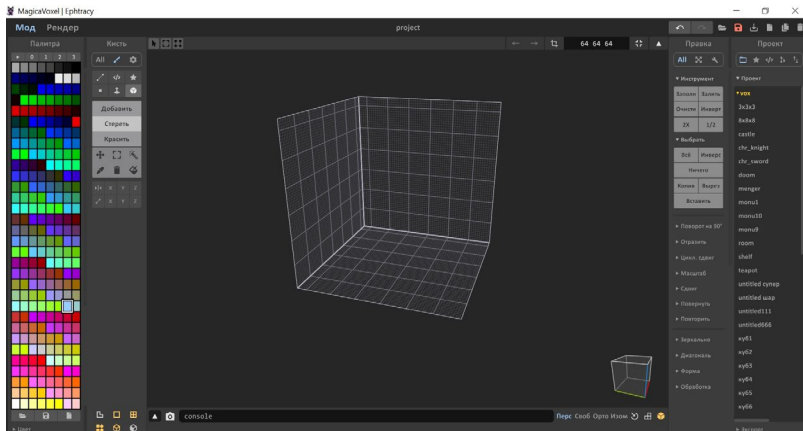


Рисунок 1.1 – Навігація у просторі сцени

3. Видалення вокселів (Erase) (рис. 1.3).

- Перемкнутись на інструмент «Erase» (E).
- У «Brush» обрати «Voxel» (один воксель).
- Клацнути по верхній чи боковій частині об'єкта та видалити кілька вокселів, щоб змоделювати потрібну форму.
- Спробувати «Brush» – «Box» і видалити більший шматок.

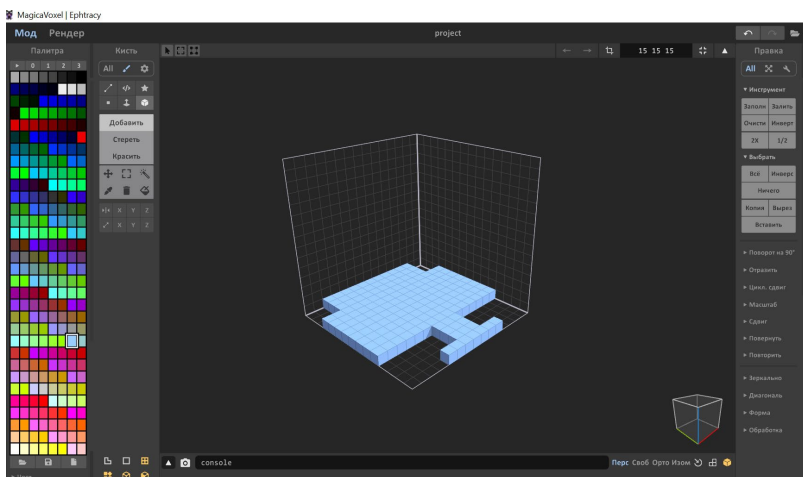


Рисунок 1.2 – Створення об'єкта інструментом Attach



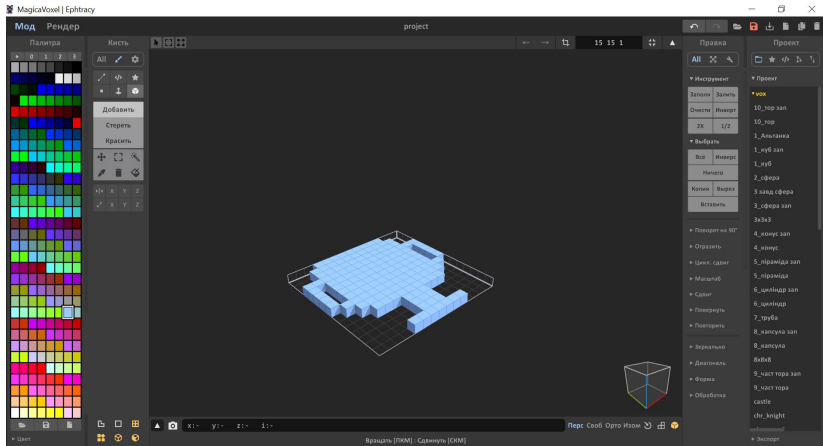


Рисунок 1.3 – Використання **Erase** для видалення частин об'єкта «Ваза» та обрізання сцени

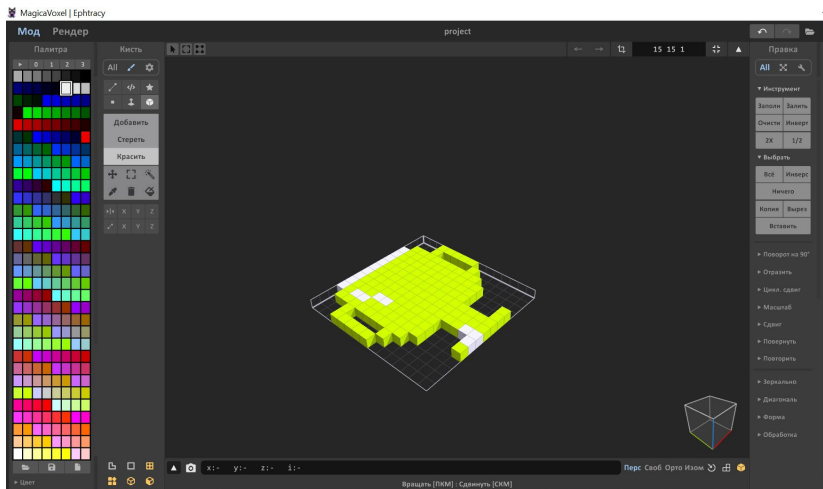


Рисунок 1.4 – Фарбування вокселів у режимі **Paint**

4. Фарбування вокселів (Paint) (рис. 1.4).

- Вибрати інструмент «Paint» (P).
- У палітрі обрати інший колір (наприклад, жовтий).
- Застосувати «Brush-Face» – клацнути по грані об'єкта, щоб пофарбувати його.



– Спробувати «Brush-Vox» – виділити й змінити колір деяких частин.

5. Збереження проєкту (рис. 1.5).

- Меню «File» – «Save As...» або Ctrl+S.
- Назвати файл: Ваза.vox.

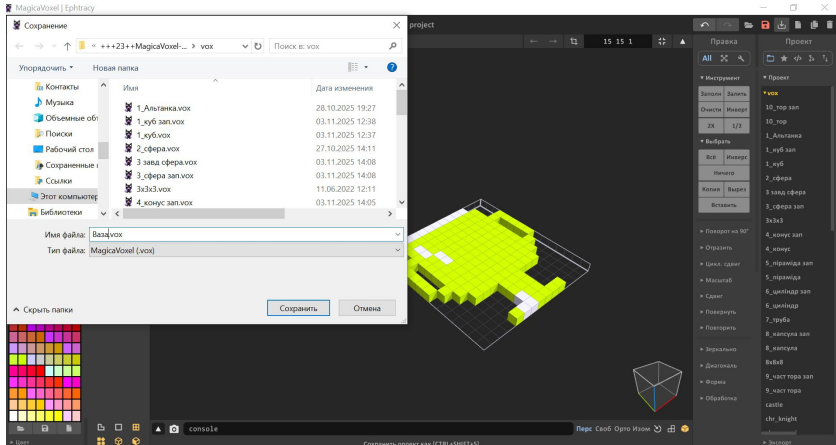


Рисунок 1.5 – Збереження файлу об'єкта «Ваза» у форматі .VOX

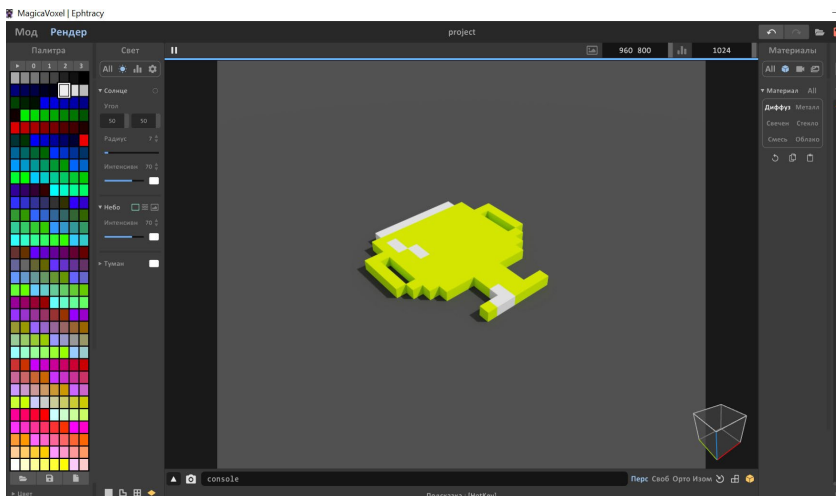


Рисунок 1.6 – Приклад рендерингу об'єкта «Ваза»



Виконати, поки без налаштувань, процедуру рендера сцени зі створеним об'єктом, активуючи команду «Рендер» у верхній стрічці (рис. 1.6).

Варіанти для вправи

Завдання: За заданими варіантами створити послідовно набір власних «плоских» геометричних форм у MagicaVoxel (рис. 1.7).



Рисунок 1.7 – Варіанти «плоских» геометричних об'єктів





Рисунок 1.8 – Серії воксельних геометричних фігур

Варіанти для завдання 1

№ варіанту	Номера рядків геометричних фігур
1	1, 7
2	2, 6
3	3, 5
4	4, 1
5	2, 3
6	2, 4
7	2, 5
8	2, 6
9	2, 7
10	3, 7
11	3, 6
12	3, 4



13	3, 5
14	3, 6
15	4, 7
16	4, 6
17	4, 5
18	5, 6
19	5, 7
20	1, 4

Послідовність виконання завдання

Після створення першого об'єкта, додати пусту сцену для створення наступного через активацію «+» зверху екрана (рис. 1.9). Моделюємо потрібний об'єкт і знову створюємо сцену.

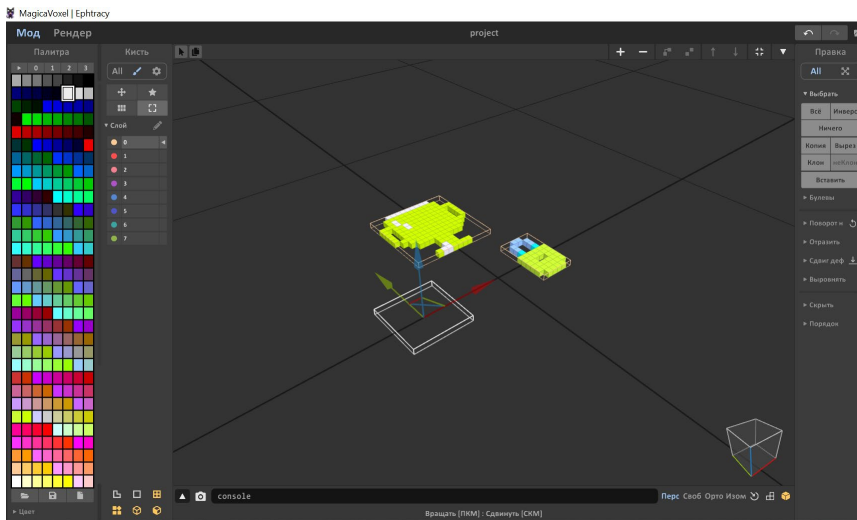


Рисунок 1.9 – Додавання наступної сцени

Приклад виконання частини завдання і звіту наведено на рис. 1.10.



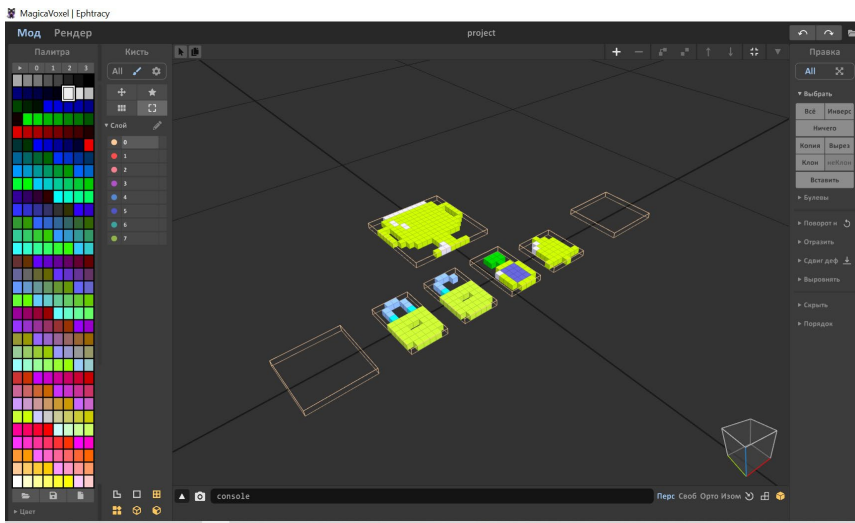


Рисунок 1.10 – Приклад формування серії об’єктів за варіантом

Практичне завдання 2

Тема: Знайомство з інтерфейсом MagicaVoxel та базовими інструментами «Attach», «Erase», «Paint» для створення 3D-моделей об’єктів. Створення модульних патернів («Pattern»).

Мета:

1. Навчитися орієнтуватися в інтерфейсі MagicaVoxel.
2. Засвоїти базові інструменти редагування:
 - «Attach» (A) – додавання вокселів;
 - «Erase» (E) – видалення вокселів;
 - «Paint» (P) – фарбування вокселів.
3. Створити серію 3D-моделей геометричних об’єктів та у вигляді патернів зберегти їх у програмі.

Підготовка

1. Відкрити програму MagicaVoxel.
2. Створити нову сцену (меню «File» – «New» або Ctrl+N).



3. Робочий розмір сцени: $64 \times 64 \times 64$.

Покрокова інструкція для роботи

1. Орієнтація у просторі (рис. 2.1).

- Обертання сцени – правою кнопкою миші.
- Наближати/віддаляти камеру слід колесиком миші.
- Панорамування (зсув сцени): середня кнопка миші або Shift+ПКМ.

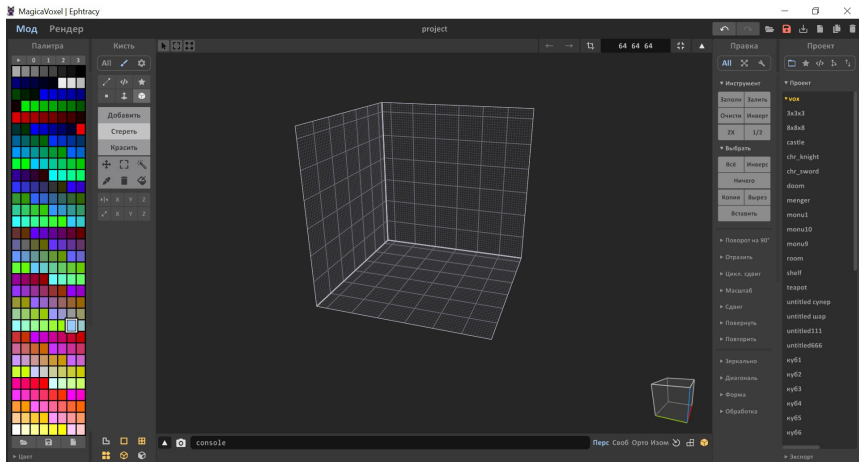


Рисунок 2.1 – Навігація у просторі сцени

2. Додавання вокселів (Attach) (рис. 2.2).

- Активувати інструмент «Attach» (A).
- У правій панелі обрати форму (Brush): «Вох».
- У палітрі кольорів обрати будь-який колір.
- Намалювати куб розміром $10 \times 10 \times 10$ вокселів у центрі сцени (тягни лівою кнопкою миші).

3. Видалення вокселів (Erase) (рис. 2.3).

- Переключитись на інструмент «Erase» (E).
- У «Brush» обрати Voxel (один воксель).
- Клацнути по верхній чи боковій частині куба та видалити кілька вокселів, щоб зробити заглиблення.
- Спробувати «Brush-Вох» і видалити більший шматок.



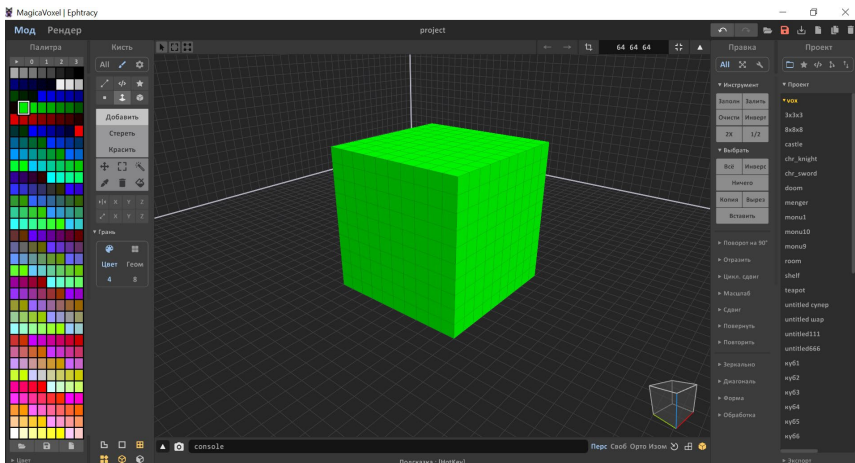


Рисунок 2.2 – Створення куба інструментом «Attach»

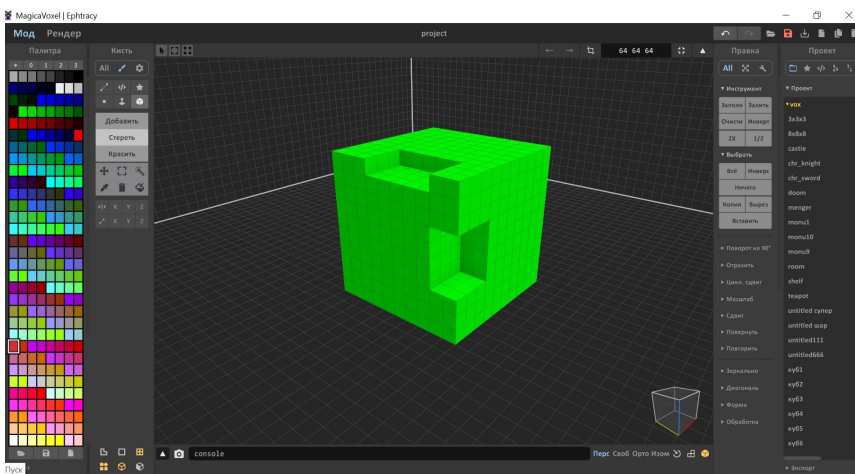


Рисунок 2.3 – Використання «Erase» для видалення частини куба

4. Фарбування вокселів (Paint) (рис. 2.4).

- Вибрати інструмент «Paint» (P).
- У палітрі обрати інший колір (наприклад, червоний).
- Застосувати «Brush-Face» – клацнути по грані куба, щоб пофарбувати її.



- Спробувати «Brush-Vox» – виділити й змінити колір усієї частини.

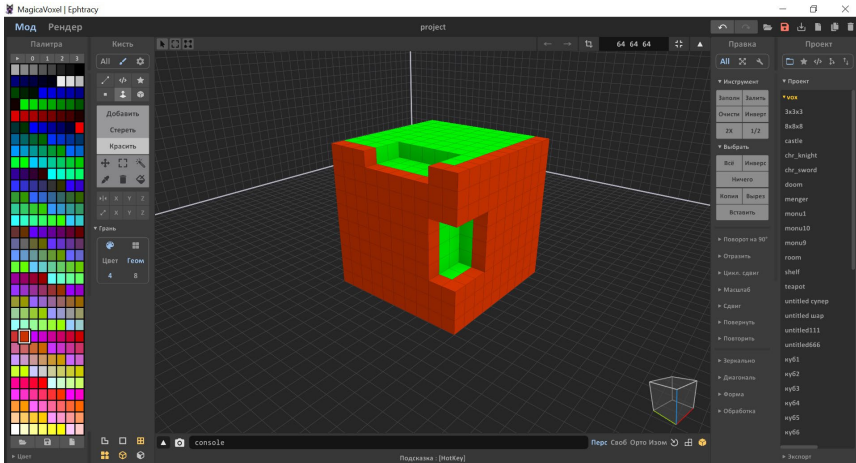


Рисунок 2.4 – Фарбування множини вокселів у режимі «Paint»

5. Збереження проєкту (рис. 2.5).

- Меню «File» – «Save As...» або Ctrl+S.
- Назвати файл: куб_1.vox.

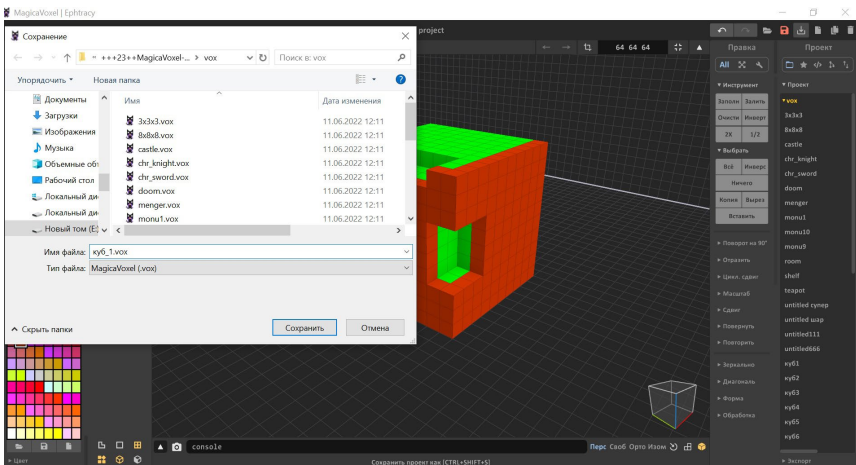


Рисунок 2.5 – Збереження файлу .vox



Варіанти для вправи

Завдання: За заданими варіантами створити послідовно кілька власних патернів геометричних форм у MagicaVoxel (рис. 2.6).

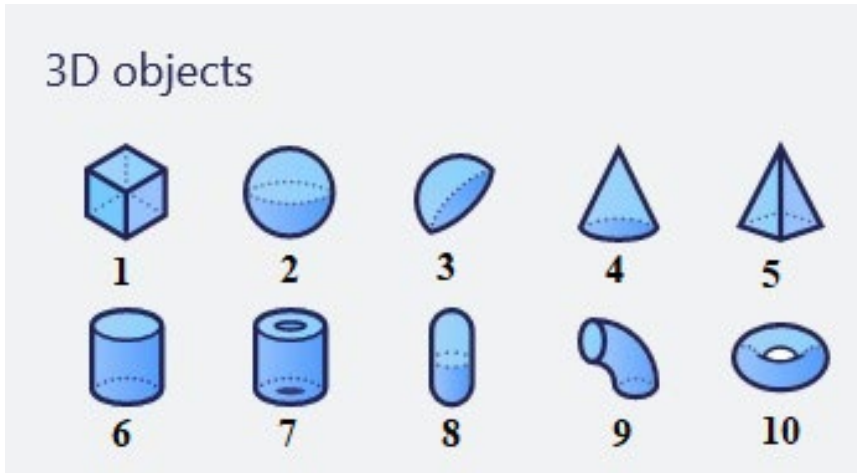


Рисунок 2.6 – Варіанти геометричних фігур

Варіанти для завдання 2

№ варіанту	Номера геометричних фігур	Колір фігур	Видалення $\frac{1}{4}$ частини фігур
1	2, 4, 5, 8, 10	червоний	4,5
2	1, 3, 6, 7, 9	синій	1,6
3	1, 4, 5, 6, 7	зелений	4,6
4	4, 5, 6, 9, 10	жовтий	4,10
5	2, 3, 6, 7, 9	фіолетовий	2,6
6	2, 3, 5, 8, 9	червоний	2,8
7	1, 3, 6, 8, 10	синій	6,8
8	1, 4, 5, 7, 8	зелений	4,8



9	2, 3, 6, 9, 10	жовтий	2,6
10	1, 4, 6, 7, 8	фіолетовий	1,4
11	1, 4, 7, 8, 9	червоний	4,8
12	1, 5, 6, 7, 8	синій	4,6
13	1, 2, 5, 8, 9	зелений	2,8
14	3, 5, 7, 9, 10	жовтий	3,7
15	2, 5, 6, 8, 9	фіолетовий	2,6
16	1, 4, 5, 7, 10	червоний	1,10
17	1, 3, 6, 8, 10	синій	1,6
18	2, 4, 6, 8, 9	зелений	2,4
19	3, 5, 7, 9, 10	жовтий	3,5
20	1, 5, 6, 8, 9	фіолетовий	1,8

Послідовність виконання завдання

У додатку Windows «Paint 3D» (3D shapes) відкрити меню 3D-фігур (рис. 2.7). Знайти потрібний об'єкт, наприклад конус (рис. 2.8) і зберегти його у форматі – .GLB (рис. 2.9).

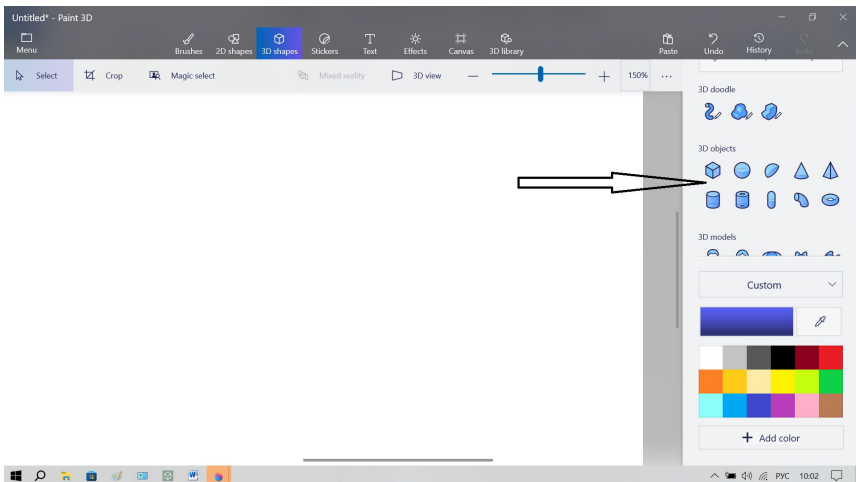


Рисунок 2.7 – Об'єкти додатку «Paint 3D»



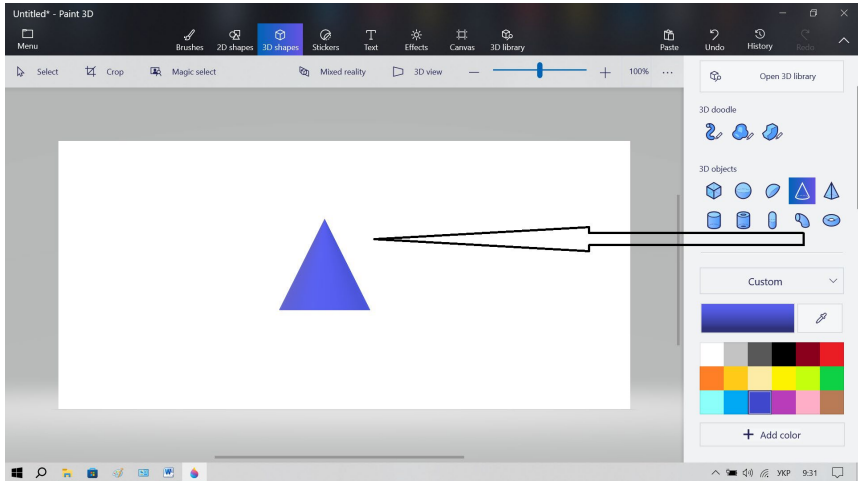


Рисунок 2.8 – Приклад створення моделі конуса у форматі .GLB

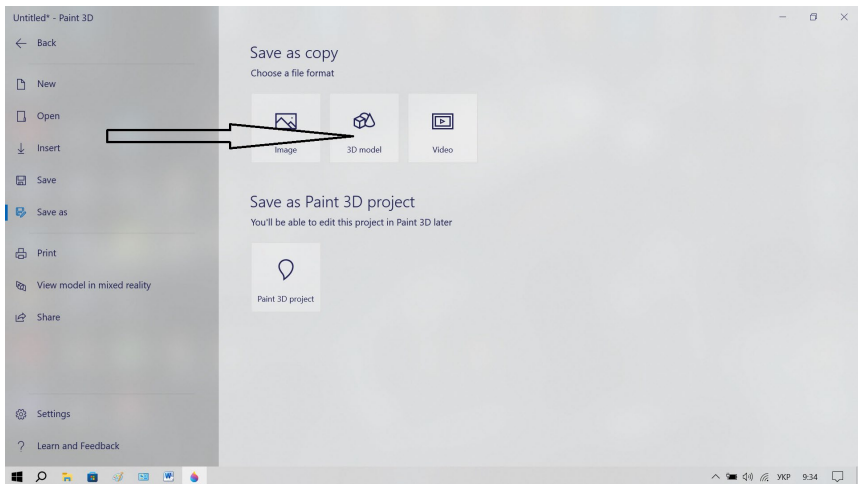


Рисунок 2.9 – Збереження 3D-моделі у форматі .GLB

Відкрити онлайн-конвертор «Image to STL» і конвертувати файл формату .GLB у формат .OBJ (рис. 2.10).



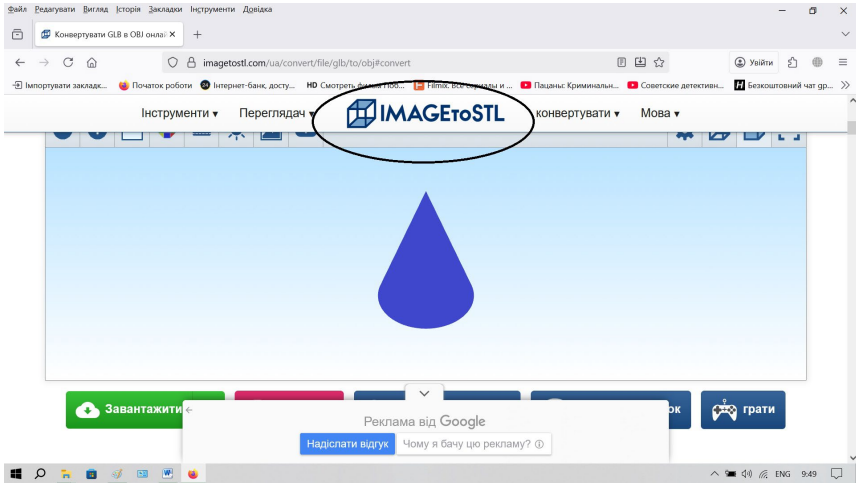


Рисунок 2.10 – Конвертування моделі в додатку «Image to STL»

Через функцію «імпорт» у MagicaVoxel відкрити об’єкт конус, редагувати його, за потреби, і зберегти як патерн у власній бібліотеці (рис. 2.11).

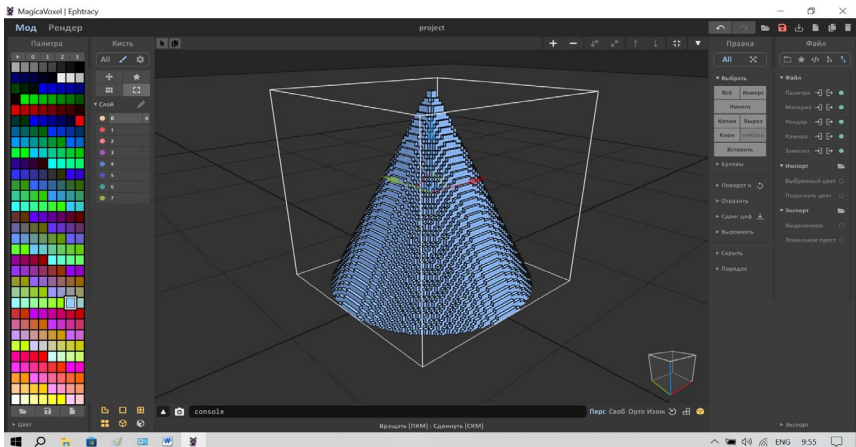


Рисунок 2.11 – Імпорт моделі в MagicaVoxel

Виріз 1/4 частини конуса виконується за допомогою інструменту «Erase» (E) – видалення вокселів (рис. 2.12).



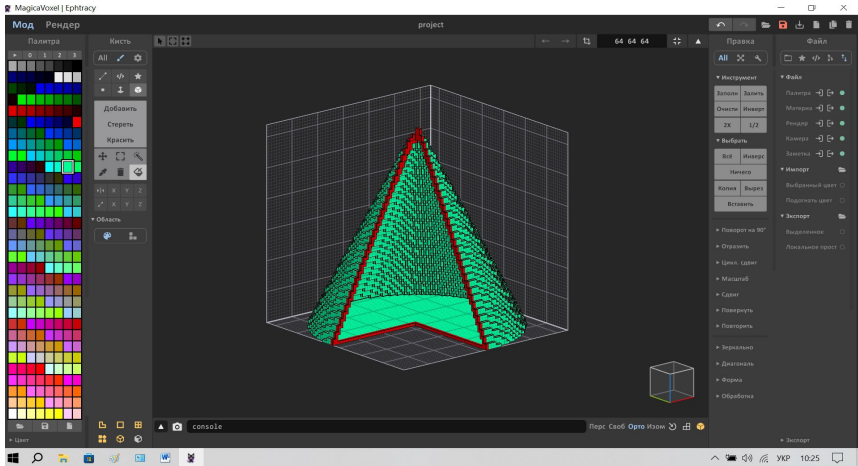


Рисунок 2.12 – Виріз четвертої частини конуса інструментом «Erase» (E)

Виконати рендеринг сцени (рис. 2.13).

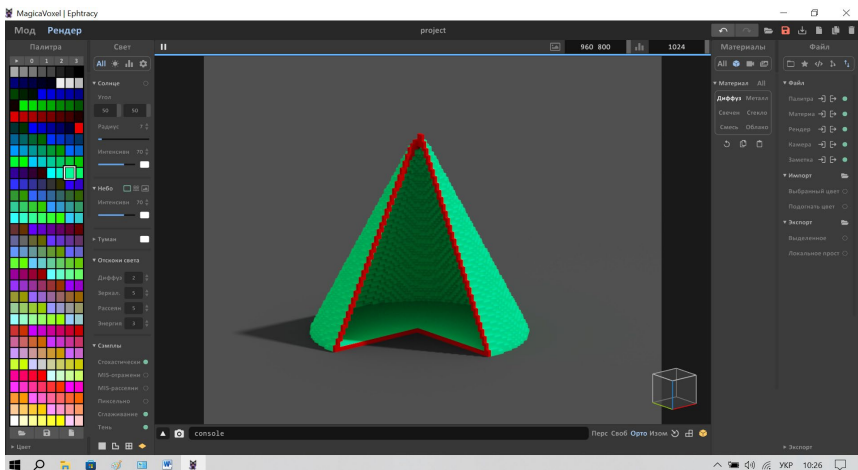


Рисунок 2.13 – Приклад рендера сцени

Приклад виконання завдання і звіту наведено на рис. 2.14 та рис. 2.15.



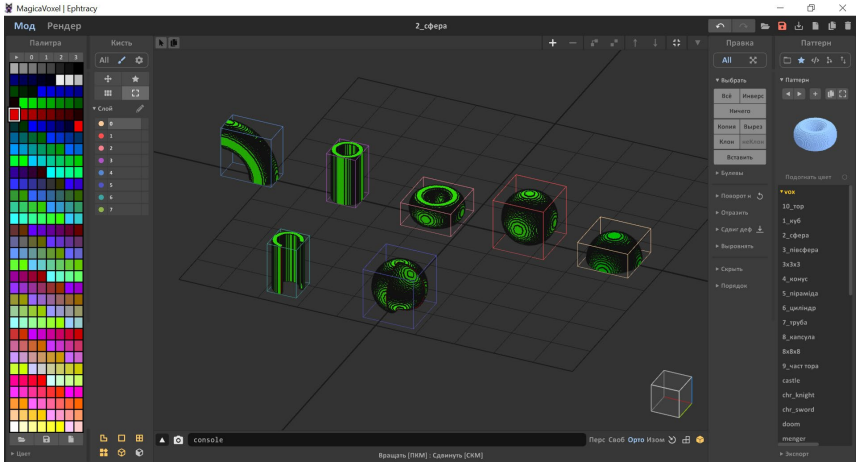


Рисунок 2.14 – Приклад створення сцени за варіантом

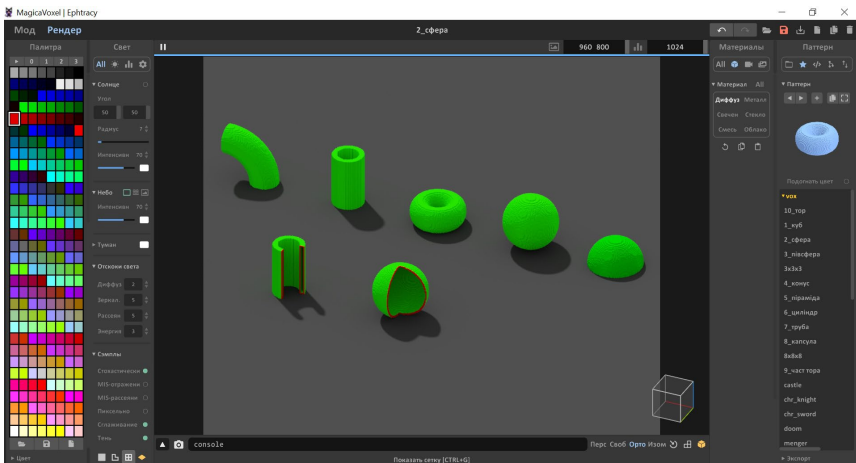


Рисунок 2.15 – Приклад і роздруковка рендера



Повний практикум складається з 12 розроблених завдань.

Для відкритого доступу в репозитарії розміщено два завдання, що демонструють загальну структуру та методику виконання практичних робіт.



РЕСУРСИ ТА ПОСИЛАННЯ

1. Офіційні ресурси
 - Офіційний сайт MagicaVoxel – завантаження останньої версії, приклади, документація. URL: <https://ephtracy.github.io/>
 - GitHub репозиторій (архів версій) – старі версії та додаткові інструменти. URL: <https://github.com/ephtracy/ephtracy.github.io>
2. Спільноти та форуми
 - MagicaVoxel Discord – активна спільнота, допомога у реальному часі. URL: <https://discord.gg/magicavoxel>
 - Reddit – r/MagicaVoxel – приклади робіт, поради, новини. URL: <https://www.reddit.com/r/MagicaVoxel/>
 - VoxelMade – галерея робіт і корисні статті. URL: <https://www.voxelmade.com/>
3. Бібліотеки моделей і матеріалів
 - VoxelModels – безкоштовні готові .vox-моделі. URL: <https://voxelmodels.com/>
 - Sketchfab (тег voxels) – 3D-моделі у стилі вокселів. URL: <https://sketchfab.com/tags/voxel>
 - Kenney.nl Assets – безкоштовні ігрові ресурси, включаючи воксельні . URL: <https://kenney.nl/assets>
4. Відеоуроки та навчальні матеріали
 - YouTube: MagicaVoxel Official Tutorials – базові та просунуті уроки.
 - YouTube: Grant Abbitt – уроки з експорту та інтеграції у Blender.
 - Voxel School – навчальні курси з воксельної графіки.



СПИСОК ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Ephtracy (Z. Yifan). MagicaVoxel User Manual: user guide. – Hong Kong: Ephtracy Studio, 2016-2024. Електрон. ресурс. Офіційний сайт MagicaVoxel – <https://ephtracy.github.io>.
2. Ephtracy (Z. Yifan). MagicaVoxel Official Examples & Shader Library: documentation. – Hong Kong: Ephtracy Studio, 2017. Електрон. ресурс. Офіційний сайт MagicaVoxel – <https://ephtracy.github.io>.
3. Nichibou Shuppan. Voxel Art Collection: Created with MagicaVoxel / Nichibou Shuppan (ed.). – Tokyo : Nichibou Shuppan, 2019. – 143 p. – ISBN 978-4-8170-2093-2. <https://www.nichibun.co.jp>.
4. Nichibou Shuppan Voxel Art Advanced Collections: Created with MagicaVoxel / Nichibou Shuppan (ed.). – Tokyo: Nichibou Shuppan, 2020. – 176 p. – ISBN 978-4-8170-2144-1. <https://www.nichibun.co.jp>.
5. Lewis, Tim. Learning MagicaVoxel: create voxel art for games and design. – [S. l.] : Independently published, 2020. – 124 p. – ISBN 979-8-6346-2788-9. <https://www.amazon.com>.
6. Green, Alex. Voxel Art with MagicaVoxel: practical guide. – [S. l.]: CreateSpace Independent Publishing Platform, 2019. – 118 p. – ISBN 978-1-0987-5642-6. <https://www.amazon.com..>
7. Hamma-neko Studio. Voxel Art Works: Created with MagicaVoxel / Hamma-neko Studio (ed.). – Tokyo, 2018. – 132 p. <https://www.pixiv.net>.
8. MagicaVoxel Artist Collection. Vol. 1 / Japanese Creative Bookstore (ed.). – Tokyo, 2019. – 144 p. <https://japanese-creative-books.com>.
9. MagicaVoxel Artist Collection. Vol. 2 / Japanese Creative Bookstore (ed.). – Tokyo, 2021. – 152 p. <https://japanese-creative-books.com>.
10. Voxel Creators File: MagicaVoxel Edition / Pixiv Inc.



(ed.). – Tokyo: Pixiv Publishing, 2020. – 160 p.
<https://www.pixiv.co.jp>.

11. MagicaVoxel Scene Design Handbook / BOOTH Publishing (ed.). – Tokyo, 2022. – 148 p. <https://booth.pm>

12. Пустюльга С. І., Самчук В. П., Бондарчук Ю. С., Заразка М. Параметризм як стиль в архітектурі та дизайні: концепція, методологія, комп'ютерний інструментарій та перспективи розвитку. Сучасні технології та методи розрахунків у будівництві. Луцьк, ЛНТУ. 2025, Випуск 23, с. 222-239. DOI: [https://doi.org/10.36910/6775-2410-6208-2025-13\(23\)-20](https://doi.org/10.36910/6775-2410-6208-2025-13(23)-20).

13. Пустюльга С.І., Самчук В.П., Пасічник О.С., Чугай Р.В., Градиська Н.Б. Дискретно-воксельний параметричний дизайн. Теорія та практика дизайну: зб. наук. праць. Дизайн. К.: НАУ, 2023. Вип. 27. С. 165-174.
<https://doi.org/10.32782/2415-8151.2023.27.21>.



Навчальне видання

Пустюльга Сергій Іванович
Самчук Володимир Петрович

ДИСКРЕТНО-ВОКСЕЛЬНИЙ ПАРАМЕТРИЧНИЙ ДИЗАЙН У MAGICAVOXEL

**НАВЧАЛЬНИЙ ДОВІДНИК для
студентів технічних ЗВО**

Комп'ютерний набір та верстка: С. Пустюльга, В. Самчук

Дизайн обкладинки: Ю. Самчук

Редактор: С. Пустюльга

Формат 60x84 1/16. Обсяг 15.56 ум. друк. арк., 16.27 обл.-вид. арк.
Наклад 300 пр. Зам. 51. Видавець і виготовлювач – Вежа-Друк
(м. Луцьк, вул. Шопена, 12, тел. + 38 066 936 25 49).
Свідоцтво Держ. комітету телебачення та радіомовлення України
ДК № 4607 від 30.08.2013 р.