



**М. Ю. Морозов, Є. В. Левус, Р. О. Моравський, П. Я. Пустельник**

*Національний університет "Львівська політехніка", м. Львів, Україна*

## ГЕНЕРУВАННЯ ЛАНДШАФТІВ ДЛЯ СФЕРИЧНИХ ПОВЕРХОНЬ: АНАЛІЗ ЗАВДАННЯ ТА ВАРІАНТИ ВИРІШЕННЯ

Проаналізовано проблему генерування ландшафтів за наявними методами для порівняння їх можливостей, виділено основні переваги і недоліки. Розглянуто програмні інструменти, які дають змогу генерувати ландшафти для різних поверхонь. Запропоновано власний метод на підставі поєднання методів, описаних у науковій літературі, який дає змогу гнучко керувати кількісними та якісними показниками моделювання ландшафтів для сферичних поверхонь завдяки введенню параметрів впливу. Зміст методу полягає у застосуванні програмних агентів для відповідного створення складових моделі, а саме – генерування планетоїда, клімату та моделі ландшафту загалом. Для попереднього оброблення моделі планетоїда здійснюють генерування опуклої оболонки та виконують вибір програмних агентів з алгоритмами для оброблення ландшафтів, а також створення бази даних для зберігання всіх результатів. Програмні агенти під час оброблення використовують маски, які потрібні для контролю впливу кожного програмного агента на модель ландшафту загалом. На відміну від відомих рішень, де шари є неподільні і їхня зміна щоразу вимагає повного перерахунку всього ландшафту, запропонований метод дає змогу вільно модифікувати вплив одних агентів на інші на підставі задавання різних масок, а також ділити створені шари на підрівні. Завдяки введенню програмних агентів і масок метод автоматизовано здійснює параметризацію процесу генерування ландшафтів деталізованих планетоїдів з подальшою їх серіалізацією та обробленням. Застосування програмних агентів дає змогу забезпечити гнучкість методу (урахування різних параметрів моделі планетоїда за різного порядку застосування програмних агентів), економічність виконання обчислень (для різної деталізації сегментів сферичної поверхні не потрібні обчислення з "нуля"). Перевагами запропонованого рішення є врахування різних деталей для забезпечення високої реалістичності результату та уникнення зайвих обчислень для різних рівнів зближення огляду поверхонь.

**Ключові слова:** висока реалістичність; планетоїд; програмний агент; маска впливу; алгоритми шуму; фрактальне оброблення.

**Вступ.** Програмні засоби моделювання віртуальних світів є необхідним інструментом сьогодення в галузях наукових досліджень, освіти, індустрії розваг, кінематографії тощо. За їх допомогою створюють, наприклад, образ цілої планети з реалістичними ландшафтами, який можна використати у курсі ландшафтознавства чи в комп'ютерній грі.

Завдання генерування ландшафтів планетоїдів містить процедуру моделювання рельєфу з урахуванням ерозії ґрунтів, даних про клімат, біоми, відтворення водних поверхонь. Створення ландшафтів високої реалістичності відбуваються за допомогою алгоритмів перетворення вхідних даних з побудовою на їх підставі карти висот і накладання відповідних ефектів. Від якості використаних зображень залежить точність їх сприймання користувачами наукових програм чи навчальних систем, прибутковість гри чи кінофільму, а також подальше удосконалення відповідних програмних систем.

Основними проблемами наявних рішень є їхня не реалістичність для великих масштабів, ціна повних вер-

сій відповідного програмного забезпечення та відсутність засобів для перетворення результатів у планетоїди із правильним збільшенням (різні рівні деталізації секторів сфери). Також ці програмні застосунки мають закрити і складно розширювану бібліотеку алгоритмів для генерування ландшафтів, що зменшує гнучкість власної розробки. У межах проєктів, для яких потрібні відносно невеликі ландшафти (візуалізація геологічних чи екологічних даних, прольоти камерою в кінематографії та мультиплікації, комп'ютерні ігри з відкритим світом), наявні рішення можна ефективно застосовувати попри наявні недоліки.

Проте, останнім часом дедалі більше виникає потреба візуалізувати дані у світових масштабах (проєкт *Digital Elevation Data, Ecological Data*), а також щораз більше фільмів і комп'ютерних ігор демонструють складні польоти камерою та взаємодію з планетами при різних наближеннях. Наявні алгоритми використовують повністю окремі моделі планетоїдів з подальшою їх заміною і згладжуванням переходів, що не завжди є

### Інформація про авторів:

**Морозов Микола Юрійович**, студент, кафедра програмного забезпечення. Email: yelevus@gmail.com

**Левус Євгенія Василівна**, канд. техн. наук, доцент, кафедра програмного забезпечення.

Email: yelevus@gmail.com; <https://orcid.org/0000-0001-5109-7533>

**Моравський Роман Орестович**, студент, кафедра програмного забезпечення. Email: yelevus@gmail.com

**Пустельник Павло Ярославович**, студент, кафедра програмного забезпечення. Email: yelevus@gmail.com

**Цитування за ДСТУ:** Морозов М. Ю., Левус Є. В., Моравський Р. О., Пустельник П. Я. Генерування ландшафтів для сферичних поверхонь: аналіз завдання та варіанти вирішення. Науковий вісник НЛТУ України. 2020, т. 30, № 1. С. 136–141.

**Citation APA:** Morozov, M. Yu., Levus, Ye. V., Moravskiy, R.O., & Pustelnyk, P. Ya. (2020). Landscape generation for spherical surfaces: problem analysis and solution. *Scientific Bulletin of UNFU*, 30(1), 136–141. <https://doi.org/10.36930/40300124>

ефективними для цих задач. Загалом проблема визначається складністю і довготривалістю розроблення реалістичних ландшафтів для високобюджетних програм, відсутністю зручних програмних продуктів для генерування планетоїдів чи великих ландшафтів (сегментів планетоїдів), нестачею вільнодоступних алгоритмів для оптимізованого відображення сферичних топологій ландшафтів.

Ця галузь знань є перспективною для проведення різних досліджень, оскільки, з одного боку, необхідно покращувати зовнішній вигляд моделей, а з іншого – не можна виходити за рамки можливої кількості обчислень, тобто потрібно розробляти нові моделі ландшафту і більш оптимізовані алгоритми. Тому розроблення нового, оптимізованого методу генерування ландшафтів для сферичних поверхонь, є актуальним завданням як з наукової, так і з практичної точок зору.

*Об'єктом дослідження* є генерування ландшафту сферичних поверхонь.

*Предметом дослідження* є метод генерування ландшафту з використанням програмних агентів і масок впливу.

*Мета роботи* – підвищення якості відображення та ефективності процесу генерування ландшафту сферичних поверхонь при будь-якому рівні збільшення.

*Завдання дослідження* охоплює: аналіз методів і програмних засобів генерування ландшафтів для сферичних поверхонь; побудову власного методу з можливістю керувати якісними та кількісними показниками процесу моделювання ландшафтів сферичних поверхонь для забезпечення високої реалістичності зображення та економії витрат процесорного часу.

*Наукова новизна дослідження* полягає в побудові методу генерування ландшафтів для сферичних поверхонь, що дає змогу керувати кількісними та якісними показниками моделювання, внаслідок чого підвищується реалістичність результату відображення та продуктивність процесу відображення при різних рівнях деталізації сферичних поверхонь.

*Аналіз літературних джерел.* На підставі літературних джерел проведено огляд методів генерування ландшафтів. Початки досліджуваної проблеми датовані 1980-ми роками. На той час розглядали алгоритми обчислення та подання фіксованих примітивних ландшафтів. Уже тоді в роботі [15] було наведено алгоритм поділу поверхні на трикутники з подальшою її деформуванням по вертикалі за допомогою фрактального шуму. Після цього було розроблено методу збирання, зберігання та оброблення ландшафтів [10].

Віднедавня почали з'являтися дослідження, орієнтовані на побудову ландшафтів за допомогою нейронних мереж. Ідея полягає в селективному (ручному чи автоматичному) відборі "красивих" ландшафтів, на підставі яких мережа може навчатися [11, 12, 13]. Після зростання популярності генеративних змагальних мереж на їх підставі спробували будувати ландшафти, значно якісніші, ніж з використанням попередніх методів [4, 14]. Однак, усі згадані вище алгоритми покривають тільки частину побудови ландшафту. Наприклад, нейронні мережі зі селективним відбором вирішували проблему карт висот, а метод TileGAN (синтез великомасштабних неоднорідних текстур) створює зображення, схожі на супутникові, проте не надає інформації про розташування об'єктів на цьому зображенні, що не дає змоги побудувати якісне 3D-подання.

Перевагою процесу генерування ландшафтів за допомогою нейронних мереж є можливість їх урізноманітнення за допомогою невеликих мутацій нейронної мережі. Окрім цього, можна застосовувати деградацію нейронної мережі, прибираючи окремі слабкі нейрони. Це призведе до деградації результату, що може бути корисним художнім ефектом для отриманого зображення.

Окрім простої побудови плоских ландшафтів, проводили дослідження й для побудови цілих планет. У 2007 р. було подано презентацію від творців гри Spore [6], де алгоритмічно генерувалася ціла галактика різних планет, які можна було відвідувати під час гри. У роботі [20] наведено якісний метод оптимізованого генерування моделей планет на підставі заготовлених даних про океани і сушу. У роботі [5] подано новий алгоритм оброблення та відображення планетоїдів, який забезпечує низьке навантаження на відеокарту та малі об'єми використаної пам'яті.

На сьогодні відомі декілька підходів до генерування планетоїдів з ландшафтами. Перший підхід – генерування плоских ландшафтів на підставі наявних алгоритмів з подальшим згинанням цих ландшафтів при віддаленні від планетоїда (коли стає видно кривизну горизонту) [20]. Цей підхід можна використовувати, якщо видимою є невелика частина ландшафту. Другий підхід – поділ горизонту на достатньо малі ділянки, які можна генерувати з урахуванням її опуклості [5]. Також створюються спрощені подання груп таких ділянок для оптимізації при віддаленні (Mip-mapping, MIP-текстурування – метод текстурування, що використовує декілька копій однієї текстури з різною деталізацією. Назва походить від лат. *multum in parvo* – "багато в одному"). Цей підхід використовують, коли необхідно розглядати планетоїд або зблизька, коли велика кількість дрібних ділянок забезпечує згладженість зображення, або здалека, коли планетоїд достатньо малий у кадрі, що відсутність заокруглення ландшафту непомітна. Третій підхід – генерування опуклого планетоїда з самого початку, з подальшим нанесенням деталей [6, 8]. Це забезпечує максимальну гнучкість під час генерування, проте значно ускладнює роботу алгоритму генерування та оброблення.

Інші дослідження, що стосуються генерування ландшафтів, зосереджені на використанні кривих. Зазвичай, криві задає користувач [17, 25] на підставі річок [16, 27] чи тектонічних плит [7, 31]. Кожна крива, на якій базується ландшафт, у об'єднанні з іншими кривими екстраполюється для отримання даних про всю територію ландшафту. Після цього будуватиметься модель поверхні ландшафту на підставі полігональної сітки, яка водночас, створюється на підставі кривих. Така побудова дає користувачу змогу легко впливати на вигляд ландшафту, впливаючи на контрольні та опорні точки кривих. Найчастіше в такому випадку використовують квадратичні та кубічні криві Без'є. Усі подальші надбудови (ріки, тектонічні плити) можна звести до перетворення вхідних даних на набір кривих.

Побудова ландшафтів через шум є найпростішим підходом серед розглянутих методик. Вона полягає в накладанні шуму з різною амплітудою та розміром на базову поверхню (зазвичай, площину), згенерованих за допомогою багатьох алгоритмів. Вихідним значенням цього накладання буде карта висот, яку можна додатково обробляти іншими алгоритмами. Найчастіше вико-

ристовують такі алгоритми шуму: броунівський рух [8], значеннєвий [24], фрактальний [28], тесесяція Вороного [22], шум Перліна [2] та простий шум [29].

Цікавий новий підхід до створення ландшафтів на підставі програмних агентів, кожен з яких містить певний алгоритм оброблення ландшафту [9]. Відповідно, наприклад, можливо за допомогою першого агента додати рельєф за допомогою алгоритму шуму. Далі другим агентом уточнити його через криві, потім обробити нейронною мережею для ще більшої реалістичності.

Отже, можна виділити основні групи алгоритмів, що застосовують для вирішення завдання генерування ландшафтів планетоїдів, робота яких базується на:

- 1) застосуванні нейронних мереж;
- 2) модифікації плоских ландшафтів;
- 3) екстраполяції кривих;
- 4) використанні шумів.

**Аналіз програмних засобів генерування ландшафтів.** Для виявлення вимог до власного програмного рішення проаналізовано можливості найвідоміших програмних інструментів, призначених для моделювання високо реалістичних ландшафтів. Серед розглянутих систем тільки дві надають можливість виконувати роботу з ландшафтами сферичних поверхонь (англ. *Planet Painter, Planetary Terrain*).

World Creator 2 – це генератор ландшафтів у реальному часі, який виконує всі процеси генерування та дизайну повністю на графічному процесорі, використовуючи тисячі ядер. Для створення реалістичних ландшафтів (рис. 1) користувачу доступні такі інструменти:

- дизайнер форми – у будь-якому місці місцевості можна змінити висоту;
- дизайнер кольору та відтінку;
- фільтри – каньйони, гори, пагорби, плато, дюни, вітри, кратери та інші;
- фільтри ерозії та седиментації;
- розміщення у генеровану місцевість таких об'єктів, як дерева, трава, каміння, міста тощо;
- створення доріг та річок;
- створення власних фільтрів ландшафту;
- експорт та імпорт (OBJ, RAW 8, RAW 16, RAW 32, PNG, JPG, FBX, BMP, GeoTIFF, TIF) мап висот, масок (місцевості, текстур, об'єктів, доріг, річок, озер), теплових карт, XML файлів для об'єктів;
- експорт у Blender, Cinema 4D, Houdini, 3DS Max, Photoshop, Terragen і синхронізація з Unity, Unreal.

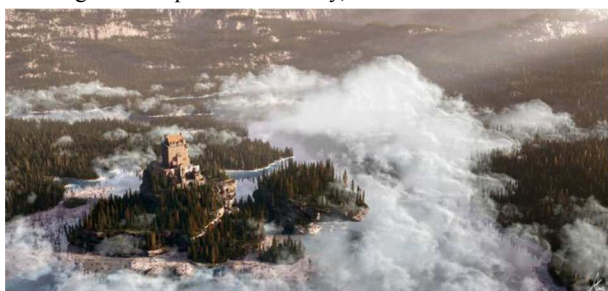


Рис. 1. Приклад згенерованого ландшафту за допомогою World Creator 2

World Machine – програма для генерування 3D-ландшафтів, яку широко використовують для створення складних сцен у AAA ігрових проектах і фільмах. Робота графічного інтерфейсу створення ландшафту базована на графах, підтримує системи blueprint, grouping, masks. Програма надає потужні інструменти для накладання ерозій, що створює ефект натуральності. Підтримується потокова ерозія, яка утворює яри та руйнує

гори, термальна ерозія, яка завдає шкоди схилам та накопичує під ними зруйновані залишки, а також прибережна ерозія, що з легкістю дає змогу формувати реалістичні пляжі. Програма World Machine підтримує створення річок, морів, сніжних покривів, формування цілих біомів. Додатковою перевагою є можливість експортувати створений ландшафт в такі програми, як Unity, Unreal Engine та інші.

Творець плагіну GeoGlyph (для World Machine) разом із командою розробили власний генератор ландшафтів – Gaia. Ця програма надає користувачу можливість налаштувати ландшафт як і через базований на графах інтерфейс, так і через Photoshop-style layers або пряме конструювання. Графічний інтерфейс графів і вузлів, порівняно з іншими схожими програмами, є більш гнучким та зручним. Можна з легкістю додавати цілі біоми, фільтри, ерозію та інші речі. Детально розроблений інструмент ерозії, а саме: термальна ерозія, природні процеси, напрямлена ерозія.

Візуалізація ландшафту доступна також у тривимірному просторі в реальному часі за допомогою програми Unity Engine. 3D-ландшафт можна експортувати в OBJ, FBX чи DAE форматі, і у 2D-висотну карту в 16-бітному або 32-бітному форматі.

Terragen – застосунок, який дає змогу створювати та рендерити симуляцію реального навколишнього середовища. Основними особливостями Terragen'у є:

- гнучка система вузлів для точного налаштування об'єктів;
- реалістичне моделювання сферичної атмосфери, в т.ч. багаторівневі хмари, вітер, озоновий шар, рух сонця та інше. Все це контролюється за допомогою кількох шейдерних мереж;
- трасування променів для реалістичного відображення оточення, відбивання світла та його взаємодії із атмосферою;
- генерування натуральних ландшафтів і можливість імпорту наявних зі сторонніх програм;
- можливість додавання та рендеру складних об'єктів із текстурами та необмеженою кількістю полігонів;
- анімація об'єктів зі зміною його параметрів, а не тільки камера;
- підтримка різних видів ерозій, таких як флювіальна ерозія, термальна ерозія, річкова ерозія.
- експорт створених сцен у інші застосунки.

Planet Painter – безкоштовна програма, яка дає змогу створювати та редагувати невеликі сферичні планети з водним або земляним покривом (рис. 2). Основними можливостями застосунку є нанесення кольору та редагування висот сферичного об'єкта. Редагування відбувається у реальному часі.

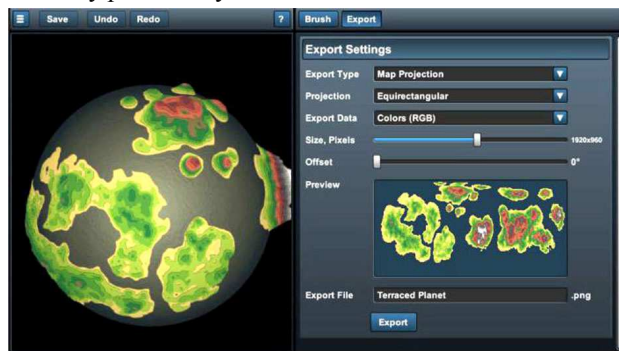


Рис. 2. Приклад роботи у програмі Planet Painter

Planetary Terrain – додаток для Unity, робота якого заснована на дереві квадрантів. Цей застосунок дає змогу генерувати сферичні планети великих розмірів, а також накладати на згенеровані планети шуми (ерозії) за



допомогою редактора вузлів. Генерування шумів можна на базі центрального або графічного процесора, що дає змогу ефективно розподіляти навантаження.

Додатковими особливостями застосунку є система листя для процедурного накладання трави та інших об'єктів на поверхню, можливість створювати довільну кількість планет в одному просторі з можливим обертанням. Також Planetary Terrain охоплює різні режими генерування планет, такі як шум, карту висот та їх гібрид (вбудований генератор карт висот на підставі шуму), а також підтримує текстурівання планети на підставі висоти та нахилу.

**Таблиця. Порівняння програмних застосунків для генерування ландшафтів**

|                   | Сферичний ландшафт | Симуляція в реальному часі | Можливість накладання фільтрів (ерозії, біоми, скелі, пагорби) | Ціна, \$ |
|-------------------|--------------------|----------------------------|--|----------|
| World Creator 2   | -                  | +                          | +  | 149      |
| World Machine     | -                  | -                          | +  | 119      |
| Gaea              | -                  | +                          | +  | 99       |
| Terragen          | -                  | -                          | +  | 349      |
| Planet Painter    | +                  | +                          | -  | Free     |
| Planetary Terrain | +                  | -                          | +  | 40       |

Хоча всі описані вище програмні засоби (окрім Planet Painter) надають досить схожий функціонал для генерування ландшафтів, сфера їх застосування частково відрізняється (таблиця). Такі програми, як Terragen та Gaea здебільшого використовують у кіноіндустрії. Наприклад, у програмі Terragen було згенеровано сцену заставки для Paramount. В ігровій індустрії застосовують переважно World Creator 2, World Machine та плагін Planetary Terrain (який інтегрований в Unity). Особливо популярною є програма World Creator 2, яка надає розробнику можливість генерувати ландшафти в реальному часі та багато інших функцій. Для розроблення Інді-ігор використовують програму Planet Painter (алгоритм якого частково описаний на сайті додатку).

Варто зазначити, що в усіх згаданих вище програм є доступна пробна версія, а вартість повної версії часто сягає декількох тисяч доларів.

**Запропонований метод.** Робота методу базується на використанні програмних агентів і процедур генерування вмісту (алгоритми генерування карти висот, робота яких базована на шумах чи фракталах). Глобальними кроками методу є попереднє оброблення, генерування планетоїда, генерування клімату та генерування моделі ландшафту загалом.

Для попереднього оброблення моделі планетоїда відбувається генерування опуклої оболонки та здійснюється вибір програмних агентів з алгоритмами для оброблення ландшафтів, а також створення бази даних для зберігання всіх результатів. Для генерування планетоїда повністю використовують програмні агенти: заповнення оболонки планетоїда речовиною, створення тектонічних плит, згладжування, додавання шуму, фрактальне оброблення ландшафту та створення рельєфу.

Наступним кроком є генерування клімату, який водночас, складається з визначення рівня моря та генерування світового океану, генерування кліматичних даних і погодних біомів, додавання інших водойм, застосування ерозії на підставі симуляції клімату, заповнення ландшафту даними про природне життя.

Останнім кроком є генерування моделі, під час якої відбувається заповнення ландшафту декоративними об'єктами, створення вибірки даних на підставі структурованого запиту користувача, генерування тривимірної моделі на підставі запиту користувача, відображення даних з їхньою оптимізацією у вигляді оклюзивного обрізання.

Пропонований метод передбачає виконання таких глобальних кроків I-IV [26].

**Крок I:** Попереднє оброблення

1. Генерування опуклої оболонки, поділеної на дерево октантів зі сферичним методом доступу [6].
2. Створення файлового подання використовуваних структур даних для подальшого збереження, завантаження та кешування [8].
3. Вибір набору алгоритмів генерування ландшафту на підставі вхідних даних у формі програмних агентів.

**Крок II:** Генерування планетоїда

1. Заповнення опуклої оболонки геологічними даними про планету.
2. Генерування спрощених моделей тектонічних плит для створення карти висот [7, 31].
3. Згладжування поверхонь.
4. Додавання до поверхні значення шуму [24].
5. Застосування до поверхні швидких алгоритмів шуму (Perlin, Simplex, Voronoi) [1, 2, 8, 18].
6. Фрактальне оброблення ландшафту [28].
7. Збереження результатів симуляції в опуклу оболонку.
8. Створення рельєфу на підставі нейронних мереж [4, 14].

**Крок III:** Генерування клімату

1. Визначення рівня моря, генерування світового океану.
2. Попереднє генерування клімату і погодних біомів [3].
3. Оброблення інших водойм (озера, ставки і т.ін.) та річок на підставі даних про клімат.
4. Застосування ерозії та створення додаткових даних про водойми на підставі симуляції клімату [32].
5. Оброблення даних для відкидання неможливих і непотрібних деталей на підставі наявних даних [19].
6. Заповнення ландшафту даними про природне життя [21, 23, 30].

**Крок IV:** Генерування моделі

1. Випадкове заповнення ландшафту декоративними об'єктами.
2. Створення вибірки даних на підставі структурованого запиту користувача.
3. Генерування тривимірної моделі на підставі запиту користувача [9].
4. Відображення даних з оптимізацією у вигляді оклюзивного обрізання на підставі структур даних.

Особливістю запропонованого методу є те, що оброблення початкової моделі планетоїда відбувається програмними агентами [9], вплив кожного для генерування ландшафту задається згідно з потребами. Програмні агенти відповідають за виконання кроків, що стосуються процедури генерування планетоїда, клімату та моделі. Програмні агенти під час оброблення використовують дво- чи тривимірні маски, які необхідні для контролю впливу кожного програмного агента на модель ландшафту загалом.

Нехай  $h_i$  – дані, згенеровані  $i$ -м агентом,  $mask_i$  – маска впливу  $i$ -го агента на ландшафт,  $h_{or}$  – початкові дані,  $mask_{or}$  – початкова маска. Тоді  $result$  – вихідна висота певної точки ландшафту після накладання маски обчислюється за формулою

$$result = \sum_{i=0}^n h_i \cdot mask_i + h_{or} \cdot mask_{or} .$$

Наприклад, у другому глобальному кроці методу отримують результати оброблення даних програмними агентами:

- $h_1$  – результат накладання Перлінового (чи іншого) шуму на висоту ландшафту (*software\_agent\_1*),
- $h_2$  – результат фрактального оброблення висоти поверхні (*software\_agent\_2*),
- $h_3$  – висота, що змінена на підставі даних карти висот з даних реального світу (*software\_agent\_3*),
- $h_{or}$  – початкова висота точки ландшафту,
- *mask* – значення маски в оброблюваній точці ландшафту, маскою є дво- чи тривимірна таблиця коефіцієнтів для точки ландшафту для кожного програмного агента. Тоді матимемо такий математичний вираз

$$result = mask(software\_agent\_1) \cdot h_1 + mask(software\_agent\_2) \cdot h_2 + mask(software\_agent\_3) \cdot h_3 + mask(height\_original) \cdot h_{op}.$$

Використання масок дає змогу позбутися частини обчислень, оскільки зміни параметрів шарів можна виконати через маски. У відомому рішенні [26] шари є неподільні, тому їхня зміна щоразу вимагає повного перерахунку всього ландшафту. Ми визначили, що програмне рішення для генерування та відображення тривимірних моделей планетоїдів і їх ділянок повинно відповідати таким функціональним характеристикам:

- створення набору програмних агентів для гнучкого оброблення планетоїдів, що містить визначення їх послідовності та сили впливу;
- застосування програмних агентів для оброблення планетоїдів з урахуванням різних деталізованих конфігурацій;
- завантаження з БД моделей ландшафтів і властивостей планетоїда, над яким відбувся процес оброблення;
- збереження обробленого планетоїда після завершення виконання певного програмного агента;
- відображення моделі на підставі обробленого планетоїда;
- створення декорованої моделі на підставі 3D-відображення планетоїда та бібліотеки декорацій.

### Висновки:

1. Проаналізовано методи та програмні засоби генерування ландшафтів для виявлення їх можливостей та переваг застосування для сферичних поверхонь.
2. Спроектовано метод генерування ландшафтів для сферичних поверхонь, який містить попередню підготовку, результат якої обробляється програмними агентами, що відповідають за генерування планетоїдів, клімату та моделі.
3. Для автоматизації процесу отримання розв'язку задачі та його параметризації запропоновано використовувати маски під час оброблення даних програмними агентами, які необхідні для контролю впливу кожного програмного агента на модель ландшафту загалом.
4. Перевагою такого підходу є те, що отримання різноманітних варіантів моделей ландшафту при будь-якому рівні зближення без значних витрат процесорного часу, що є результатом автоматизації та параметризації процесу генерування ландшафтів.
5. Продовження дослідження – це програмна імплементація розробленого методу у вигляді системи, що відповідає зазначеним вище функціональним характеристикам.

### References

1. Academic Mindtrek '17 Proceedings of the 21st International Academic Mindtrek Conference. (pp. 17–25). <https://doi.org/10.1145/3131085.3131099>
2. Bambich, J. (2015). Procedural terrain generation with Perlin noise. University of Ljubljana.

3. Bambich, J. (2018). Procedural terrain generation for use in computer games. University of Ljubljana.
4. Beckham, C., & Pal, C. (2017). A step towards procedural terrain generation with GANs.
5. Clasen, M., & Hege, H.-Ch. (2006). Terrain Rendering using Spherical Clipmaps. *EuroVis06: Joint Eurographics – IEEE VGTC Symposium on Visualization*, Lisbon, Portugal. <https://doi.org/10.2312/VisSym/EuroVis06/091-098>
6. Compton, K., Grieve, J., Goldman, E., et al. (2007). Creating spherical worlds in Proceedings of the International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '07), ACM, New York. (p. 82). <https://doi.org/10.1145/1278780.1278879>
7. Cordonnier, G., Braun, J., Cani, M.-P., et al. (2016). Large Scale Terrain Generation from Tectonic Uplift and Fluvial Erosion. *Computer Graphics Forum*, Wiley. <https://doi.org/10.1111/cgf.12820>
8. d'Oliveira, R., Iago, P. do E. Santo, & Apolinario, Jr. A. (2018). Procedural Planet Generation based on derivate fBm noise. *Proceedings of SBGames*, 667 p.
9. Doran, J., & Parberry, I. (2010). Controlled Procedural Terrain Generation Using Software Agents. *IEEE Transactions on Computational Intelligence and AI in Games*, 2(2). <https://doi.org/10.1109/TCIAIG.2010.2049020>
10. Ebner, H., & Reinhardt, W. (1988). *Generation, management and utilization of high fidelity digital terrain models*, July 1988ISPRS XVI Congress At: Kyoto, Japan, Vol. 27, part B11.
11. Frade, M., Fernandez de Vega, F., & Cotta, C. (2008). Breeding Terrains with Genetic Terrain Programming: The Evolution of Terrain Generators. *International Journal of Computer Games Technology*, 2009, 13. Malaga, Spain. Article ID 125714. <https://doi.org/10.1155/2009/125714>
12. Frade, M., Fernandez de Vega, F., & Cotta, C. (2008). Genetic Terrain Programming An Aesthetic Approach to Terrain Generation. Malaga, Spain.
13. Frade, M., Fernandez de Vega, F., & Cotta, C. (2008). Modelling Video Games' Landscapes by Means of Genetic Terrain Programming – A New Approach for Improving Users' Experience. Malaga, Spain, Geography, *Computer Science Published in Evo-Workshops*. [https://doi.org/10.1007/978-3-540-78761-7\\_52](https://doi.org/10.1007/978-3-540-78761-7_52)
14. Frühstück, A., Alhashim, I., & Wonka, P. (2019). TileGAN: Synthesis of Large-Scale Non-Homogeneous Textures. *Journal ACM Transactions on Graphics*, 38(4). <https://doi.org/10.1145/3306346.3322993>
15. Gavin, S. P. Muller. (1986). The definition and rendering of terrain maps. *SIGGRAPH*, Dallas, August 18–22. (Vol. 8, № 4). <https://doi.org/10.1145/15886.15890>
16. Genevaux, J.-D., Galin, E., Guérin, E., et al. (2013). *Terrain Generation Using Procedural Models Based on Hydrology*. Université de Lyon, LIRIS, CNRS, UMR5205, France. <https://doi.org/10.1145/2461912.2461996>
17. Hnaidi, H., Guérin, E., Akkouche, S., et al. (2010). *Feature based terrain generation using diffusion equation*. LIRIS – CNRS – Université Lyon 1, Franc. <https://doi.org/10.1111/j.1467-8659.2010.01806.x>
18. Tuomo Hyttinen, Erkki Mäkinen, & Timo Poranen. (2017). Terrain synthesis using noise by example. *AcademicMindtrek '17: Proceedings of the 21st International Academic Mindtrek Conference*, pp. 17–25. <https://doi.org/10.1145/3131085.3131099>
19. Korneliusz, K. Warszawski, & Sławomir, S. Nikiel. (2014). A proposition of erosion algorithm for terrain models with hardness layer. *Terrain modelling, Erosion simulation*.
20. Lambrechts, J., Comblen, R., & Legat, V. (2008). Multiscale mesh generation on the sphere. *Ocean Dynamic*. <https://doi.org/10.1007/s10236-008-0148-3>
21. Makowski, M., Haedrich, T., Scheffczyk, J., et al. (2019). Synthetic Silviculture: Multi-Scale Modeling of Plant Ecosystems. *Computing methodologies, Interactive simulation*.
22. Olsen, J. (2004). Realtime Procedural Terrain Generation. Realtime Synthesis of Eroded Fractal Terrain for Use in Computer Games, pp. 1–20. Retrieved from: <https://web.mit.edu/cesium/Public/terrain.pdf>

23. Onrust, B., Bidarra, R., Rooseboom, R., J., & van de Koppel. (2017). Ecologically Sound Procedural Generation of Natural Environments. *International Journal of Computer Games Technology*. <https://doi.org/10.1155/2017/7057141>
24. Parberry, I. (2014). Designer Worlds: Procedural Generation of Infinite Terrain from Real-World Elevation Data. *Journal of Computer Graphics Techniques*, 3(1).
25. Puig-Centelles, A., Varley, P., A., at al. (2012). Automatic Terrain Generation with a Sketching Tool. *Springer Science and Business Media*. <https://doi.org/10.1007/s11042-012-1214-x>
26. Smelik, R. M., Tutene, T., at al. (2008). A Proposal for a Procedural Terrain Modelling Framework. *EGVE Symposium*.
27. Soon Tee Teoh. (2008). River and Coastal Action in Automatic Terrain Generation. San Jose, USA.
28. Stachniak, S., & Stuerzlinger, W. (2005). An Algorithm for Automated Fractal Terrain Deformation. *Proceedings of Computer Graphics and Artificial Intelligence*, 1, 64–76.
29. Thorimbert, Y., & Chopard, B. (2018). Polynomial methods for fast Procedural Terrain Generation.
30. Valencia-Rosado, Oswaldo, L., & Starostenko, O. (2019). Methods for Procedural Terrain Generation: A Review. *Mexican Conference on Pattern Recognition*. Springer, Cham.
31. Viitanen, L. (2012). Physically Based Terrain Generation: Procedural Heightmap Generation Using Plate Tectonics. Helsinki Metropolia University of Applied Sciences.
32. Vuontisjärvi, H. (2014). *Procedural planet generation in game development*. Oulu University of Applied Sciences.

**M. Yu. Morozov, Ye. V. Levus, R.O. Moravskiy, P. Ya. Pustelnyk**

*Lviv Polytechnic National University, Lviv, Ukraine*

## LANDSCAPE GENERATION FOR SPHERICAL SURFACES: PROBLEM ANALYSIS AND SOLUTION

Landscape image quality for spherical surfaces, including planetoids, affects the accuracy of their perception by the users of scientific or educational systems, profitability of games or movies, evolution of relevant software systems. The main issues of available solutions are lack of realism on a big scale, price of full versions of relevant software and absence of tools for conversion from software processing results to planetoids with effective scaling. There are several existing approaches to planetoid landscape generation. Based on a combination of known methods, a solution is proposed, that relies on the use of software agents. The global steps of the designed method consist of preprocessing and generation of planetoids, climate and 3 d models. For planetoid model preprocessing, a convex hull is generated, a set of software agents with landscape processing algorithms is chosen and a database is created for generation results. Planetoid generation consists entirely of software agents: filling the hull with substance, generating tectonic plates, smoothing, adding noise, processing landscapes with fractals and creating relief based on neural networks. Climate generation stage includes determining sea level and generating the world ocean, generating climate data, weather biomes and other water reservoirs, applying erosion based on climate simulation and filling landscape with natural life data. The final step is generating the 3d model, during which the landscape is filled with decorations, data samples are created based on structured user queries, 3d models are generated based on user requests, data is visualized with occlusion culling optimization. Unlike known solutions, where data layers are atomic and their altering requires full landscape recalculation every time, our proposed method allows to freely modify the influence of one software agent on the others, while dividing layers into sublayers. Software agents during processing use two- or three-dimensional masks, which are required in order to control the influence of each software agent on the model as a whole. The advantage is the automatization and the parameterization of detailed planetoid landscape generation with further serialization and processing. This area of research is promising, because on one side, it is necessary to improve the look of models, while on the other side, it is desirable not to go beyond the possible number of calculations. In other words, there is a demand for new models and more optimized algorithms.

**Keywords:** high realistic; planetoid; software agent; exposure mask, noise, fractal processing.