

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
ПрАТ «ПРИВАТНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«ЗАПОРІЗЬКИЙ ІНСТИТУТ ЕКОНОМІКИ
ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ»

Предметно-циклова комісія інформаційних технологій

ДО ЗАХИСТУ ДОПУЩЕНА

Голова ПЦК _____

спеціаліст в/к Сабанов С.О.

ВИПУСКНА РОБОТА МОЛОДШОГО СПЕЦІАЛІСТА

**РОЗРОБКА ПРОГРАМНОГО КОМПЛЕКСУ СИМУЛЯЦІЇ
ЕКОСИСТЕМИ "АКВАРІУМ"**

Виконав

ст. гр. ІПЗ-119К9 _____

Мельник І.М.

Керівник

професор _____

Сабанов С.О.

Запоріжжя
2023

СТРУКТУРНИЙ ПІДРОЗДІЛ «ФАХОВИЙ КОЛЕДЖ ЕКОНОМІКИ ТА
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРАТ «ПВНЗ «ЗІЕІТ»

Предметно-циклова комісія інформаційних технологій

ЗАТВЕРДЖУЮ

Голова ПЦК спеціаліст в/к

Сабанов С.О. _____

“21” лютого 2023 року

ЗАВДАННЯ

НА ВИПУСКНУ РОБОТУ МОЛОДШОГО СПЕЦІАЛІСТА

студенту гр. ППЗ-119К9

спеціальності: 121 – Інженерія програмного забезпечення

Мельнику Іллі Максимовичу

(прізвище, ім'я, по батькові)

1. Тема: Розробка програмного комплексу симуляції екосистеми "Акваріум"

затверджена наказом по інституту: № 09.2-14 від 20 лютого 2023 року

2. Термін здачі студентом закінченої роботи: 20 червня 2023 року

3. Перелік питань, що підлягають розробці:

1. Провести огляд літератури та інтернет-джерел, присвячених
тематиці випускної роботи.

2. Виконати огляд найбільш популярних комп'ютерних симуляцій водних
екосистем.

3. Розглянути інструментальні засоби створення симуляторів з метою
вибору оптимальних для створення проекту.

4. Визначитися з основними функціональними блоками проекту та
структурою бази даних.

5. Реалізувати алгоритм роботи додатку.

6. Протестувати розробку та провести остаточне налагодження

7. Оформити результати роботи у вигляді пояснювальної записки, що

відповідає стандартам підприємства щодо оформлення випускних робіт.

4. Календарний графік

| № етапу | Зміст | Термін виконання | Готовність (%), підпис керівника | Підпис керівника про повну готовність етапу, дата |
|---------|---|---------------------|----------------------------------|---|
| 1 | Збір практичного матеріалу за темою | 16.01.23-17.02.23 | | |
| 2 | I атестація. I розділ випускної роботи | 27.03.23-01.04.23 | | |
| 3 | II атестація. II розділ випускної роботи | 01.05.23-06.05.23 | | |
| 4 | III атестація. III розділ випускної роботи, висновки та рекомендації, додатки, реферат. | 29.05.23-03.06.23 | | |
| 5 | Перевірка випускної роботи на оригінальність | 15.05.23-12.06.23 | | |
| 6 | Доопрацювання випускної роботи, підготовка презентації, отримання відгуку керівника та рецензії | 05.06.23-10.06.23 | | |
| 7 | Попередній захист випускної роботи | 12.06.23-18.06.23 | | |
| 8 | Подача випускної роботи на кафедру | за 3 дні до захисту | | |
| 9 | Захист випускної роботи | 19.06.23-24.06.23 | | |

Дата видачі завдання: 22 лютого 2023 р.

Керівник випускної роботи _____
(підпис)

Сабанов С.О.

Завдання прийняв до виконання _____
(підпис студента)

Мельник І.М.

РЕФЕРАТ

Випускна робота молодшого спеціаліста містить 46 сторінок, 18 рисунків, 2 формули, 17 бібліографічних посилань.

Об'єкт розробки – програмна симуляція замкненої екосистеми.

Метою роботи є розробка додатку, що моделює екосистему акваріуму.

У випускній роботі розглядаються поняття та устрій екосистем, зокрема замкнених водних екосистем типу акваріуму, а також інструментальні засоби створення подібних ресурсів.

Детально описано процес розробки додатку на базі моделі поведінки водної екосистеми з урахуванням особливостей функціонування типових флори та фауни акваріуму.

В якості рушія в роботі застосовано Unity, система керування базою даних – SQLITE. Мовою програмування обрано C#.

3D, SQLITE, UNITY, БД, АКВАРІУМІСТИКА, ЕКОСИСТЕМА, СИМУЛЯЦІЯ,
СКБД

ЗМІСТ

| | |
|--|----|
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ | 7 |
| ВСТУП | 8 |
| РОЗДІЛ 1 КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ ЕКОСИСТЕМ..... | 10 |
| 1.1. Характеристика екосистем. Їх види | 10 |
| 1.2. Водні екосистеми | 12 |
| 1.3. Акваріум. Акваріумістика..... | 13 |
| 1.4. Моделювання екосистем | 20 |
| 1.5. Огляд комп'ютерних симуляцій водних екосистем. Аналоги..... | 23 |
| 1.6. Висновки за першим розділом..... | 27 |
| РОЗДІЛ 2 ІНСТРУМЕНТАЛЬНІ ЗАСОБИ СТВОРЕННЯ СИМУЛЯТОРА ЕКОСИСТЕМИ | 28 |
| 2.1. Платформа розробки в реальному часі та рушій Unity..... | 28 |
| 2.1.1. Основні відомості про платформу | 28 |
| 2.1.2. Технологія створення додатків у середовищі Unity..... | 29 |
| 2.2. Система керування базою даних SQLite | 33 |
| 2.3. Інтегроване середовище розробки Microsoft Visual Studio | 34 |
| 2.4. Мова програмування C#..... | 37 |
| 2.5. Висновки за другим розділом | 37 |
| РОЗДІЛ 3 РЕАЛІЗАЦІЯ СИМУЛЯТОРА «АКВАРІУМ» | 39 |
| 3.1. Призначення та функціональні можливості додатку | 39 |
| 3.2. Блок-схема та алгоритми роботи..... | 40 |
| 3.3. Структура бази даних | 41 |
| 3.4. Основні програмні модулі..... | 42 |

| | |
|----------------------------------|----|
| 3.5. Інтерфейс користувача | 43 |
| ВИСНОВКИ..... | 44 |
| ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ..... | 45 |

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

| Слово / словосполучення | Скорочення |
|------------------------------------|------------|
| І | |
| Integrated Development Environment | IDE |
| U | |
| Unity Graphical User Interface | uGUI |
| User Interface | UI |
| Б | |
| База даних | БД |
| С | |
| Система управління базою даних | СУБД |
| Система керування базою даних | СКБД |

ВСТУП

Неолітична революція охрестила нову віху у історії людства — перехід від привласнювального господарства до відтворювального. Тобто, від застосування існуючих екосистем, людина перейшла до побудови та моделювання власних.

Із плином часу прості посівні ниви перетворюються на багатогектарні поля; малі загони на автоматизовані комбінати; малі пруди на величезні аквакультурні ділянки. Розвиток технологій невинно розширює масштаб та складність антропогенних систем. Цей розвиток, також, призводить до більших витрат на реалізацію останніх. Величезна коштовність таких процесів та масштабність впливу таких систем на натуральні створює попит на моделювання останніх. Окрім того для розвитку ефективності антропогенних систем є доцільне моделювання натуральних.

Сучасні досягнення агро-, аквакультур та тваринництва дозволяють вдовольняти харчові потреби більшої частини населення земної кулі. Але зістаються естетичні потреби. Для їх вдоволення існують парки, заповідники, зоопарки тощо. Зокрема, існують антропогенні екосистеми, що відносно нові і досить популярні – акваріумні.

Акваріумістика досить популярне і цікаве заняття, що зокрема має низку позитивних факторів для акваріуміста та атмосфери приміщення. Зокрема, мають заспокоюючий та антистрес-ефект. [1] Проблеми акваріумістики, зокрема, при створенні і підтриманні екзотичних екосистем – коштовність та складність підтримки системи. Тому існує потреба у попередньому моделюванні екосистеми.

Із розвитком цифрових технологій, акваріумістика почала розвиватися у напрямку діджиталізації: від моніторингових та автоматичних систем до повної віртуалізації акваріума. І ця віртуалізація існує, зазвичай, як проста заміна

звичайному акваріуму, без симуляційних факторів. Але існує ніша під саме симуляційний продукт, що може бути корисним як для досвідчених акваріумістів, так і для зовсім новачків. Такий продукт допоміг би досвідченим зекономити гроші моделюванні їх акваріуму мрії чи зовсім міг би його замінити; а для новачків у справі така програма стала б корисним помічником та цікавими «дверима» у світ сучасної акваріумістики.

У випускній роботі реалізовано проект, що являє собою комп'ютерну симуляцію акваріуму. Модель враховує основні особливості поведінки флори і фауни замкненої водної екосистеми.

РОЗДІЛ 1

КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ ЕКОСИСТЕМ

1.1. Характеристика екосистем. Їх види

Екосистема – базова структурна одиниця в біосфері; взаємодіюча сукупність живих організмів та їх оточення, середовища. Хоча концепція взаємодії неживої та живої природи існує ще з античних часів, саме сучасний термін «екосистема» був висунутий англійським вченим Артуром Тенслі [2].

Роздивляючись екосистему чи «біогеоценоз» глибше ніж просту взаємодію біоценозу та еко/геотопу, ці два складники зазвичай поширюють на:

- фітоценоз – сукупність усіх рослин;
- зооценоз – сукупність усіх тварин;
- мікроценоз – сукупність усіх мікроорганізмів.

Агентів біоценозу, також , можливо розкласти, за функціональною ознакою у екосистемі, на:

- продуцентів – автотрофні організми, що синтезують неорганічні сполуки у органічні;
- консументів – гетеротрофні організми, що трансформують та змінюють органічні матеріали;
- редуцентів – зазвичай, сапротрофні організми, що конвертують органічні сполуки у неорганічні. [2]

Екотоп, зазвичай, поділяють на атмосферу (кліматоп) та почву (адафотоп). За ознаками екотопу, зазвичай, екосистеми класифікують на біоми, бо саме він диктує кліматичні та, як наслідок, біоенергетичні та біологічні властивості.

Важливою характеристикою будь-яких екосистеми є гомеостаз, або ж стійкість. Ця характеристика відповідає за можливість до відновлення структури екосистеми із плином часу. Гомеостаз системи прямолінійно залежить від біорізноманітності останньої. Висока різноманітність розширює трофічний ланцюг, що зменшує «вагу» окремого виду у збереженні стійкості екосистеми. Гомеостаз екосистем традиційно поділяють на два види [3]:

- резистентний – стійкість системи характеризуються збереженням структури екосистеми підчас впливу ззовні;
- пружний – стійкість системи характеризується можливістю відновлювати свої структури під впливу ззовні.

Екосистеми, зазвичай, поділяють на дві великі групи за походженням: природні та штучні.

Природні екосистеми створені у результаті дій сил природи. Дуже важливими ознаками та відмінностями природніх екосистем від штучних є повний та замкнутий колообіг речовин та досить висока стійкість. Таким екосистемам притаманна строга ієрархія – у лісовій екосистемі за ярусами, в водній за глибиною. На відміну від штучних, природні системи формуються за дуже довгий період часу, а досягнення клімаксу може зайняти століття. Такий процес формування і зумовлює високу стійкість системи. [4]

Штучні екосистеми створені людиною та існують для експлуатації ресурсів системи для вдоволення потреб. Дуже важливою і частою ознакою «штучності» екосистеми є її гетеротрофність. Тобто, більша частина штучних систем потребує регулярного втручання з боку людини для збереження існування середовища та біотопу. Штучним системам також притаманний незамкнутий колообіг речовин, що є наслідком гетеротрофності та сутності існування системи. Антропогенні екосистеми, також, мають низький біорізновид і, як наслідок, високу волатильність (низьку стійкість) [4].

1.2. Водні екосистеми

Екосистему вважають водною, якщо її кліматоп здебільшого представлений водою. Також характерною ознакою таких систем є дуже висока волатильність та крихкість.

Головними факторами впливу водних систем можна виділити:

- температуру середовища;
- хімічний склад середовища;
- вміст солей;
- прозорість середовища;
- концентрація розчиненого кисню.

Водні екосистеми за солоністю, традиційно, поділяють на морські та прісноводні.

Прісноводні екосистеми займають лише $\sim 0.8\%$ від загальної поверхні планети та лише $0,009\%$ від загального обсягу води. Швидкість течії та її наявність є важливим класифікаційним фактором у прісноводних системах. За цією ознакою виділяють стоячі (лентичні) та проточні (лотичні) системи. Серед них, також іноді виділяють заболочені системи.

Стоячі системи зазвичай представлені ставками, озерами, водосховищами, тощо.

Для стоячих є характерним стратифікація біотопу в залежності від слою води. Літораль представлена спільнотами планктону, амфібій, водоплавних птахів. Профундаль, здебільшого, населений хижими рибами. Бенталь – придонними рибами, молюсками. [5]

1.3. Акваріум. Акваріумістика

Окремим та досить поширеним типом штучної водної екосистеми є «акваріуми». Акваріум — будь-який штучний резервуар, що створений для спостереження за його екосистемою та обладнаний для її підтримки.



Рисунок 1.1. — Рифовий домашній акваріум, без кришки, з УФ освітленням.

Акваріуми, як антропогенна екосистема, зазвичай, виконують естетичну функцію. Термін «акваріум» сформував один з перших акваріумістів — англійський натураліст Філіп Генрі Госсє. Принципи функціонування акваріуму,

як екосистеми, сформував співвітчизник Госсе, хімік Роберт Варінгтон у 1950-му році. [6]

Щодо класифікації, акваріуми, як інші водні екосистеми, поділяють на дві широкі групи:

- 1) прісноводні — акваріуми, що підтримують існування прісноводної екосистеми та її агентів (найбільш розповсюджені);
- 2) солоні — акваріуми, що емулюють солоне водне середовище.

Інколи, виділяють третій тип — «солонуваті» (кальк. англ. brackish), що створюють умови, подібні до естуаріїв або мангрових заростів.

За температурою можна виділити два типи акваріумів:

- 1) холодноводні — акваріуми, у яких температура води не підігривається і знаходиться на рівні кімнатної або близько до неї;
- 2) тропічні — акваріуми, які потребують підтримки підвищеної температури середовища для його існування.

Заняття, що включає проектування акваріумів, створення та підтримку екосистеми та її гомеостазу, називається «акваріумістика». Поняття акваріумістики, термінологія та принципи були сформовані лише на початку другої половини 19-го століття на фоні відкриття першого публічного акваріуму у Лондонському зоопарку та публікацією Госсе першого в світі мануалу з популярного у сучасності хоббі. [6]

Не дивлячись на відносну новину акваріумістики, бажання людини утримувати та споглядати водну флору та фауну є досить давнім.

Ще 5–6 тис. років до н.е. у районі сучасного Єгипту були знайдені писемні докази декоративного рибоводства у ставках. На території Помпеї були знайдені залишки риб у кімнатних басейнах та фресках, що свідчить про зацікавленість давніх людей акваріумістикою ще у античні часи.[7]

Найбільшого поширення давня акваріумістика набула на території Азії. У давньому Китаї, перший імператор династії Мін заснував порцелянову компанію, що займалася створенням порцелянових «акваріумів» для золотих рибок (які пізніше стануть найбільш популярною акваріумною рибкою). Акваріумістичний бум та популярність заняття у Європі станеться лише у другій половині 19-го століття.[6]

Структурно сучасний акваріум зазвичай складений із самого резервуару та обладнання для підтримки гомеостазу екосистеми.

Зазвичай, резервуари у домашніх акваріумах мають форму прямокутного паралелепіпеда, рідше сфери. Форма паралелепіпеду, зазвичай, зумовлена простотою виготовлення та зручністю розташування обладнання у такій формі та масштабування. Також, така форма дозволяє розташовувати ребра жорсткості для зменшення вірогідності механічної деформації стінок. Матеріалом для стінок резервуару акваріуму зазвичай виступає скло, акрил, або інший хімічно неактивний матеріал. Кришка виготовляється з термостійких матеріалів та має позиції для утримання освітлювального обладнання. Акваріумна накривка також має щільно з'єднуватися з стінками, щоб забезпечити мінімальну вірогідність контамінації та пришвидшеного випаровування води. Для акваріумів з високим об'ємом виготовляють піддони з пружних матеріалів для зменшення механічного тиску на дно резервуару.

Для збереження гомеостазу екосистеми та забезпечення придатного для існування життя середовища в акваріумі необхідно встановлення спеціального обладнання [8].

Мінімальний набір для акваріума, зазвичай, складений з:

- системи фільтрації та аерації;
- системи слідкування за показниками акваріума;
- системи освітлення.

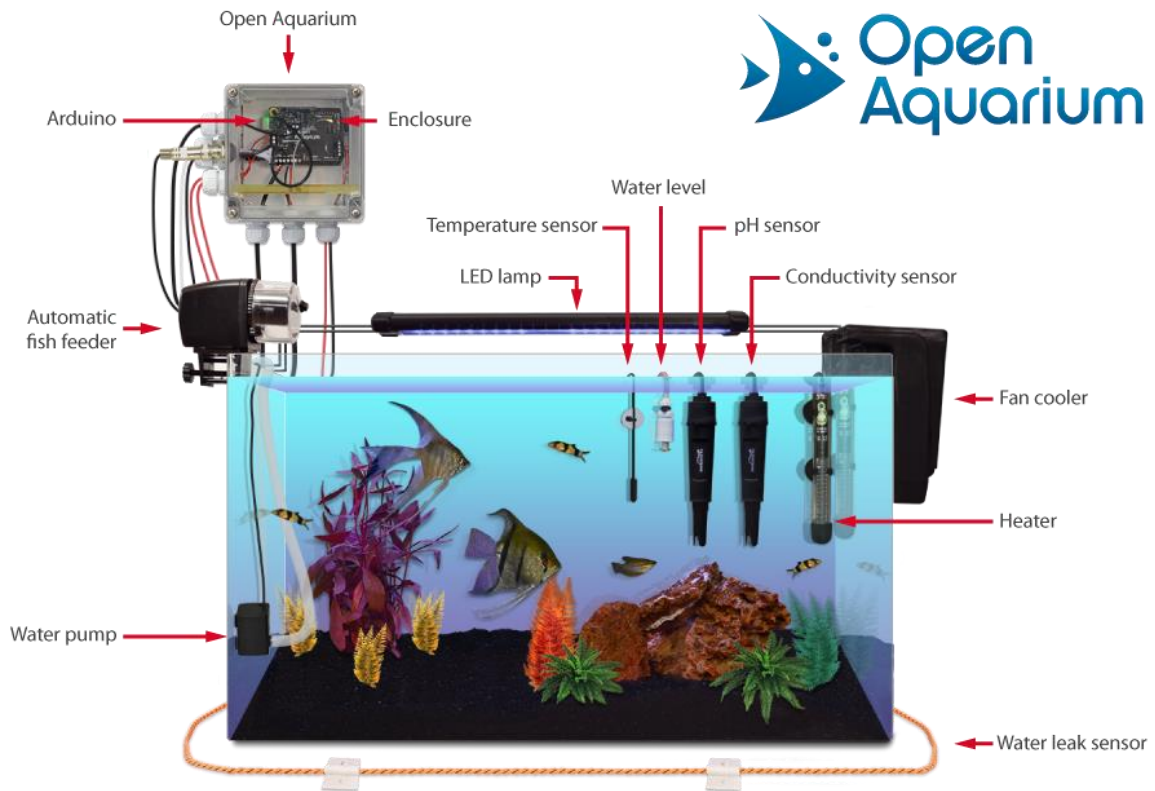


Рисунок 1.2 — Технічно обладнаний акваріум системою «Open Aquarium»

Система фільтрації та аерації для домашніх акваріумів у більшості випадків представлена фільтром для біологічної та механічної очистки води та іноді кисневим компресором (аератором). Фільтр (фільтраційний компресор) є необхідним для будь-якого акваріума (рисунок 1.3).



Рисунок 1.3. — Різні моделі зовнішніх акваріумних фільтрів

Він очищує воду від суспензії з продуктів життєдіяльності та екологічного циклу. Також, фільтр створює течію, що є необхідною для рівномірного розподілення розчинених речовин у воді та балансування екосистеми. Окрім того, фільтр є осередком розмноження популяцій акваріумних бактерій, зокрема нітрифікаторів, які є важливими для азотного циклу середовища акваріуму [9] (рисунок 1.4).

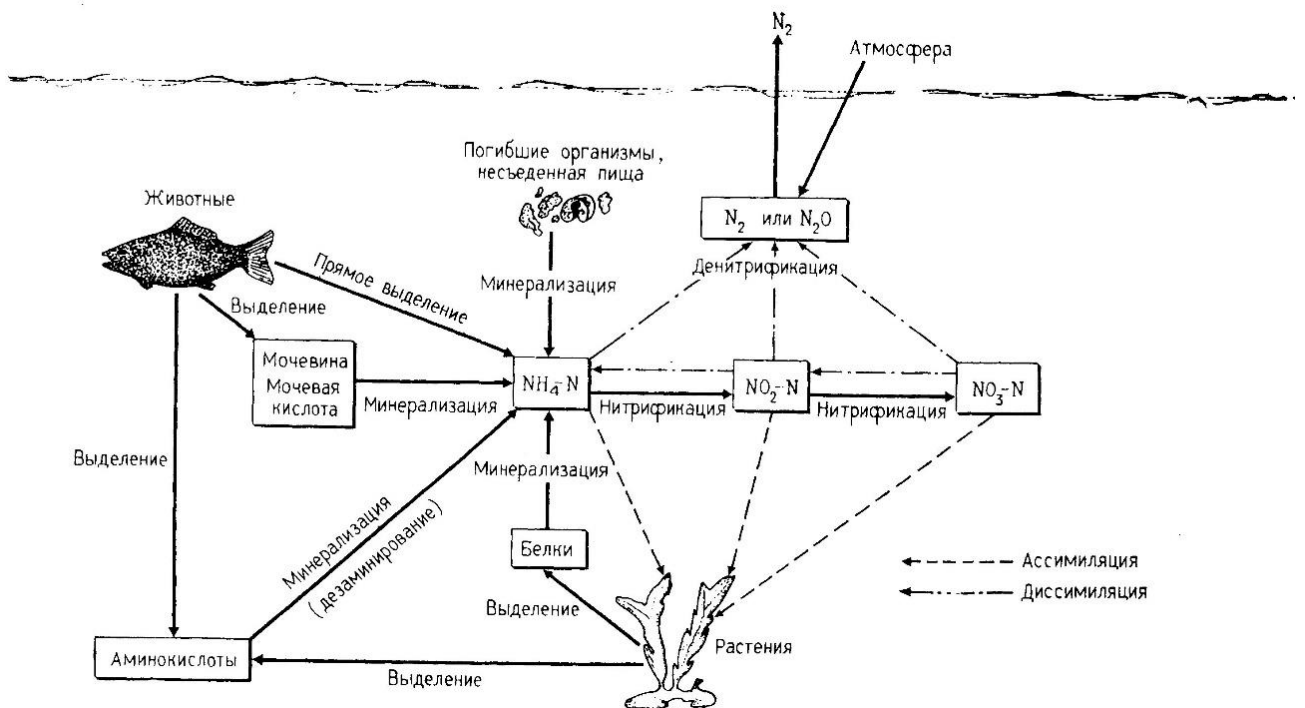


Рисунок 1.4. — Модель азотного циклу у акваріумі

Аератор, зазвичай встановлюється лише у великі акваріуми, або у випадку низької кількості рослинності.

Для підтримки сталих показників середовища у акваріумі встановлюють обладнання для слідкування за ними. Не дивлячись на велику кількість важливих показників якості води, у більшості випадків, аматори-акваріумісти встановлюють лише термометр. Температура води є одним із дуже важливих

показників, який не можливо виміряти «на око». Інколи, встановлюється детектор вуглекислого газу, зазвичай, якщо у резервуар організована його штучна подача. Рівень вуглецю є важливим параметром, що впливає на здоров'я рослин та придонних мешканців акваріуму. Для дуже точного контролю можна встановити повноцінну систему детекторів, що буде відстежувати: рівень води, кількість розчинених азотних сполук, жорсткість, кислотність, рівень освітленості та купу інших параметрів. Але таке обладнання має сенс лише у рифових акваріумах чи акваріумах малої місткості.



Рисунок 1.5. — Система моніторингу акваріуму

Важливим елементом «добробуту» акваріуму є світло. Для освітлення застосовують флуоресцентні лампи, що закріплюються або на кришці, або на спеціальній фермі. Інколи додатково до флуоресцентного освітлення встановлюють ультрафіолетове освітлення для зменшення популяції мікроорганіки. Майже постійним компаньйоном освітлення у акваріумі є таймер,

що допомагає і економити електроенергію, і зберігати сталий режим день/ніч у акваріумі.

Не дивлячись на наявність автоматичних систем очистки, акваріумне середовище будучи антропогенною екосистемою потребує і ручного втручання. Звісно, розміри резервуару, якість автоматичних систем фільтрації та біотоп системи будуть детриментально впливати на частоту та типи втручання, але не скасують потребу у ньому.

Оптимально, акваріуміст має щотижнево замінювати близько 10% відсотків води з об'єму екосистеми. Звісно, ця норма не є догматичною, але є базовою для більшості домашніх акваріумів. Окрім того, внутрішні фільтри дуже часто не в змозі зібрати у себе всі частки суспензії, тому вони можуть осідати в субстраті. Будь-які органічні залишки в акваріумі є досить великою загрозою для азотного балансу акваріуму, тому проводять «сифонування» субстрату акваріуму. Тобто, за допомогою спеціального сифону із насадкою (рисунок 1.6) вручну фільтрують гравій, чи інший наповнювач.



Рисунок 1.6. — Звичайний ручний сифон для акваріума

Інколи, зокрема, у наслідок азотного дизбалансу, на стінках акваріума може утворюватися наліт або з'являться некультивована аквафлора. У таких ситуаціях застосовують спеціальні магнітні щітки, скребки (які інколи споряджені лезом).

1.4. Моделювання екосистем

Екосистемна модель – абстрактне представлення екологічної системи, що має на меті вивчення її функціонування. Екосистемні моделі застосовуються у широкій предметній області – від екотоксикології, вчення про ефекти впливу токсинів на екосистеми, до агрикультурних напрямів. Для створення подібних моделей застосовують екологічні відношення, як наприклад «хижак жертва».

Математичне моделювання, також, дає змогу реплікувати та симулювати екосистеми (рисунок 1.7).

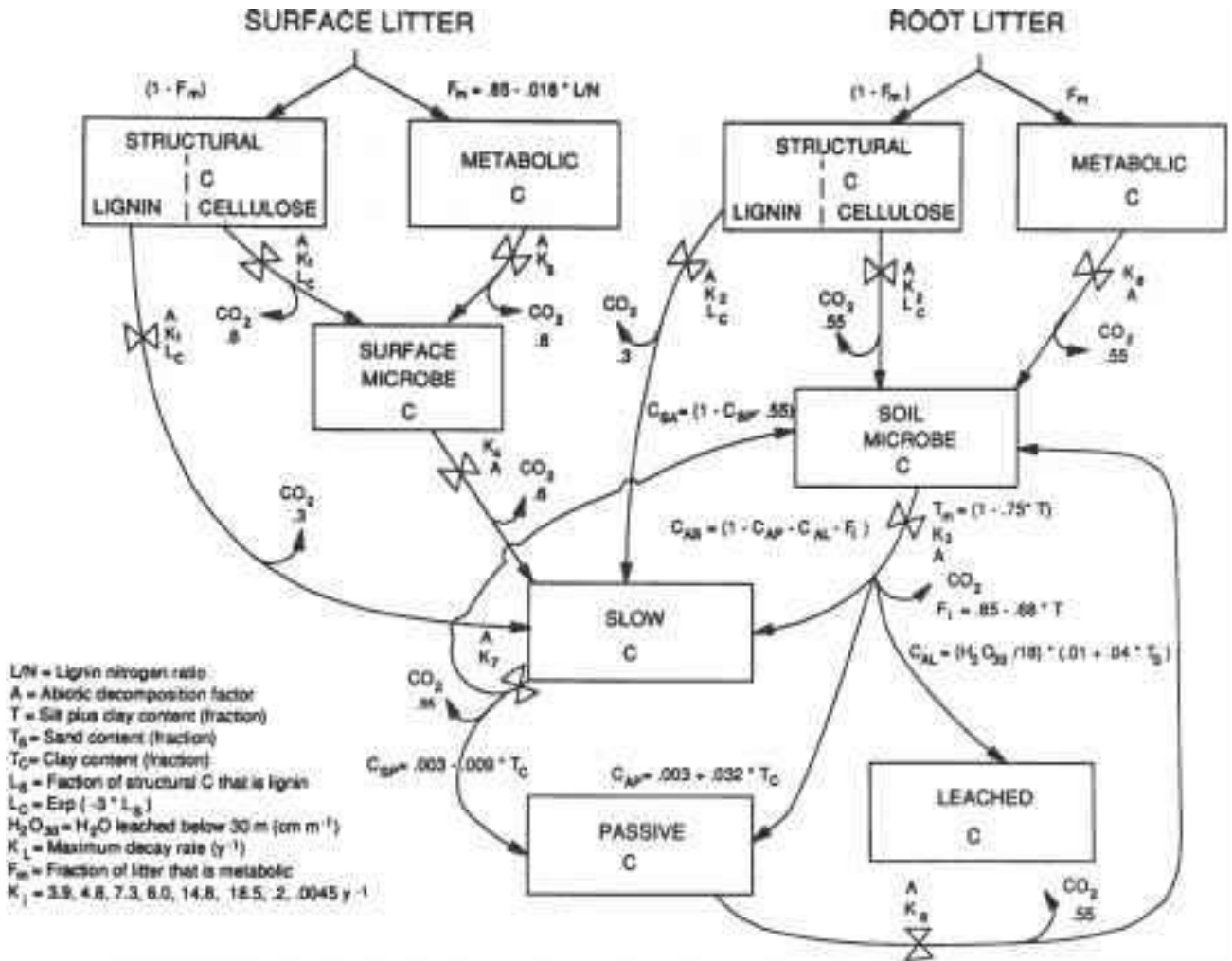


Рисунок 1.7. — Екосистемна модель вуглецевого обміну у ґрунті

Це, у свою чергу, знижує ресурсну та часову коштовність експериментів та дозволяє проводити їх без загрози реальній екосистемі. Більш того, дослідження кореляції між реальною екосистемою та її моделлю дозволяють знаходити нові екосистемні відношення, або поліпшувати розуміння існуючих.

Першим та найпоширенішим прикладом екологічної моделі, зокрема «хижак-жертва», є рівняння Лотки-Вольтерри. Ця екологічна модель представлена двома звичайними диференційними рівняннями, що наведені нижче.

$$\frac{dx}{dt} = \alpha x - \beta xy, \quad (1.1)$$

$$\frac{dy}{dt} = \delta xy - \gamma y, \quad (1.2)$$

де x , y — густота популяції жертви та хижака відповідно, α — максимальний темп росту жертви на одиницю популяції, β — ефект присутності хижака на темп росту жертви, γ — смертність на одиницю популяції хижака, δ — ефект присутності жертви на темп росту хижака.

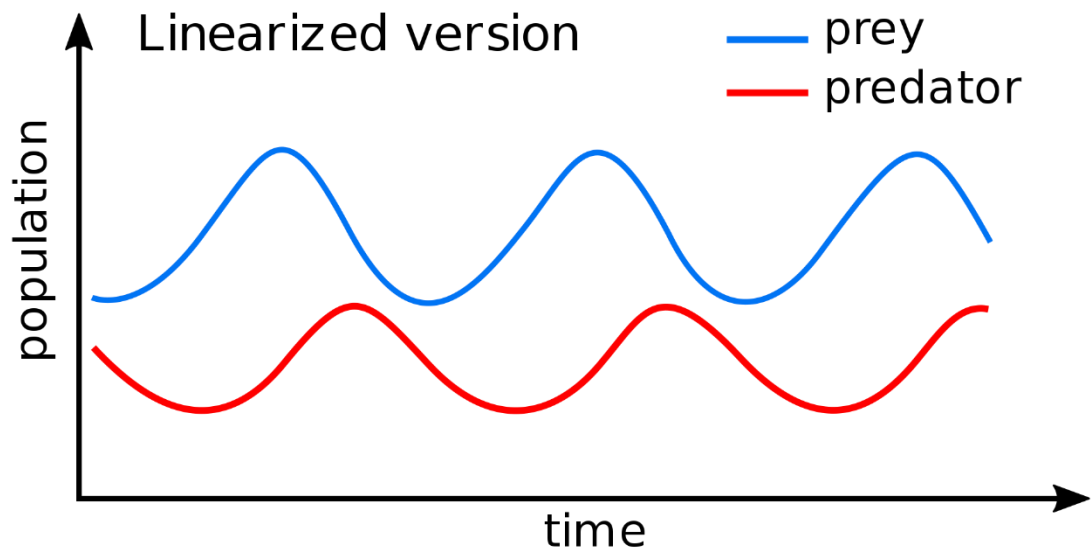


Рисунок 1.8. — Графік лінійного рішення рівнянь Лотки-Вольтерри

Початково модель була сформована для пояснення флуктуації популяцій риби та акул у Адріатичному морі після заборони на їх вилов. Але через простоту і відносну точність, застосовується для пояснення будь-якого відношення подібного типу [10].

1.5. Огляд комп'ютерних симуляцій водних екосистем. Аналоги

Для розробки будь-якого продукту, дуже важливий аналіз аналогів. Такий аналіз допомагає більш тонко дослідити ринок та усунути ймовірні помилки ще до початку розробки.

«FISHERY» — стилізована гра-симулятор акваріуму у ранньому доступі. «Використовуйте свій творчий потенціал для створення та підтримки красивих водних середовищ існування. А потім розслабтеся і спостерігайте, як ваші прекрасні рибки насолоджуються своєю домівкою». Розробляється компанією шведською інді-компанією «Lilou Studios» з міста Цюріх. [11]



Рисунок 1.9. — Аналог «FISHERY»

Гра досить казуальна і дозволяє дійсно розслабитись та реалізувати свої акваріумістичні фантазії, а стилізована графіка естетична та доповнює релаксовану атмосферу.

Свій віртуальний акваріум можна наповнити різними видами риб, яких у сумі більше 15-и. Більш того, у грі відображене явище полового диморфізму, що ще більш розширює різноманіття риби. Окрім риб у акваріум можливо «заселити» різною рослинністю (як водорослями, так і мхами) та креветкою. Але

гра дуже сильно спрощує симуляційний аспект із просуванням процесу розробки. Екосистема може зберігати баланс навіть у ситуації із перевищенням норми розчиненого аміаку у резервуарі, а риби виживають навіть у ситуації з майже повною відсутністю розчиненого кисню. Окрім того, гра, фактично, підтримує лише створення прісноводних акваріумів. Рифові та солоні варіації не мають достатньої кількості видів біотопу та ігросимуляційних механік.

Для керування грою здебільшого застосовується миша та інколи клавіатура. Інтерфейс є естетичним, але не дуже зручний для постійного користування. Гра підтримує лише Windows платформу.

Сильні сторони:

- дуже приємне естетичне оформлення;
- велика кількість симульованих шаблонів поведінки риби;
- наявність широкого вибору агентів біо- та екотопу.

Слабкі:

- слабкий симуляційний аспект;
- підтримка додатком лише Windows;
- незручний інтерфейс.

Аналог «Biotop» є повною протилежністю «FISHERY», з точки зору ігросимуляції. «Biotop» — «...стане найбільш деталізованим та реалістичним акваріумним симулятором на ринку. Biotop дозволяє створювати комплексні водні екосистеми та симулювати реалістичні життєві цикли рослин та риб». «Biotop» розроблений німецькою компанією «MBL Development», фінансованою європейським фондом регіональної розробки EFRE.[12] Гра відразу відповідає стереотипам німецької ігрової розробки, віддаючись повністю нюансам ігромеханічної розробки та симуляційної точності.



Рисунок 1.10. — Аналог «Biotopie»

Естетично, гра не має унікального графічного стилю та намагається виглядати реалістично, через що страждає. Параметризація акваріуму є досить бідною та дозволяє обрати лише 3 заздалегідь готові варіанти, з яких два — тропічні. Ігровий цикл базується на виконанні завдань на розмноження риби. Окрім того, гра дуже сфокусована на водному хімічному балансі, через що більша частина ігрового циклу складається з мікроменеджменту його параметрів, що є нецікавим та нереалістичним у випадку з форм-фактором наданих резервуарів. На додачу, проект застиг на ранній стадії розробки та фактично є «мервим».

Додаток може бути встановлений на Windows та MacOS, керування виконується переважно мишкою, клавіатура майже не застосовується.

Сильні сторони:

- наявність симуляції розмноження риби;
- підтримка роботи на багатьох ОС.

Слабкі сторони:

- гіперреалістичність симуляційного аспекту;
- відсутність довільного налаштування резервуарів акваріуму.

«Fish Farm 3» є дуже наочним прикладом більшості схожих за функціоналом програм. Dodatok розроблений компанією «BitBros Inc» є

«останнім поколінням акваріумістичної популяції — придбайте, розмножуйте, схрещуйте, продавайте та споглядайте за величезним вибором прісно-, солоноводної риби, у тому числі морськими кониками, дельфінами, китами, акулами, морськими зірками, медузами, скатами та черепахами». [13]



Рисунок 1.11. — Fish Farm 3

Аналог має дуже широкий вибір довільних декорацій до акваріуму, та його населення. Але фактично є ближче до казуальної аркади, бо симуляційний аспект не є центральним у реалізації та не потребує майже будь-яких дій від користувача. Естетично додаток не виділяється і не викликає ніяких емоцій.

Розроблений аналог підтримує усі доступні мобільні платформи: iOS, Android та Windows Phone. Керування у додатку реалізовано дотиками пальців.

Сильні сторони:

- широкий вибір населення акваріуму;
- підтримка багатьох мобільних ОС;
- висока візуальна інтерактивність.

Слабкі сторони:

- банальний візуальний стиль;
- низька симуляційна глибина.

1.6. Висновки за першим розділом

У розділі було розглянуто основні поняття, стосовно екосистем, їх структуру, внутрішню взаємодію складових та тенденції. Окремо було розібрано водні екосистеми із їх специфікою, також було розглянуто унікальний антропогенний підтип такої екосистеми — акваріум. Оглянуто основні пункти та параметри, що будуть потребувати реалізації у додатку.

Було оглянуто явище та заняття екосистемного моделювання. Інформація з цього підрозділу стала дуже в нагоді у момент реалізації функціоналу додатку.

Достатньо велику кількість уваги було приділено освітленню устрою акваріуму та акваріумістичному промислу, його специфіці та історії. Були приведені приклади необхідного обладнання для сучасного акваріуміста та рекомендації застосування тих чи інших варіантів.

Було розібрано декілька аналогів запланованого до розробки додатку. Оглянуто представлений функціонал аналогів та зроблено висновки щодо реалізації власного.

Із оглядом зібраної та представленої у розділі інформації було сформовано функціональні вимоги до продукту та оцінена доцільність його розробки. Окрім того, зібрана інформація надала думки щодо технологій що будуть обрані та розглянуті у наступному розділі.

РОЗДІЛ 2

ІНСТРУМЕНТАЛЬНІ ЗАСОБИ СТВОРЕННЯ СИМУЛЯТОРА ЕКОСИСТЕМИ

2.1. Платформа розробки в реальному часі та рушій Unity

2.1.1. Основні відомості про платформу

Unity – платформовий комплекс, що включає у себе ігровий рушій, інтегроване середовище розробки та програмний каркас. Цей комплекс розробляється компанією Unity Technologies з 2005 року. На момент появи, платформа була націлена на розробку ігор для MacOS X, але на зараз комплекс можливо застосовувати і для кроссплатформової розробки.

Unity дозволяє розробляти програми для більшості операційних систем (Windows, Unix-подібні [Linux, Android, iOS, MacOS]) та апаратних платформ (ІВМ-сумісні комп'ютери, ігрові приставки 8–9 поколінь, мобільні пристрої).

Не дивлячись на те, що платформа розповсюджується під назвою «середовище розробки комп'ютерних ігор», Unity є популярним та доступним вибором для будь-яких задач, що потребують комп'ютерної візуалізації. Наприклад, блокбастер Джеймса Кемерона «Avatar 2» або ж «Аватар: шлях води» застосовував пропріетарні інструменти Unity — «Unity Wētā Tools» для створення комп'ютерної графіки та візуальних ефектів, і на момент виходу отримав численні нагороди, у тому числі за візуальні ефекти.[14] Автомобільна компанія Volvo широко застосовує Unity для проектування та поєднання процесів виробництва. [15] Також Unity обирають для розробки тренувальних комплексів

у віртуальній реальності. Але найбільша частка користувачів продуктами Unity припадає саме на розробників ентузіастів.

2.1.2. Технологія створення додатків у середовищі Unity

Створення додатку у Unity починається з створення Unity проєкта за допомогою компонента Unity Hub, де обирається його тип або ж «трафарет». Тип проєкту диктує апаратну платформу та операційну систему на якій буде можливо виконання готового додатку. Від типу залежить і набір пакунків за замовчуванням. На вибір доступні дву- та тривимірні типи проєктів, з їх «Unity Universal Renderer» варіантами та типи під конкретні апаратні платформи, ігровий жанр. Також, доступні навчальні проєкти, що вже містять готові компоненти. Окрім того, навчальні проєкти мають закріплені глави у Unity Manual (документації до продукту) із наочними прикладами реалізацій та найкращих практик розробки.

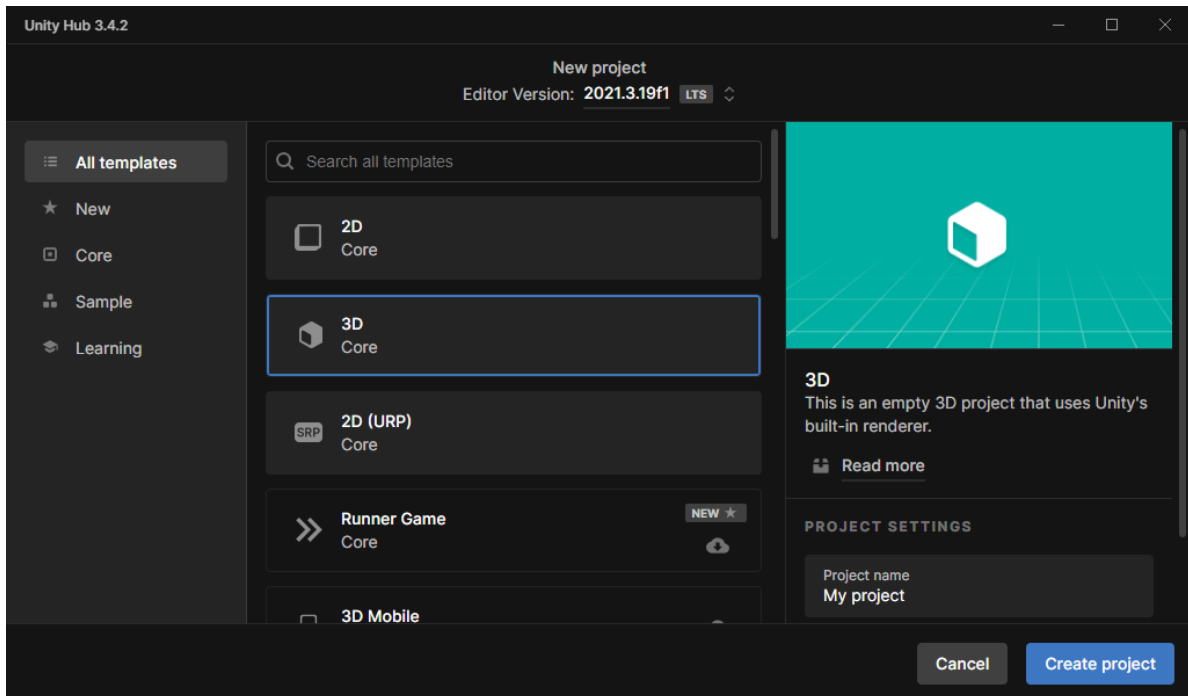


Рисунок 2.1. — Інтерфейс компоненту Unity Hub на етапі створення нового проекту

Далі, можна обрати розміщення файлів проекту на диску та конкретну версію редактора (компонента Unity Editor), що буде застосована. Вибір версії редактору є досить важливим кроком у створенні проекту, бо різні версії редактору підтримують різні версії самого рушію. Це у свою чергу може ускладнити процес підтримки додатку у майбутньому, чи зовсім поставити хрест на можливості його подальшого розвитку. Створення проекту займає деякий час, після закінчення цього процесу з'являється вікно з Unity редактором обраної раніше версії.

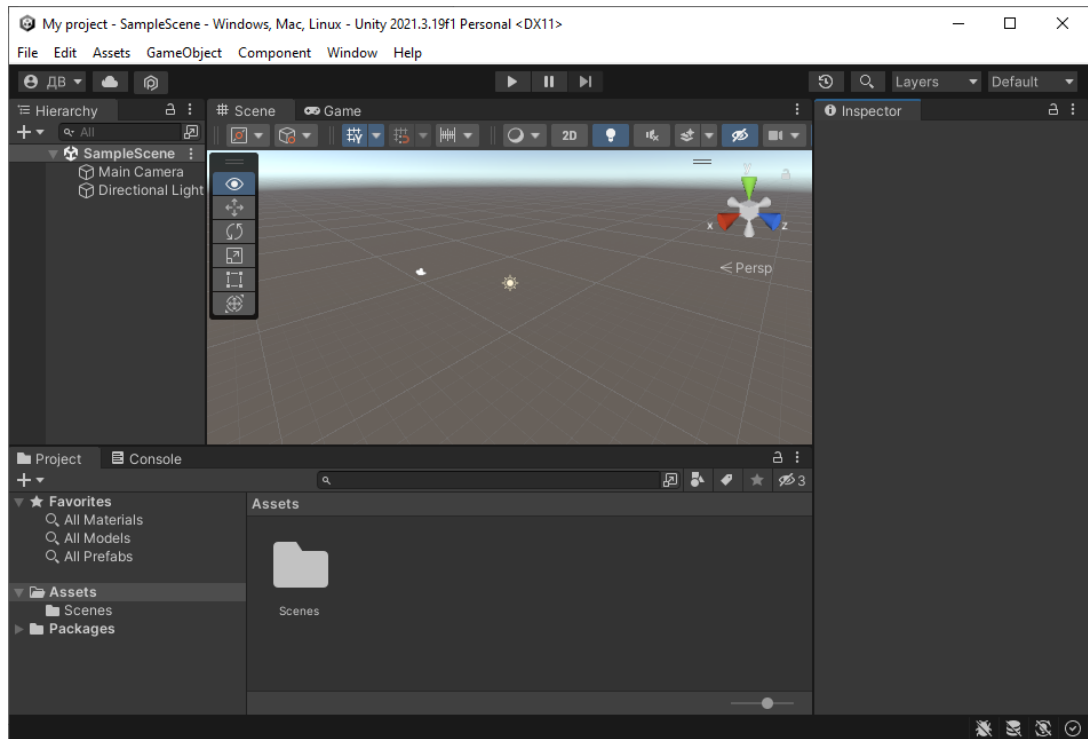


Рисунок 2.2. — Інтерфейс редактора Unity Editor (2021.3.19f1) для новоствореного тривимірного проекту

Інтерфейс редактору Unity Editor пережив багато ітерацій, що покращило його інтуїтивність та естетичність, але завжди мав сталий набір основних компонентів.

Першим компонентом інтерфесу є «Toolbar», панель інструментів, що знаходиться у верхній частині робочої області. Ця частина інтерфесу виконує керування відтворенням сцени, збереженими трафаретами графічного інтерфесу та компонентом Unity Services. Останній відповідає за вбудовання аналітичного функціоналу у додаток, застосування хмарного сервісу Unity Cloud Build та імплементації Unity Multiplayer. «Toolbar», також, являється єдиним компонентом графічного інтерфесу користувача, що не може змінювати свою позицію.

По замовчуванню, у верхній лівій частині робочої області знаходиться «Hierarchy window», вікно ієрархії. Unity ієрархія — це графічне відображення

Unity GameObjects (екземплярів сутностей) на сценах, їх взаємин та типів відображення. Також вікно ієрархії дозволяє вручну створювати екземпляри сутностей та налаштовувати їх відносини.

У центрі містяться два компонента, що у задіяній версії редактора займають одну площу робочої області — «Game view» та «Scene view», огляд «гри» та сцени відповідно. Обидва компоненти відповідають за візуалізацію зовнішнього виду додатку. Різниця лише у тому, що огляд гри емулює повністю скомпільовану версію робочого додатку. У той час як огляд сцени дозволяє займатися ручним редагуванням змісту сцени та оглядати його у абстракції від виконавчого процесу, тобто у де-факто, режимі налагодження. Для редагування у «Scene view» застосовується компонент «Overlays», набір базових інструментів для ручної маніпуляції екземплярами сутностей, зокрема зміни геометричних параметрів.

У правій частині вікна знаходиться «Inspector» або ж «оглядач». «Inspector» реалізує функціонал зручного налаштування екземплярів на сцені. Він дозволяє дуже точно регулювати параметри «GameObject»-ів та налаштовувати та прикріпляти до них виконавчі «скрипти» мовою C#. Також «оглядач» дозволяє налаштовувати «Prefab» елементи.

Нижню частину робочої області виділено «Project window» та «Status bar». Перший (вікно проекту) відображає файлову структуру проекту та задіяні бібліотеки з пакунками, та дозволяє налаштовувати та керувати тими компонентами. Статусний рядок (він же «Status bar») відображає інформацію відлагодження та сервісні оповіщення.

Всі ці компоненти (окрім «Tool bar») Unity Editor'а можуть бути довільно розміщені у робочій області або зовсім прибрані, що надає свободу редагування робочого простору. Це у свою чергу збільшує ефективність розробки додатків.

Окрім того, компоненти редактора досить зручні для користування та покривають усі варіанти потрібної для розробки додатку взаємодії.

2.2. Система керування базою даних SQLite

Для керування БД була обрана вбудована безсерверна система SQLite, а саме її C# ADO.NET інтерфейс «System.Data.SQLite», що був перекомпільований для роботи у середовищі Unity.

SQLite – бібліотека, початково розроблена Річардом Хіппом для мови C, що реалізує малу, швидко та надійну вбудовану СКБД. На сьогоднішній день, бібліотека має безліч API для різних мов програмування та програмних каркасів. SQLite застосовує PostgreSQL як стандарт SQL, за виключенням перевірки типів (SQLite не застосовує її за замовчуванням). Зберігання БД реалізовано файлово, що накладає обмеження на кількість та швидкість обробки запитів на запис. Але, ці обмеження не грають великою ролі, через лише одного користувача БД — програми. Окрім того, через обраний спосіб взаємодії з базою даних, що буде розглянутий у наступному розділі, низька частота запис-запитів не є проблемою ні для розробника, ні для кінцевого користувача.

Саме її було обрано через досить велику кількість користувацької документації, портативність СКБД та БД та достатню, для обраного функціоналу та вимог до програми, швидкодійність. Основним фактором, зіграла саме портативність, бо SQLite не потребує попереднього налаштування серверу або його встановлення на машинах із екземплярами додатку. Це у свою чергу зменшує вимоги до налаштування екземплярів додатку для кінцевого користувача, що є досить великим плюсом для будь-якого сучасного програмного продукту.

2.3. Інтегроване середовище розробки Microsoft Visual Studio

Microsoft Visual Studio – інтегроване середовище розробки, створене компанією Microsoft. Ця IDE дозволяє розробляти як native-додатки, так і web. Visual Studio має вбудовану підтримку інструментарію для розробки під операційну систему Windows та у середовищі .NET. Продукт належить до сімейства Visual Studio та має декілька видань, що відрізняються моделями розповсюдження та функціоналом. Для використання у розробці, було обрано саме «Community» видання, що доступне для завантаження безоплатно, у випадку некомерційної розробки. [16]

Будь-яке IDE має бути зручним для роботи, бо саме у ньому буде проходити найбільше часу для розробника. А Microsoft Visual Studio робить усе для створення якнайкращого середовища.

IDE містить вбудований редактор коду з динамічним автодоповненням та широким спектром користувацьких налаштувань, що дозволяють швидко працювати із кодом. Окрім того, редактор коду має зручний функціонал, для рефакторінгу та підтримки коду, дозволяючи вирішувати ці задачі натиском однієї клавіши.

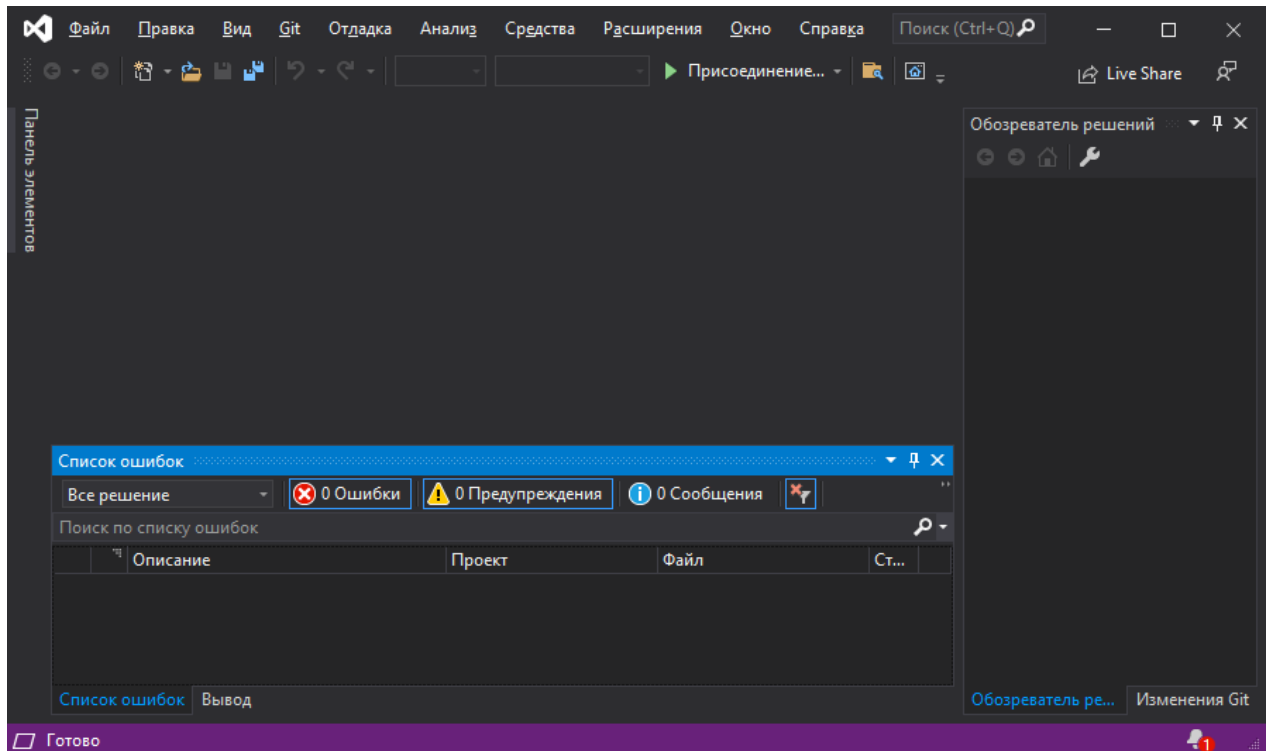


Рисунок 2.3 — Интерфейс Microsoft Visual Studio Community Edition

Якщо автодоповнення IntelliSense не задовольняє усі потреби, то є підтримка макросів, більш того IDE підтримує «Github Copilot». Це автодоповнення на базі технологій штучного інтелекту, яке надає ще більших темпів розробці.

Окрім засобів написання програмного коду, обране IDE має дуже зручні інструменти для відлову винятків та програмних помилок. Вбудована утиліта для відлагодження працює у реальному часі, підтримує точки зупинки і може працювати не тільки на рівні початкового коду, але й на машинному. Ця утиліта, доповнена «рідним» профайлером початкового коду. Це дозволяє швидко отримувати статистичну інформацію про розроблюємий додаток, швидко знаходити слабкі місця програми та оптимізувати початковий код.

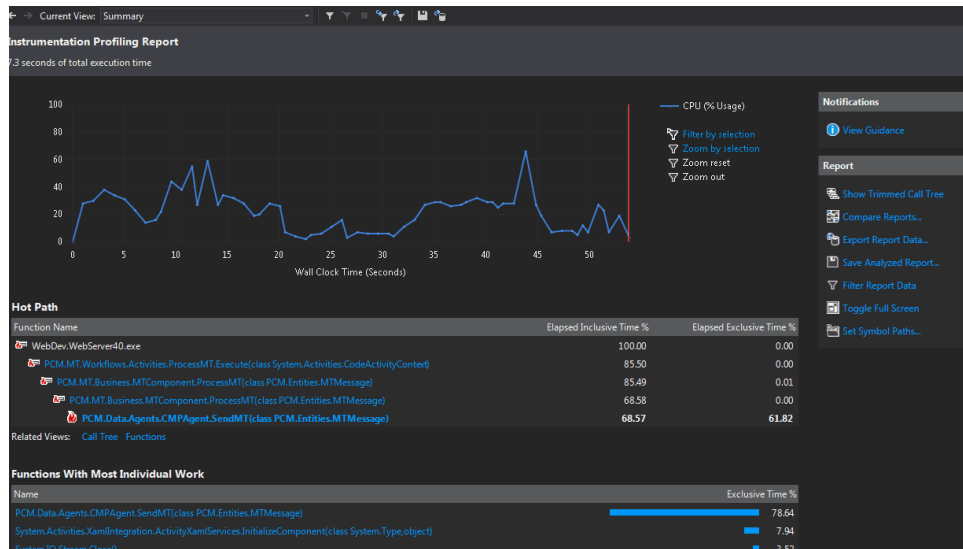


Рисунок 2.4. — Інтерфейс вбудованого в Microsoft Visual Studio профайлеру

На додачу, Microsoft Visual Studio інтегровано підтримує систему контролю версій Git. Це, у свою чергу, надає багато можливостей та спрощує розробку великих додатків, як для розробника одинака, так і для цілих аутсорс компаній.

2.4. Мова програмування C#

Мовою програмування для реалізації проекту було обрано C#. Це мультипарадигмна мова високого рівня, що була розроблена компанією Microsoft у 2000-му році під керівництвом Андерса Хейлсерга. Характерною ознакою мови є застосування та тісна взаємодія із .NET середовищем. [16]

Насамперед, цю мову програмування було обрано через підтримку ООП та вбудовану підтримку рушієм-платформою Unity. Не останню роль зіграла можливість динамічної типізації та наявність «збирача-мусору», що бере на себе основні задачі в управлінні пам'яттю. Ця особливість дозволяє пришвидшити розробку додатку та покращити зрозумілість та можливості рефакторінгу програмного коду.

2.5. Висновки за другим розділом

У цьому розділі було розібрано інструментарій та технології, що були застосовані для розробки симулятора екосистеми.

Зокрема, детально розглянуто як платформу, так і рушій Unity з його компонентами. Досліджено область застосування платформи та рушію Unity та його актуальність. Оглянуто процес розробки додатків та інтерфейс Unity.

Увагу приділено системі керування базою даних SQLite, що буде застосована у проекті. Було оглянуто сильні сторони та причини вибору саме її з поміж аналогів.

Проаналізовано інтегроване середовище розробки Microsoft Visual Studio. Детально оглянутий функціонал середовища, якість роботи у ньому. Обґрунтований вибір саме цього продукту.

Була оглянута застосована для розробки додатку мова програмування C#. Був обґрунтований її вибір та потрібний для розробки функціонал.

За результатами огляду було прийнято рішення терміново перейти до процесу розробки за пайплайном Unity в обраному стеці. Були сформовані принципи та виділений функціонал, що буде дуже детально розглянутий у наступному розділі випускної роботи.

РОЗДІЛ 3

РЕАЛІЗАЦІЯ СИМУЛЯТОРА «АКВАРІУМ»

3.1. Призначення та функціональні можливості додатку

З оглядом на аналоги та означення проблеми, додаток у розробці створюється для акваріумістів-ентузіастів. Програма дозволяє користувачу симулювати віртуальну антропогенну водну екосистему «акваріум». Ціль та призначення додатку — дозволити акваріумісту будь-якого рівня швидко моделювати водну екосистему та її агентів. У свою чергу, це моделювання має давати змогу зекономити гроші та час, що необхідний для макетування подібної існуючої екосистеми у житті.

Згідно призначення, потрібно забезпечити функціональні вимоги до продукту. Ці вимоги мають відповідати призначенню додатку — надавати можливість симулювати водну екосистему, налаштовану за потребами кінцевого користувача. Симуляція водної екосистеми передбачає можливості до налаштування параметрів усіх її складових, як біотопу, так і екотопу.

Зважаючи на усі зауваження, симуляційний продукт, що розробляється має вдовольняти такі вимоги і мати наступний функціонал:

- реалізовувати базову взаємодію з базою даних, де міститься інформація;
- виконувати базове моделювання екологічних агентів.

3.2. Блок-схема та алгоритми роботи

Для симуляції руху риби застосовано простий алгоритм на базі маршрутної точки, що у вигляді блок-схеми зображений на рисунку 3.1.



Рисунок 3.1. — Блок-схема руху риби

Для симуляційного додатку необхідний був алгоритм «існування» риби. Нажаль, у мережі не було знайдено готових рішень, тому було організовано власний. Власний алгоритм застосовує відомі показники води (зокрема, температуру, кислотність, кількість аміаку, нітратів, нітритів), вимоги видів риби до цих показників та параметри Health та Stress (що відповідають за показники здоров'я та стресу відповідно). Знаючи усі ці дані, отримуються коефіцієнти що впливають на Stress, який кожен цикл симуляції понижує значення Health на показник стресу. По досягненню показника Health нуля, агент помирає.

3.3. Структура бази даних

База даних для додатку, що розроблюється має містити інформацію про симуляційні об'єкти та їх параметри. Згідно того, що системою керування базою даних була обрана SQLite, база даних було поміщена у файл «machina-aquaria-database.db». Оскільки база даних має зберігати інформацію для ініціалізації екземплярів класів, то складена вона була відповідно до цього.

Структурно, базу даних було поділено на декілька пов'язаних таблиць. А саме: на три таблиці Fish, FishTypes та Environments, що представлені на рисунку 3.2.

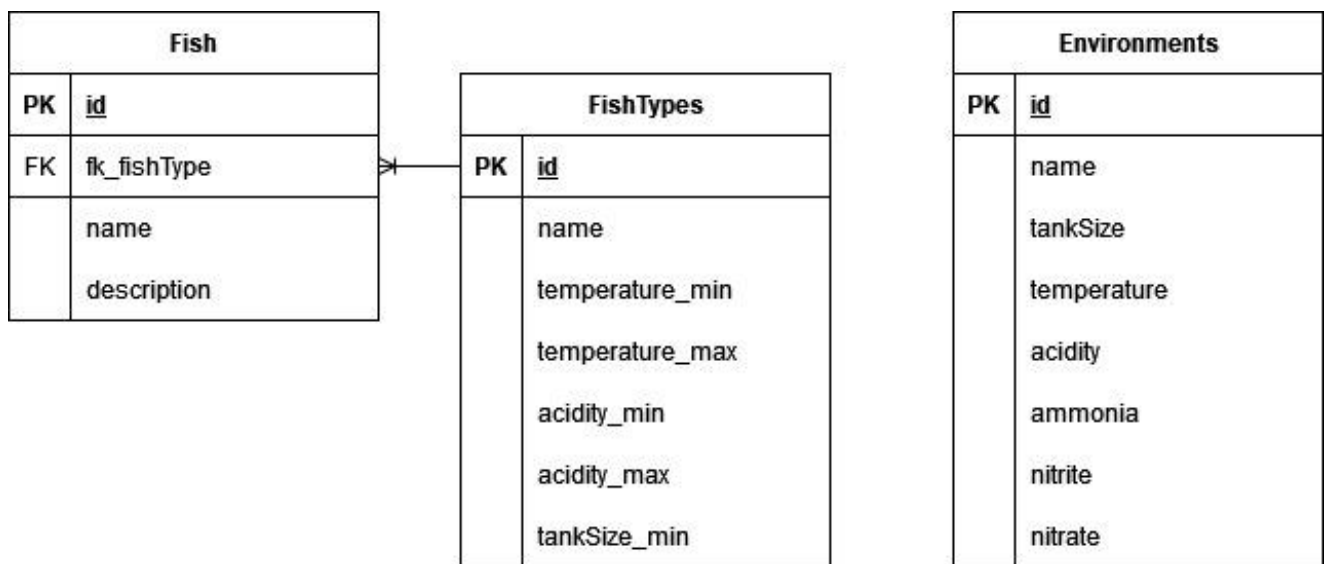


Рисунок 3.2. — Структура бази даних додатку

Таблиця Fish зберігає назву риби та її опис у полях name та description відповідно. Поле fk_fishType містить у собі посилання на тип риби у таблиці FishTypes. Застосований тип даних для обох полей — varchar(255).

FishTypes містить симуляційну інформацію про конкретний вид риби та назву типу. Поля temperature-, acidity- та tankSize_min містять у собі дані типу float. Поле name — текст типу varchar(255).

Таблиця ж `Environments` відповідає за збереження заздалегідь створених середовищ для симуляції, разом з їх параметрами. Усі поля окрім `name` та, звісно, `id` мають тип даних `float`.

3.4. Основні програмні модулі

Розділення програми на дискретні виконавчі модулі є необхідним кроком, що покращує не тільки швидкодійність готового продукту для користувача, але поліпшує процес написання та налаштування додатку для розробника. Згідно цієї методології на кожен клас було виділено один файл з початковим кодом. Згідно принципів ООП кожен клас задовольняє та виконує окремий елемент функціоналу програми.

Контроль базою даних був винесений у статичний клас `DatabaseController`. Логіка користувацького інтерфейсу була винесена у відповідні `UIController` класи, а зв'язка унутрішньої логіки з інтерфейсом реалізована у класі `SaveVariables`. Логіка екологічних агентів із середовищем виділена у класах `Entity` (та його поширеним класом `Fish`) та `Environment` відповідно. Створення візуальних об'єктів риб реалізовано у класі `FishSpawner`.

З точки зору Unity, весь проект був розділений на дві сцени — `Menu` та `Simulation`. Ці сцени містили у собі усі необхідні компоненти для реалізації функціоналу головного меню та вікна симуляції відповідно.

3.5. Інтерфейс користувача

При створення інтерфейсу користувача було вирішено переслідувати методологію мінімалізму та утилітаризма. Через це інтерфейс може не виглядати естетично, але він задовольняє усі вимоги користувача та зручним для користування.

Саме ж створення користувацького інтерфейсу потребувало обрати одну з трьох систем відображення і конструкції його у Unity: UI Toolkit, uGUI та IMGUI. З цих трьох систем було застосовано Unity UI Toolkit, для користувацького інтерфейсу у просторі Menu та uGUI у просторі симуляції.



Рисунок 3.3. — Зовнішній вигляд інтерфейсу головного меню програми

ВИСНОВКИ

За результатом роботи був створений симуляційний додаток, що дозволяє макетувати водну антропогенну екосистему «акваріум». Додаток відповідає мінімальним вимогам до подібних додатків.

Для вирішення задач було досліджено галузь акваріумістики та екосистемного моделювання взагалом. Також було оглянуто вже наявні аналоги задля поліпшення розуміння програмної реалізації та вирішення подібних задач. Саме, було оглянуто проекти «FISHERY» «Biotope» «Fish Farm 3».

Було розглянуто інструментар, що було обрано для реалізації функціоналу додатку та його загальної розробки. Зокрема, було обрано середовище розробки Visual Studio із платформою Unity. Мовою розробки для проекту стала C#, а для керування базою даних було обрано файлова вбудована СКБД SQLite.

Готовий симуляційний додаток виконував мінімальний функціонал подібного типу додатків. Реалізований проект, також, мав дуже примітивне та мінімалістичне візуальне оформлення. Робота над додатком та запискою. Збір інформації та організація її викладу у документі випускної роботи сильно поліпшило практичні навички з пошуку і інтерналізації інформації. Ці навички є дуже важливими для сучасного спеціаліста будь-якої галузі. Окрім того, був розширений кругозір студента-випускника та теоретичні знання у галузі екології, біології та, зокрема, акваріумістики.

Не дивлячись на досить хиткий вид розробленого фінального додатку був отриманий безцінний досвіт у роботі із новими та актуальними інструментами. Були поліпшені та здобуті нові навички роботи у методології ООП мовою C#. Був розглянутий функціонал та опанований пайплайн розробки додатків у Unity. Також, були поліпшені навички роботи із SQL СКБД, на прикладі SQLite.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Lundberg, A., & Srinivasan, M. (2021). Effect of the presence of an aquarium in the waiting area on the stress, anxiety and mood of adult dental patients: A controlled clinical trial. *PloS one*, 16(10), e0258118. <https://doi.org/10.1371/journal.pone.0258118>;
2. Що є екосистема // ел. журнал «Grandars». URL: <https://www.grandars.ru/shkola/geografiya/ekosistema.html>;
3. Стійкість у екосистемах // загальноосвітній ел. журнал «Сезоны года». URL: <https://сезоны-года.рф/%D1%83%D1%81%D1%82%D0%B%D0%B9%D1%87%D0%B8%D0%B2%D0%BE%D1%81%D1%82%D1%8C%20%D1%8D%D0%BA%D0%BE%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC.html>;
4. Антропогенні та натуральні екосистеми // загальноосвітній ел. журнал «Сезоны года». URL: <https://сезоны-года.рф/%D1%8D%D0%BA%D0%BE%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B.html>;
5. Водні екосистеми // загальноосвітній ел. журнал «Сезоны года». URL: <https://сезоны-года.рф/%D0%B2%D0%BE%D0%B4%D0%BD%D1%8B%D0%B5%20%D1%8D%D0%BA%D0%BE%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B.html>;
6. Brunner, Bernd (2003). *The Ocean at Home*;
7. Шереметьев І. І. Акваріумні риби. К.: Рад. шк., 1989—221 с.: іл. ISBN 5-330-00394-6;

8. Опис правил встановлення та загального обладнання акваріуму // акваріумістичний веб-портал «Акваріумні рибки». URL: <https://rybkiexpert.ru/oborudovanie-dlya-akvariuma/>;
9. Бактерії у акваріумному середовищі. URL: <https://balani.com.ua/bakteri%D1%97-korisni-dlya-akvariuma/>;
10. Begon, M.; Harper, J. L.; Townsend, C. R. (1988). Ecology: Individuals, Populations and Communities;
11. Сторінка продукту «FISHERY» // ел. магазин «Steam». URL: <https://store.steampowered.com/app/809730/FISHERY>;
12. Сторінка продукту «Biotope» // ел. Магазин «Steam». URL: <https://store.steampowered.com/app/549040/Biotope>;
13. Сторінка продукту «Fish Farm 3» // ел. Сторінка компанії «BitBros inc», розділ портфоліо. URL: <https://www.bitbros-games.com/portfolio/fishfarm3>;
14. Застосування технологій Unity у фільмі «Аватар: шлях води» // блог компанії «Unity». URL: <https://blog.unity.com/industry/technology-behind-avatar-the-way-of-water>;
15. Застосування технологій Unity автоконцерном «Volvo» // блог компанії «Unity». URL: <https://unity.com/case-study/volvo>;
16. Сторінка IDE «Visual Studio». URL: <https://visualstudio.microsoft.com/>;
17. Розділ опису мови C# // офіційна документація-специфікація мови // сайт компанії «Microsoft». URL: <https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/language-specification/introduction>.