

C++

SFML & AI



Gemini деген не?

Gemini — бұл Google компаниясы жасап шығарған ең заманауи және қуатты жасанды интеллект (AI). Ол жай ғана чат-бот емес, мәтінді, кодты, аудионы, кескінді және видеоны қатар түсініп, өндей алғатын мультимодальды жүйе.



Бұгін не істейміз?

Біз 3 негізгі қадам арқылы ойынды құрастырамыз:

1. Stage 1: Ойын идеясын құрастыру
2. Stage 2: Gemini-ге өз идеямызды дұрыс жеткізу
3. Stage 3: Gemini құрастырған кодпен ойынды тексеру

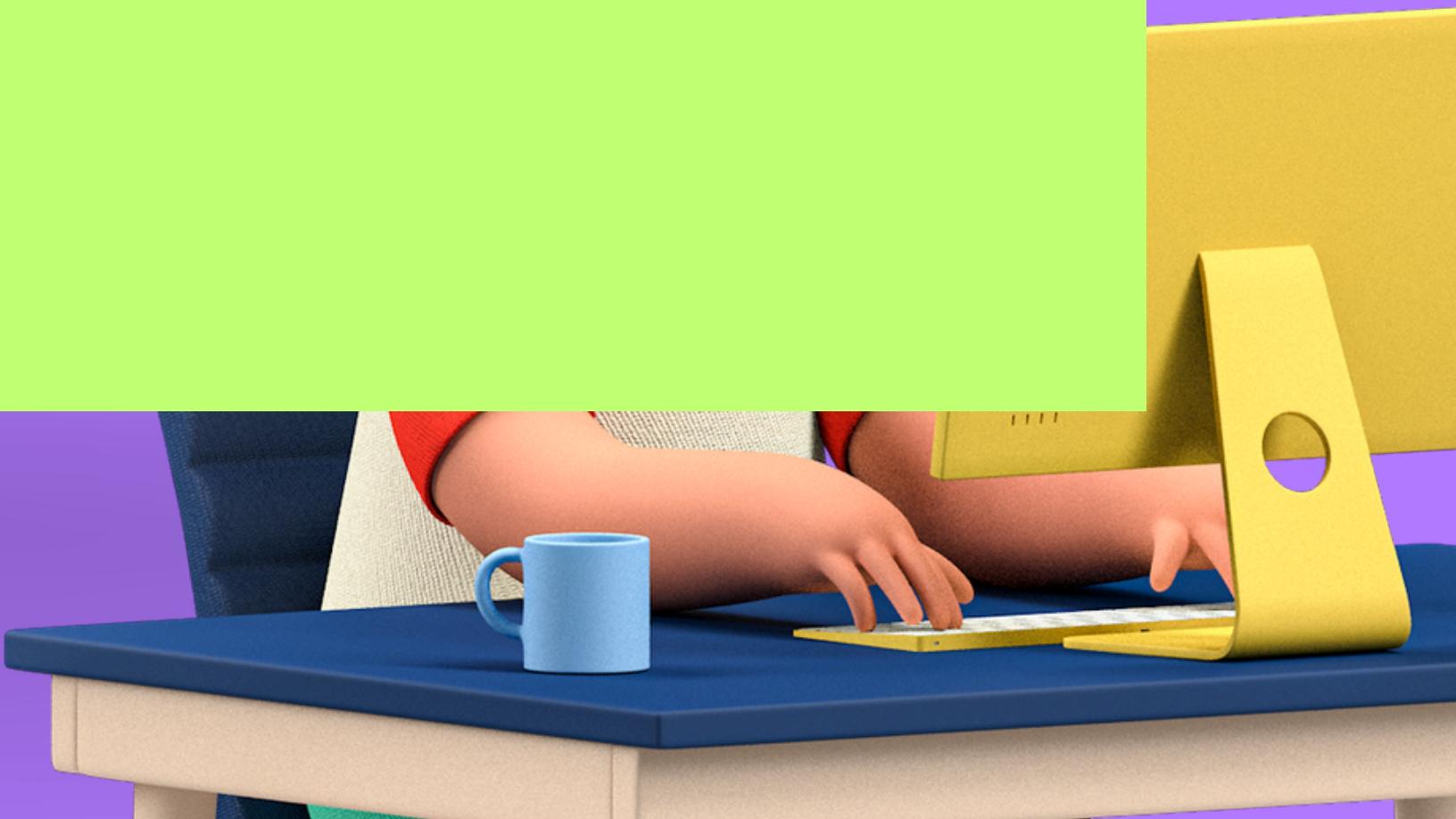


<https://gemini.google.com/app>

"I am a student working on an iMac using SFML 3.0. Please write a C++ main.cpp for this goal:

Goal: [e.g., A player-controlled rocket dodging meteors]

1. Gemini генерациялаған кодты main.cpp файлына қой
2. Build батырмасын бас
3. Run (>) батырмасын бас



Еркекан & Зере

DNA



"Маған iMac (SFML 3.0) үшін ДНҚ молекуласының айналмалы 3D моделін жасайтын C++ кодын жазып бер. Жобада экранда тек нұктес емес, толыққанды ақпараттық мәтін көрінуі тиіс.

Кодқа қойылатын маңызды талаптар:

1. Мәтінді шығару (UI): > - Экранның сол жағында үлкен sf::RectangleShape (фон) жаса.
 - Оның үстіне sf::Text арқылы ДНҚ-ның құрылымы (Аденин, Тимин, Гуанин, Цитозин), қос шиыршық (double helix) туралы биологиялық ақпаратты және математикалық формулаларды ($x = R \cdot \sin(\theta)$) қазақ тілінде жаз.
 - Мәтінді жаңарту үшін міндетті түрде sf::String::fromUtf8(str.begin(), str.end()) синтаксисін қолдан.
2. Графикалық модель:
 - Екі спираль екі түрлі түсті болсын.
 - std::cos функциясын қолданып, спиральдың артқы жағындағы нұктелерді кішірейтіп (scale) және мөлдірлігін (alpha) азайтып, 3D эффектісін жаса.
3. iMac/SFML 3.0 Қатесіздік ережелері:
 - RenderWindow: sf::RenderWindow window(sf::VideoMode({1000, 800}), sf::String::fromUtf8(title.begin(), title.end()));.
 - sf::Vertex: sf::Vertex үшін конструктор қолданба. Объектіні жасап, v.position = sf::Vector2f({x, y}); және v.color = color; деп жеке меншікте.
 - Типтер: sf::Uint8 орнына std::uint8_t қолдан.
 - Event: std::optional<sf::Event> форматындағы event loop қолдан.

Кодты толық түсініктемелермен және барлық мәтіндік ақпарат экранда анық көрінетіндей етіп ұсын."

Мөлдір & Амина

Fourier



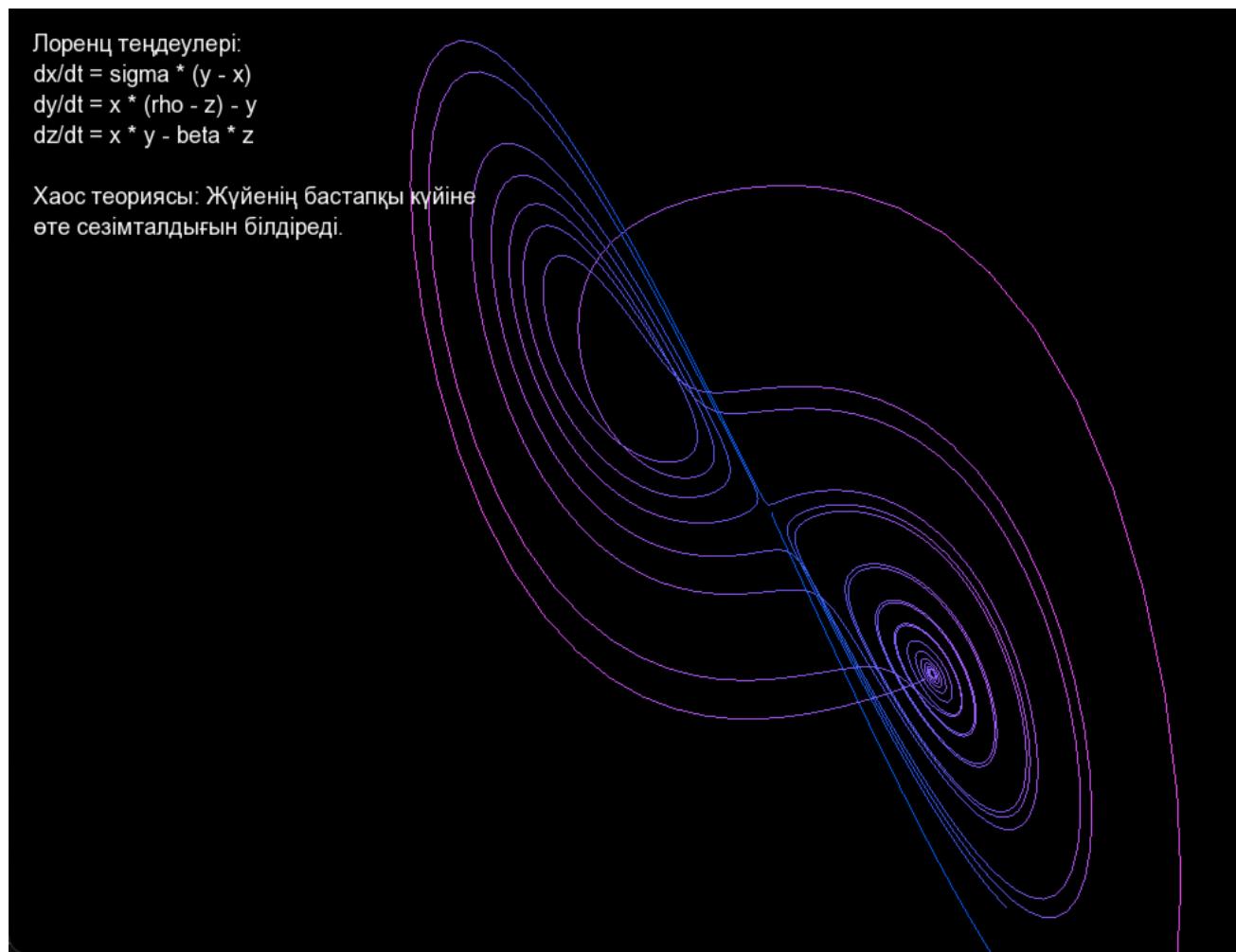
"Маған iMac-те SFML 3.0 кітапханасымен жұмыс істейтін 'Фурье түрлендіруі' жобасына арналған C++ кодын жазып берші. Кодты жазғанда келесі қатаң ережелерді сақта, әйтпесе iMac компиляторы қате (error) береді:

1. `sf::Vertex` инициализациясы: SFML 3.0-де `sf::Vertex` үшін конструктор жоқ. Сондықтан `sf::Vertex v({x, y}, color)` деп жазба. Оның орнына айқын түрде былай жаз: `sf::Vertex v; v.position = {x, y}; v.color = color;` немесе екі қабатты жақша қолдан: `sf::Vertex v{{x, y}, color};`.
2. iMac/SFML 3.0 стандарттары:
 - Барлық `setPosition`, `setSize`, `setOrigin` функцияларында `{x, y}` жақшасын қолдан.
 - Event loop үшін `std::optional<sf::Event>` қолдан.
3. Кириллица (Қазақ тілі): Экрандағы Dashboard-та қазақша мәтін шығу үшін `sf::String::fromUtf8(str.begin(), str.end())` әдісін қолдан.
4. `std::complex`: `std::complex<float>(real, imag)` конструкторын пайдалан (фигуралық жақшасыз).
5. Жоба мазмұны: Эпизиклдер арқылы фигура сзып жатқанда, ең соңғы нүктенің (tip) координаталарын, фазасын және гармоника санын экранда қазақ тілінде көрсет. Кодты толықтай түсініктемелермен қазақ тілінде ұсын."



Ясмин & Айша

Loranz



"Маған iMac (SFML 3.0) үшін Лоренц атTRACTорын (Chaos Theory) визуалдайтын C++ кодын жазып бер. Кодта келесі қатаң ережелерді сақта, әйтпесе компилятор қате (error) береді:

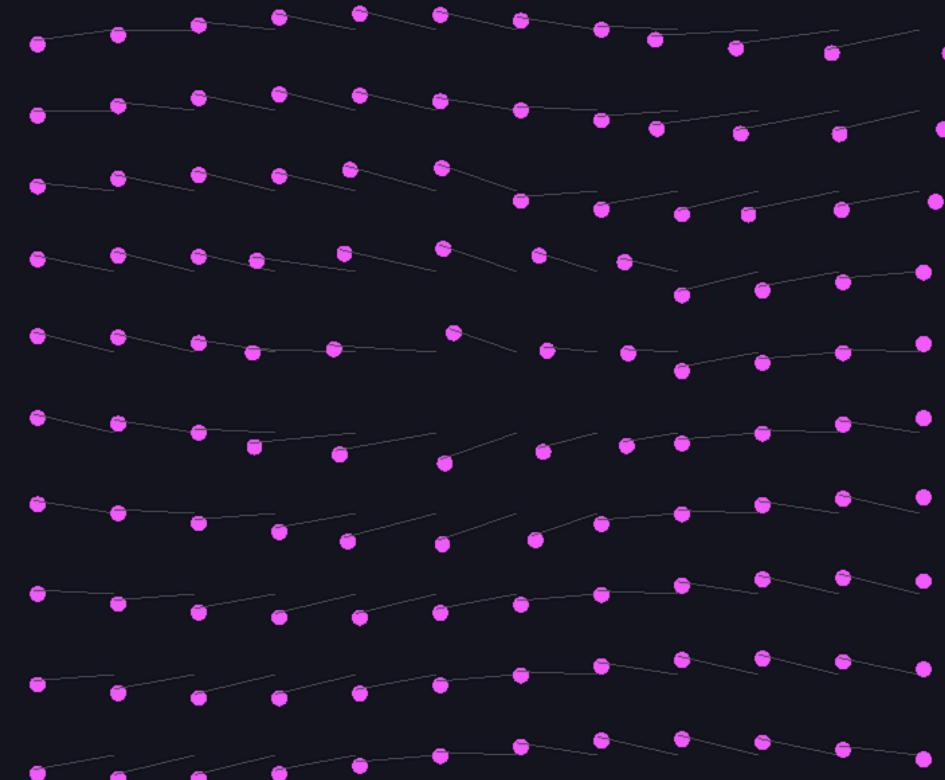
1. String Literal қатесі: Тырнақшадағы мәтінге (мысалы, "Текст") тікелей .begin() қолданба. Оның орнына мәтінді алдымен std::string айнымалысына сақта, сосын барып sf::String::fromUtf8(var.begin(), var.end()) деп қолдан.
2. Мысалы: std::string title = "Lorenz"; window.create(..., sf::String::fromUtf8(title.begin(), title.end()));
3. sf::Text инициализациясы: SFML 3.0-де sf::Text объектісін бос жасауға болмайды. Оны тек қаріп жүктелгеннен кейін былай жаса: sf::Text info(font);.
4. sf::Vertex инициализациясы: Vertex үшін конструктор қолданба. Объектіні жасап алып, мүшелерін жеке меншікте:
5. sf::Vertex v; v.position = sf::Vector2f({x, y}); v.color = color;
6. Vector2f меншіктеуі: Айқын конструктор қолдан: obj.position = sf::Vector2f({x, y});.
7. Лоренц есептеулері: Хаос теориясын көрсету үшін $\sigma=10$, $\rho=28$, $\beta=8/3$ мәндерін және уақыт қадамын ($dt = 0.01$) қолдан. Траекторияны sf::VertexArray арқылы сал.
8. Dashboard: Экранда қазақ тілінде Лоренц тендеулерін және хаос теориясы туралы мәліметті көрсет. Кириллица үшін sf::String::fromUtf8 қолдан.
9. Типтер: sf::Uint8 орнына std::uint8_t қолдан.

Кодты толық түсініктемелермен және барлық техникалық қателер жөнделген күйде ұсын."

Зульфия & Немат

Lattice

Гук заңы: $F = -k \cdot x$ | Деформация деңгейі



"Маған iMac (SFML 3.0) үшін Материалтану (Material Science) саласындағы Кристалдық тордың кернеуін (Lattice Structure Stress Test) визуалдайтын C++ кодын жазып бер. Жоба кристалдық тордың деформациясын және тербелісін симуляциялауы тиіс.

Кодқа қойылатын ҚАТАҢ техникалық талаптар:

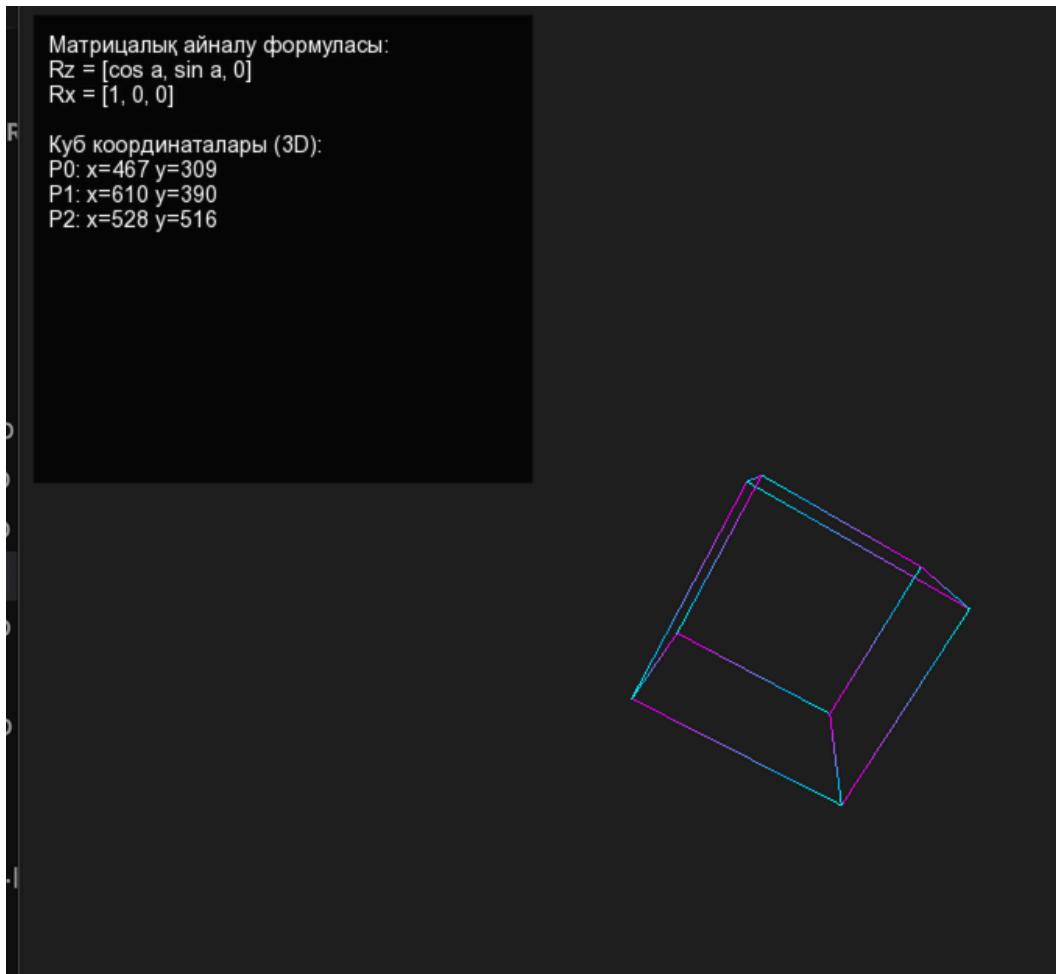
1. `fromUtf8` Қатесін түзету: `sf::String::fromUtf8` функциясына міндепті түрде екі параметр бер: мәтіннің басы мен аяғын.
2. Мысалы: `std::string s = "Мәтін"; text.setString(sf::String::fromUtf8(s.begin(), s.end()));`
3. `RenderWindow`: Терезе атын жазғанда да осы әдісті қолдан:
4. `std::string title = "Lattice Stress Test";`
5. `sf::RenderWindow window(sf::VideoMode({1000, 800}), sf::String::fromUtf8(title.begin(), title.end()));`
6. `sf::Vertex`: Байланыстарды (bonds) салу үшін `sf::Vertex` қолдан. Конструктор қолданбай, мүшелерін жеке меншікте:
7. `sf::Vertex v; v.position = sf::Vector2f({x, y}); v.color = color;`
8. `Vector2f` меншіктеуі: Айқын конструкторды қолдан: `obj.position = sf::Vector2f({x, y});`
9. Типтер мен `Event`: `sf::Uint8` орнына `std::uint8_t` қолдан. `std::optional<sf::Event>` арқылы `event loop` жаса.

Мазмұнға қойылатын талаптар:

- Кристалл моделі: Экранда атомдардың 3D текше құрылымын (Lattice) сал.
- Стресс-тест: Тінтуірді басқанда атомдардың тербелісі (vibration) күшейіп, түсі көктен қызылға ауыссын.
- Инфо-панель: Экранда қазақ тілінде Гук заңы ($F = k \cdot x$) және материалдың деформациясы туралы ақпаратты көрсет.

Кодты толық түсініктемелермен және барлық техникалық қателер жөнделген күйде ұсын."

Едіге & Ихсан



Matrix

"Маған iMac (SFML 3.0) үшін 3D Wireframe Cube (Матрициалық трансформация) жобасын жасап бер.

Кодтағы қателерді болдырмау үшін келесі ережелерді ҚАТАҢ сақта:

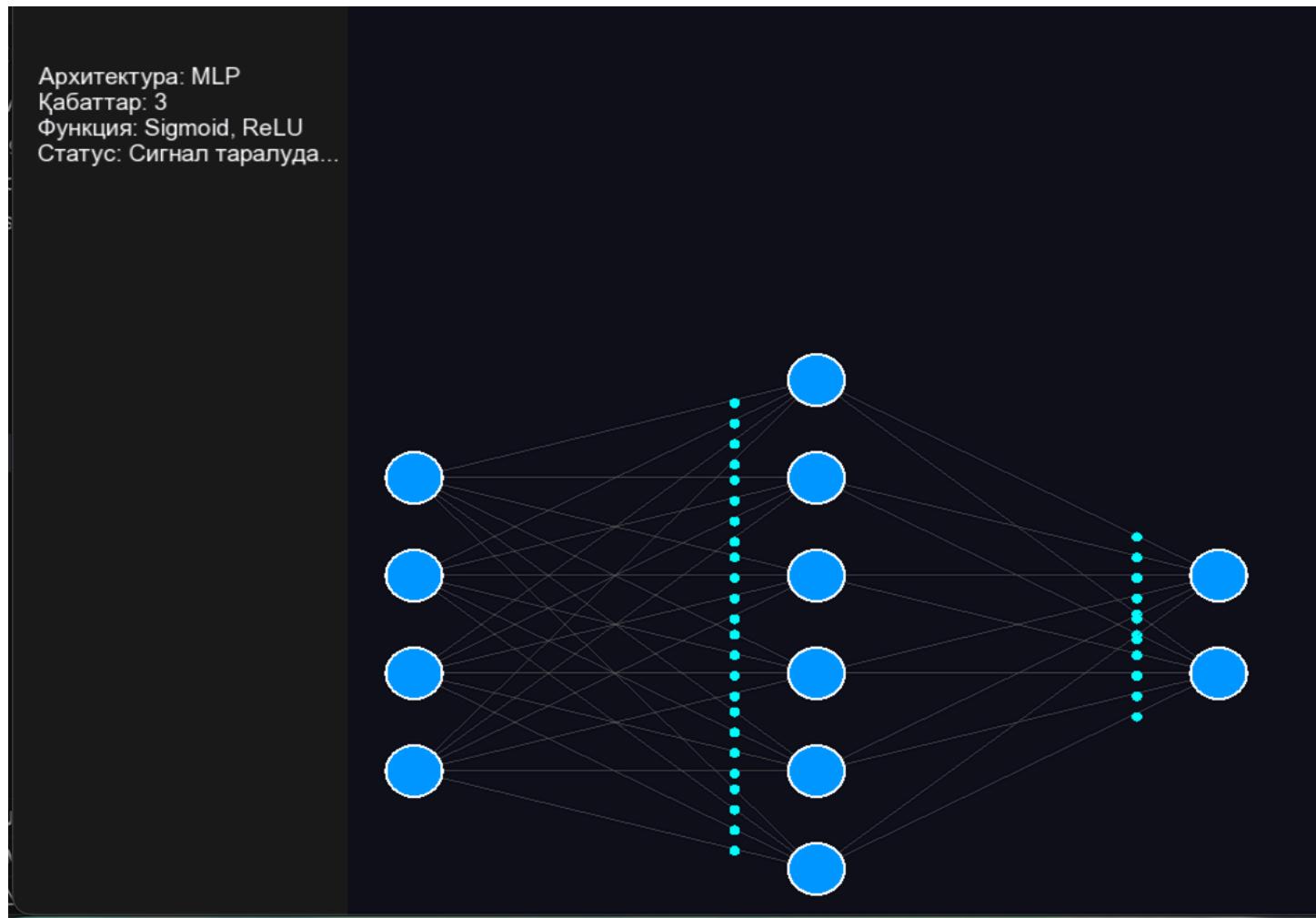
1. String Literal қатесін түзету: Ешқашан тырнақшадағы мәтінге тікелей .begin() немесе .end() қолданба (мысалы, "text".begin() деп жазба). Оның орнына әрқашан std::string айнымалысын қолдан:
2. std::string infoStr = "Матрица мәні...";
3. text.setString(sf::String::fromUtf8(infoStr.begin(), infoStr.end()));
4. RenderWindow: Терезе атын бергенде де осы ережені сақта:
5. std::string title = "3D Matrix Engine";
6. sf::RenderWindow window(sf::VideoMode({1000, 800}), sf::String::fromUtf8(title.begin(), title.end()));
7. sf::Text және Font: sf::Text нысанын жасағанда бірден font параметрін бер: sf::Text uiText(font);. Қаріптің сәтті жүктелгенін if (!font.openFromFile(...)) арқылы тексер.
8. Матрициалық жүйе: 4x4 Матрица құрылымын жаса. Кубтың 8 нүктесін (\$x, y, z\$) Матрица арқылы айналдырып, Projection Matrix арқылы 2D экранға өткіз.
9. SFML 3.0 Graphics: > - sf::Vertex үшін конструктор қолданба, мүшелерін жеке меншікте.
 - sf::Vector2f({x, y}) деп айқын түрде жаз.
 - std::optional<sf::Event> арқылы event loop жаса.
 - sf::Uint8 орнына std::uint8_t қолдан.

UI мазмұны: Экранның сол жағында қара-мөлдір панель үстінде айналу матрицасының формулаларын және 3D нүктелердің координаталарын қазақ тілінде көрсет.

Кодты толық түсініктемелермен және ешқандай компиляция қатесінсіз ұсын.

Даниал & Мейірлан

Neural Network



"Маған iMac (SFML 3.0) үшін Neural Network Visualizer (Нейрондық желіні визуалдаушы) бағдарламасын C++ тілінде жазып бер. Жобаның мақсаты – көпқабатты перцептронның (MLP) құрылымын және сигналдың таралуын көрсету.

Кодтағы техникалық қателерді болдырмау үшін келесі ережелерді ҚАТАҢ сақта:

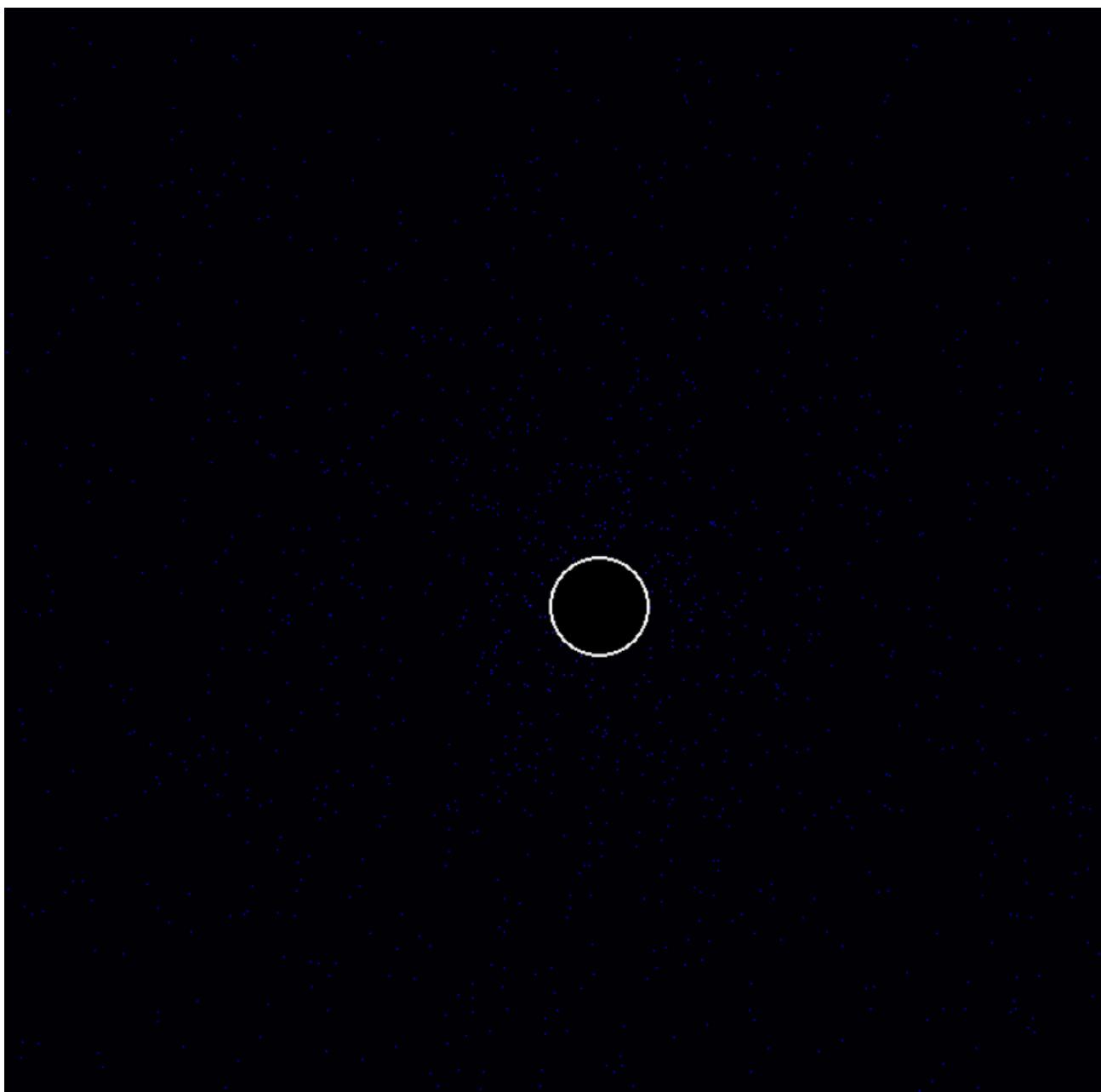
1. String Literal & fromUtf8: Тырнақшадағы мәтінге тікелей .begin() қолданба. Мәтінді алдымен std::string айнымалысына сақта, сосын барып sf::String::fromUtf8(var.begin(), var.end()) арқылы sf::Text мен sf::RenderWindow үшін қолдан.
2. sf::Text & Font: sf::Text нысанын жасағанда бірден қаріпті көрсет: sf::Text text(font);. Қаріпті жүктеуді if (!font.openFromFile("Arial.ttf")) арқылы міндетті түрде тексер.
3. sf::Vertex (Сынапстық байланыстар): Нейрондар арасындағы байланыстарды салу үшін sf::Vertex қолдан. Конструктор қолданбай, мүшелерін жеке меншікте: sf::Vertex line[2];line[0].position = sf::Vector2f({x1, y1}); line[0].color = color1;line[1].position = sf::Vector2f({x2, y2}); line[1].color = color2;
4. Vector2f Assignment: Барлық жерде obj.setPosition(sf::Vector2f({x, y})) немесе obj.position = sf::Vector2f({x, y}) форматын қолдан.
5. SFML 3.0 Стандарттары: std::optional<sf::Event> арқылы event loop жаса. sf::Uint8 орнына std::uint8_t қолдан.

Визуализация талаптары:

- Құрылым: Экранда кем дегенде 3 қабат (Input, Hidden, Output) болсын. Нейрондарды sf::CircleShape арқылы бейнеле.
- Динамика: Сигналдың өтуін (feed-forward) көрсету үшін байланыс сызықтарының бойымен кішкентай жарық нүктелер (сигналдар) қозғалып тұрсын.
- UI Панель: Сол жақта қара-мөлдір фон үстінде нейрондық желінің архитектурасы, белсендіру функциялары (Sigmoid, ReLU) және салмақтар (weights) туралы ақпаратты қазақ тілінде көрсет.

Кодты толық түсініктемелермен және ешқандай компиляция қатесінсіз ұсын.

Сейтжан Ибрагим & Амиржан

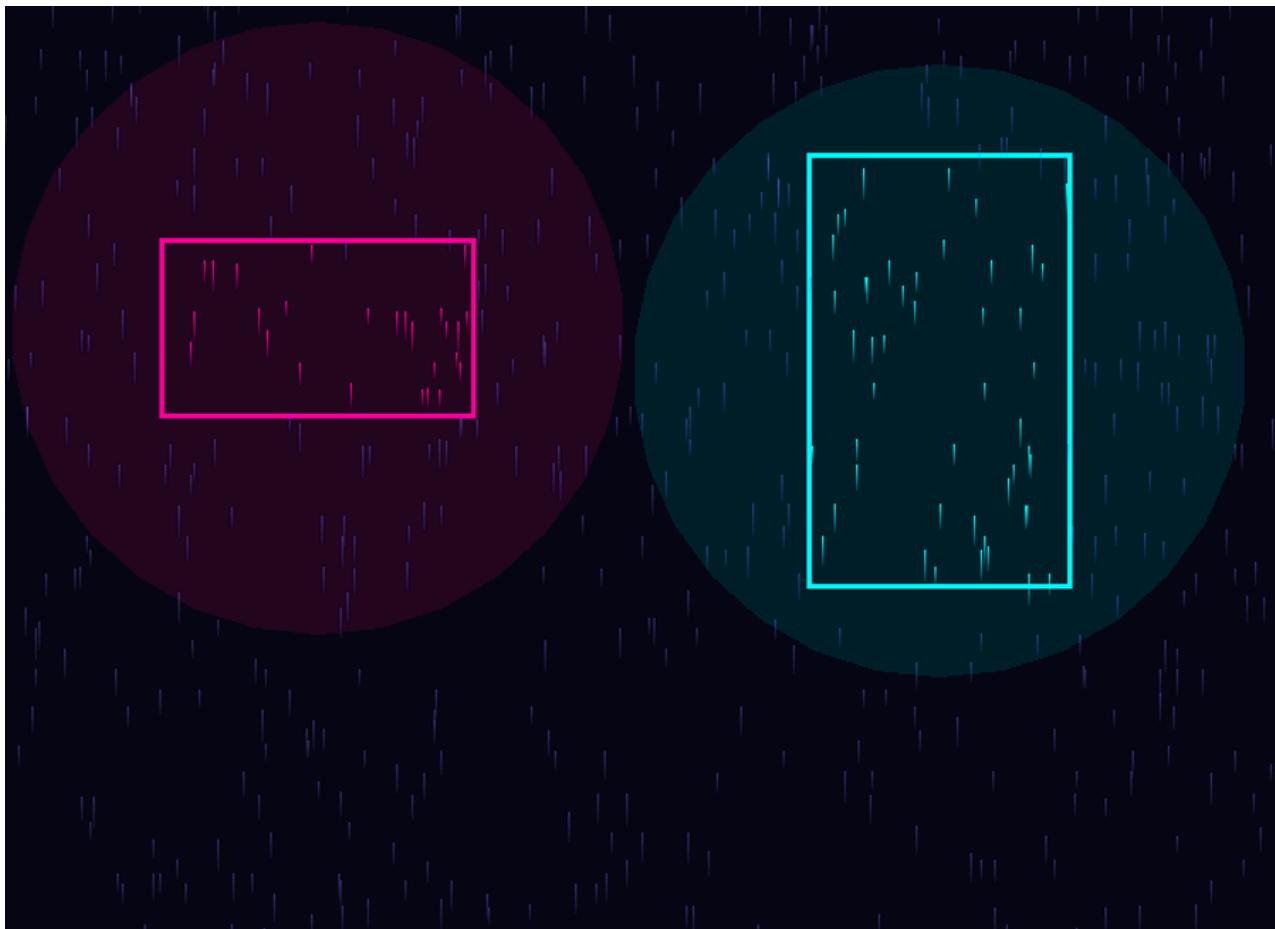


Рабат & Даulet Black hole

"Маған SFML 3.0 кітапханасын қолданып, C++ тілінде 'Black Hole Singularity' (Қара құрдым) атты визуалды жоба жазып бер. Код мына талаптарға сай болуы керек:

1. Терезе: 800x800 өлшеміндегі терезе аш.
2. Жұлдыздар жүйесі: 2000 жұлдыз (stars) жаса. Олар sf::Vertex немесе кішкентай sf::CircleShape арқылы нүкте түрінде көрінуі керек.
3. Математика: Жұлдыздар полярлық координаттар жүйесімен (dist, angle) қозғалуы тиіс. Олар орталыққа қарай тартылып, жақындаған сайын айналу жылдамдығы артуы керек (Faster spin when closer).
4. SFML 3.0 Синтаксисі (Маңызды):
 - Оқиғаларды (Events) std::optional<sf::Event> арқылы тексер.
 - Барлық позициялар мен өлшемдерді фигуралық жақшамен жаз: setPosition({x, y}).
 - Түс мөлдірлігі үшін static_cast<std::uint8_t> қолдан.
5. Визуал: Ортада қара шеңбер (singularity) болсын. Жұлдыздар ортаға жақындаған сайын түсі ақшылданып, жарығы күшеюі керек.
6. Орталық: iMac-тегі GCC/Clang компиляторына ыңғайлы, қатесіз код жаз."

Жұбатқан Ибраһим & Мұхаметали



Rain

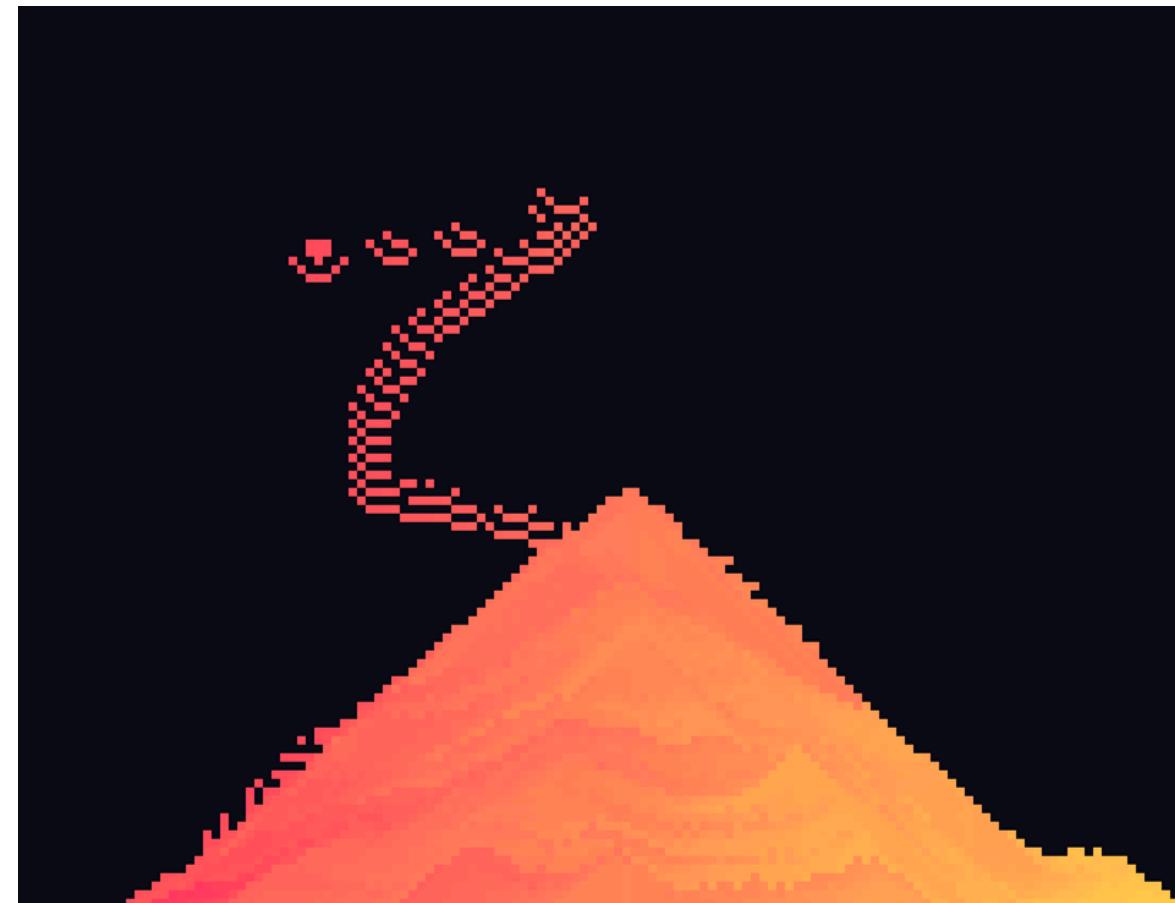
"Маған SFML 3.0 кітапханасын қолданып, C++ тілінде 'Cyberpunk Neon Rain' атты визуалды симуляция жобасын жазып бер.

Код мына талаптарға сай болуы керек:

1. Жаңбыр жүйесі: Терезеде 600-ге жуық жаңбыр тамшылары (drops) жоғарыдан төмен қарай әртүрлі жылдамдықпен жаууы керек. Жаңбыр тамшыларын `sf::Vertex` сыйықтары (Lines) арқылы жаса.
2. Неон эффектісі: Экранда екі неонды белгі (Neon Signs/Boxes) болсын (біреуі қызығылт, біреуі көгілдір). Жаңбыр тамшылары осы белгілердің қасынан өткенде, олардың түсі сол неонның түсіне өзгеруі тиіс (Dynamic Color Masking).
3. Жерге тию: Тамшылар экранның төменгі жағына (жерге) жеткенде, кішкентай шеңберлер түріндегі "шашырау" (Splashes/Ripples) эффектісі пайда болып, жоғалып отыруы керек.
4. SFML 3.0 Синтаксисі (Міндетті):
 - Оқиғаларды (Events) `std::optional<sf::Event>` арқылы тексер.
 - Барлық позициялар, өлшемдер мен векторлар үшін міндетті түрде фигуралық жақшаларды қолдан: `setPosition({x, y})`, `setSize({w, h})`, `sf::VideoMode({800, 600})`. Бұл іMac-тегі компилятор қателерін болдырмау үшін маңызды.
 - Түс мөлдірлігі үшін `static_cast<std::uint8_t>` қолдан.
5. Құрылым: Кодты түсінікті болуы үшін `struct Drop`, `struct Splash` сияқты құрылымдарға бөліп жаз."

Бек & Амаль

Sand



"Маған SFML 3.0 кітапханасын қолданып, C++ тілінде 'Falling Sand Physics Sandbox' атты жоба жазып бер.

Код мына функционалдық талаптарға сай болуы керек:

1. Тор (Grid) жүйесі: Экранды кішкентай ұяшықтарға (мысалы, 160x120) бөл. Ол үшін `std::vector<int>` қолдан. 0 - бос орын, 1 - құм.
2. Құм физикасы: Әр кадр сайын құм түйіршіктері төмен түсүі керек. Егер асты бос болмаса, диагональ бойынша (төмен-солға немесе төмен-оңға) сырғып түсу логикасын жаса.
3. Интерактив: Тінтуірдің сол жақ батырмасын (Mouse Left Click) басқанда, тінтуір түрған жерге құм себілуі (Spawn) керек.
4. SFML 3.0 Синтаксисі (Міндетті):
 - Оқиғаларды (Events) `std::optional<sf::Event>` арқылы тексер.
 - Барлық позициялар мен өлшемдерді міндетті түрде фигуralық жақшамен жаз: `setPosition({x, y})`, `setSize({w, h})`. Бұл iMac-тегі компилятор қателерін болдырмау үшін өте маңызды.
 - Түстер мен координаталар үшін `static_cast<std::uint8_t>` және `float` түрлерін дұрыс қолдан.
5. Визуал: Құм түйіршіктерінің түсі олардың позициясына қарай өзгеріп тұратын "Rainbow" немесе "Neon" әффектісін қос.