

Projekt-Programowanie w języku C++

temat projektu:

Wyspa wilków

Michał Domański

Instrukcja obsługi

1) Główne sterowanie:

- a) „Dodaj królika”- wciśnięcie przycisku 1 umożliwia dodanie królika za pomocą prawego przycisku myszy na dane pole
- b) „Dodaj wilka”- wciśnięcie przycisku 2 umożliwia dodanie wilka za pomocą prawego przycisku myszy na dane pole
- c) „Dodaj wilczyce”- wciśnięcie przycisku 3 umożliwia dodanie wilczyce za pomocą prawego przycisku myszy na dane pole
- d) „Dodaj krzak”- wciśnięcie przycisku 4 umożliwia dodanie krzaka za pomocą prawego przycisku myszy na dane pole
- e) „Rozpocznij symulację/Zatrzymaj symulację”- wciśnięcie przycisku “space” rozpoczyna symulację albo ją zatrzymuje.
- f) „Rozpocznij ture symulacji”- wciśnięcie przycisku “v” rozpoczyna pojedynczą ture symulacji.

2) dokładny opis menu:

g) „Dodaj królika”

wciśnięcie przycisku 1 , włącza ramkę pokazującą że królik został wybrany w lewym dolnym rogu (pierwszy od lewej obrazek), umożliwia dodanie królika za pomocą prawego przycisku myszy na dane pole.

królik działa następująco:

po upływie określonego czasu przenoszą się z równym prawdopodobieństwem na jedno z sąsiednich pól (ośmiu, chyba że znajdują się na wybrzeżu). Przez 1/9 czasu siedzą więc po prostu spokojnie. Każdy królik ma 20% szans przekształcenia się w dwa króliki. Może przechodzić przez krzaki.

Ich liczebność pokazuje się po prawej dolnej stronie aplikacji przy głowie królika.

- h) „Dodaj wilka”- wciśnięcie przycisku 2, włącza ramkę pokazującą że wilk został wybrany w lewym dolnym rogu (drugi od lewej obrazek) umożliwia dodanie wilka za pomocą prawego przycisku myszy na dane pole

wilk działa następująco:

Każdy wilk zachowuje się tak, jak wilczyca, chyba że w pobliżu nie ma królika, natomiast jest wilczyca; wówczas wilk rzuca się w pogoń. Wilki pozbawione tłuszczu giną. Jeżeli wilk i wilczyca znajdują się na tym samym polu i nie będzie tam królika do zjedzenia, to produkują potomka o losowej płci. Nie może przechodzić przez krzaki

Ich liczebność pokazuje się po prawej dolnej stronie aplikacji przy głowie wilka.

- i) „Dodaj wilczyce”- wciśnięcie przycisku 3, włącza ramkę pokazującą że wilczyca została wybrana w lewym dolnym rogu (trzecia od lewej obrazek) umożliwia dodanie wilczyce za pomocą prawego przycisku myszy na dane pole

wilczyca działa następująco:

Każda wilczyca również porusza się losowo, chyba że na jednym z ośmiu sąsiednich pól znajduje się królik. Wówczas wilczyca rozpoczyna pościg. Jeżeli znajdzie się na tym samym polu co królik, to zjada go i zyskuje jedną porcję tłuszczu; jeżeli nie, to traci 0,1 porcji tłuszczu. Wilki pozbawione tłuszczu giną. Jeżeli wilk i wilczyca znajdują się na tym samym polu i nie będzie tam królika do zjedzenia, to produkują potomka o losowej płci. Nie może przechodzić przez krzaki

Ich liczebność pokazuje się po prawej dolnej stronie aplikacji przy głowie wilczycy.

- j) „Dodaj krzak”- wciśnięcie przycisku 4, włącza ramkę pokazującą że krzak została wybrana w lewym dolnym rogu (czwarty od lewej obrazek) umożliwia dodanie krzaka za pomocą prawego przycisku myszy na dane pole

krzak to obszar niedostępny dla wilków.

- k) „Rozpocznij symulację/Zatrzymaj symulację”- wciśnięcie przycisku “space” rozpoczyna symulację albo ją zatrzymuje.

gdy jest symulacja zatrzymana to na środku pokazuje się play(obrazek strzałki w prawo) a gdy symulacja jest uruchomiona to na środku pokazuje się stop(obrazek dwóch pojedynczych kresek)

- l) „Rozpocznij ture symulacji”- wciśnięcie przycisku “v” rozpoczyna pojedynczą ture symulacji.

Dokumentacja

Klasy użyte w programie:

- 1)Animal
- 2)Texturki
- 3)mapa
- 4)krolik
- 5)wilk

1) Animal-

Klasa zawierające podstawowe rzeczy do stworzenia zwierzęcia takie jak losowy ruch czy grafikę.

Pola publiczne:

- **int x,int y** -służą do określenia aktualnej pozycji
- **int next_x,int next_y** -planowany ruch
- **sf::Sprite sprite**-sprite by móc przedstawić graficznie obiekt
- **sf::Texture textura**-tekstura by móc przedstawić graficznie obiekt
- **Animal(int _maxKlatek)**-konstruktor określający maxKlatek by móc wykonać animacje

- **Animal()**-zwykły konstruktor
- **int klatka**-aktualna klatka animacji
- **void aktAnim(float deltaTime)**-aktualizacja obrazu na następną klatkę z wykorzystaniem czasu klatek na sekundę
- **virtual ~Animal()**-destruktor
- **int getFlipX()**-informacja w którą stronę jest obraz obrócony

Pola chronione:

- **int maxKlatek** -maksymalna ilość klatek ile jest w obrazie
- **float sekunda** -aktualny czas animacji
- **int FlipX**-zmienna określająca obrócenie obrazu

2) Texturki

Klasa zawierające wszystkie tekstury obrazów , czcionki oraz odpowiedzialna za wczytanie tych elementów.

Pola publiczne:

- **sf::Texture wilk** -tekstura wilka
- **sf::Texture wilczyca**-tekstura wilczycy
- **sf::Texture krolik**-tekstura krolika
- **sf::Texture mapa**-tekstura mapy
- **sf::Texture poleKursora**-tekstura pola Kursora
- **sf::Texture krzaki** -tekstura krzaków
- **sf::Texture bunnyHead**-tekstura głowy krolika
- **sf::Texture wolfManHead**-tekstura głowy wilka
- **sf::Texture wolfWomanHead**-tekstura głowy wilczycy
- **sf::Font adventureFont**-czcionka adventureFont

- **sf::Texture pause**-tekstura pauzy
- **sf::Texture play**-tekstura playa
- **sf::Texture ramka**-tekstura ramki
- **Texturki()**-konstruktor
- **Texturki(std::string sciezka)**-konstruktora z sciezka do grafik oraz czcionki
- **virtual ~Texturki()**-destruktor

3) mapa-

Klasa zawierająca pola mapy,opcje stawiania zwierząt albo krzaków na pola,pola aktualnych zwierząt oraz ilość wszystkich zwierząt

Pola publiczne:

- **char mapaTekstowa[25][25]**-mapa charów gdzie dany symbol symbolizuje co jest na danym polu
- **char mapaTekstowaNextTurn[25][25]**-mapa następnej tury charów gdzie dany symbol symbolizuje co jest na danym polu.
- **mapa(Texturki * textura_mapy,int _maxIloscWilkow,int _maxIloscKrolikow)**-konstruktor mapy,gdzie są podane wszystkie textury(przez klasę texturki) oraz ustalenie limitów liczby zwierząt
- **virtual ~mapa()**-destruktor mapy,czyści też wskaźniki ****rabits** oraz **** wolfs**
- **void drawMap(sf::RenderWindow *window,float odl,bool akt,float fps)**-aktualizacja grafiki w okienku(pierwszy argument to okno,drugi to dystans pól między sobą ,fps to czas klatki na sekunde w danym momencie)

- **void setRabit(int x,int y,Texturki *textura_mapy)**-funkcja do postawienia królika na danym polu
- **void setWolf(int x,int y,Texturki *textura_mapy,double _fat,bool _facet=false)**-funkcja do postawienia wilka albo wilczycy (zależne od _facet) na danym polu
- **int totalnalloscKrolkow**
- **krolik **rabits**-wskaznik do wskazników krolików
- **int totalnalloscWilkow**
- **wilk **wolfs**-wskaznik do wskazników wilków
- **void changeSelect(int newSelect)**-zmienia dany wybór na nowy oraz aktualizuje położenie ramki symbolizującą dany wybór
- **void setObject()**-stawia obiekt na danePole

Pola prywatne:

- **Texturki *texturki**-wskaźnik zawierająca wszystkie tekstury
- **sf::Sprite sprite**-sprite do ziemi
- **sf::Sprite spriteKrolik**
- **sf::Sprite spriteWilk**
- **sf::Sprite spriteWilczyca**
- **sf::Sprite spritePoleKursora**
- **sf::Sprite spriteKrzaki**
- **sf::Sprite spriteBunnyHead**
- **sf::Sprite spriteWolfManHead**
- **sf::Sprite spriteWolfWomanHead**
- **sf::Sprite spritePlay**
- **sf::Sprite spritePause**
- **sf::Sprite spriteRamka**

- **sf::RectangleShape legenda**-prostokąt do legendy
- **sf::Text textLegendy**-tekst , dokładnie służący do pokazania cyfr w ilości zwierząt
- **int maxIloscWilkow**
- **int maxIloscKrolkow**
- **int iloscWilczyc**
- **int select**-aktualny wybór zwierząt które postawić (albo krzaka)
- **int mouseX,int mouseY**-współrzędne myszki w okienku

4) krolik-

Klasa która dziedziczy od Animal i opisuje krolika

Pola publiczne:

- **krolik()**-konstruktor
- **krolik(int _x,int _y,char _symbol_char _mapa[][25],Texturki * texturki,int maxKlatek)**-konstruktor z współrzędnymi,symbolem char na mapie(która jest tablicą charów) oraz z maksymalną ilością klatek animacji
- **krolik(int _x,int _y,char _symbol_char _mapa[][25],sf::Texture texturaKrolika,int maxKlatek)**-ten sam konstruktor co powyzej tylko zamiast wskaźnikiem do texturki mamy podaną texture królika
- **void ruch(char _mapa[][25],krolik **rabits,int *totalnaIloscKrolikow,char nextTurnMap[][25],int maxIloscKrolikow)**-wykonanie czynności

królika(stanie w miejscu albo losowy ruch oraz
możliwość rozmnożenia)

- **virtual ~krolik()**-destruktor

Pola chronione:

- **bool checkWolnePole(char _mapa[][25],char nextTurnMap[][25])**-sprawdza wolne pole dookoła

Pola prywatne:

- **int tura**-zmienna która odlicza czas na bezruch(czyli co 9 turę)
- **void rozmnoz(char _mapa[][25],krolik **rabits,int *totalnalloscKrolikow,char nextTurnMap[][25],int maxlloscKrolikow)**-funkcja do rozmnazania krolików

5) wilk-

Klasa która dziedziczy od Animal i opisuje wilka (albo wilczyce)

Pola publiczne:

- **double fat**-poziom tłuszczu wilka,gdy spadnie do 0 albo poniżej wilk ginie.Może zwiększyć poziom tłuszczu przez jedzenie krolikow
- **bool facet**-opisuje czy to samiec czy samica
- **Texturki *texturka**-wskaźnik do texturki
- **wilk(int _x,int _y,char _mapa[][25],Texturki *texturki,double _fat,bool _facet,int _maxKlatek)**-konstruktor
- **void ruch(char _mapa[][25],wilk **wolfW,krolik **rabits,int**

***totalnalloscKrolikow,int ***

totalnalloscWilkow,char

nextTurnMap[][25])-wykonanie czynności wilka(losowy ruch lub pogoń za wilczycą albo krolikiem)

- **virtual ~wilk()**

Pola chronione:

- **void zjadanieKrolika(krolik **rabits,int *totalnalloscKrolikow,char nextTurnMap[][25])**-zjadanie krolika (albo krolików) na danym polu oraz zwiększenie fat
- **void umieranie(wilk **wolfW,int *totalnalloscWilkow)**-usuniecie wilka

Pola chronione:

- **void rozmnoz(char _mapa[][25],wilk **wolfW,int *totalnalloscWilkow)**-rozmnozenia wilkow(gdzie losowana jest płeć)