СУ “Св. Климент Охридски”,
ФМИ – Софтуерно инженерство
Курсов проект по Обектно-ориентирано програмиране

God

Иван Георгиев Механджиев фак.№61798

Въведение

Програмата симулира *Бог, който може да контролира планети и съществата, които ги населяват. Съществата са 4 вида: Entity, Animal, Human, God. Богът има различни способности, сред които да унищожи дадена планета или да насели планета със същества. Основните действия, които могат да извършват съществата са да атакуват и да се движат по планетата.*

Описание на програмния код

Клас Point2D:

Полета:

Double x , y- Те определят положението на съществата по планетите.

Методи:

Void: setX(int), setY()- set-ери, задаващи стойността на съответни X и У в зависимост от подаден double

double: getX(Point2D) ,(Point2D) – get-ери, връщащи стойността на съответно X и У на подадената инстанция

double getDistance(Point2D& A, Point2D& B)– set-ер, връщащ разстоянието между подадените инстанции

Клас Entity:

Полета:

string name, double energy, double size, double weight, double strength- Те определят съответно името, енергията, размера, теглото и силата на всеки обект Entity

Методи:

void setName(string), void setEnergy(double), void setSize(double), void setWeight(double), void setStrength(double), void setX(int), void setY(int) –set-ери, задаващи стойностите на съответните полета

string getName(), double getEnergy(), double getSize(), double getWeight(), double getStrength, int getX(), int getY() –get-ери, връщащи стойностите на съответните полета

Point2D Position-get-ер, връщащ на координати чрез обект от класа Point2D

void setCurrentState(State currentState)-set-ер, задаващ състоянието на обект Entity

void Attack()- проверява се дали наблизо има същество, което може да бъде атакувано, ако има се атакува

void Move()-придвижване на случаен принцип, ако съвпадне с друго същество се сливат (по условие)

virtual void DoAction()-избира се произволно действие

Конструктор:

Entity ()-конструктор по подразбиране, задаващ стойности за името, енергията, размера, теглото, силата, позицията и състоянието на обект Entity.

Клас Animal(наследява Entity):

Методи:

void Eat(), void Sleep(), void SearchingForFood(), void DoAction(State action)- set-ери, взависимост от действието на обекта, повишават или намалят неговата енергия и тегло.

virtual void DoAction(State)-избира се произволно действие

Конструктори:

Animal ()-конструктор по подразбиране

Animal(std::string name)-конструктор, задаващ име на обекта

Клас Human(наследява Animal):

Методи:

virtual void DoAction()-избира се произволно действие

void Analyse();-позволява на обекта да се регенерира

Конструктори :

Human ()-конструктор по подразбиране

Human(string)-конструктор, задаващ име на обекта

Клас God(наследява Human):

Полета:

Scene* sc(pointer to Scene)– подава се в конструктора

Методи:

void CreatePlanet()-създава планета със случайно име

void GetStatistics()-изкарва на екрана колко същества има на всяка планета

void AddEntity(string, int, EntityType) създава вид Entity по подаден стринг, който се използва от метод , създаващ n на брой същества от типа същество

void ErasePopulation(string)-извиква clear() на вектора със същества от планетата с име, съвпадащо с това на подадения стринг

void DestroyPlanet(stirng)-извиква .erase() на вектора със планети за планетата с име, съвпадащо с това на подадения стринг

int Check()-проверява дали има създадени планети

Конструктори и деструктори:

God ()-конструктор по подразбиране

God(string)-задава се име на обекта

~God()-деструктор по подразбиране, извикващ delete за указателите

Клас Planet:

Полета:

string name – запазва името на планетата

vector<Entity*> population-вектор, запазващ всички същества на дадената планета

Методи:

std::string getName()const-константно зададен get-ер, връщащ името на Планетата.

int getPopulation()const-константно зададен get-ер, връщащ размера(популацията) на Планетата.

void erasePopulation();-метод, изчистващ векторът population чрез clear() и освобождаващ място от паметта чрез delete

void movePopulation()-метод, извикващ функцията Move() за всяка инстанция от вектора

void findPosition()-функция, която в зависимост от позицията на съответния обект атакува произволен друг или премества позицията на дадения

void addPopulation(EntityType type, int number)-метод, добавящ нови инстанции на вектора, в зависимост от вида и броя им.

Конструктори и деструктори:

Planet(string)-извиква setName(string) за подадения стринг

~Planet()-деструктор по подразбиране

Клас RandomGenerator:

Методи:

Point2D getRanodmCordinates()-генерира произволни координатни X и У, вариращи от 1 до 100, с помощта на srand(time(0))

string getRandomName()-генерира произволно име, създадено от подходящо избрани символи, с помощта на srand(time(0))

void getRandomDamage(Entity& sth)-генерира произволна големина на щетата, нанесена върху обекта Entity

Клас CommandManager:

Полета:

string nameofPlanet,int number,EntityType type-Полета, съдържащи името на планетата, от която е, броя единици и типа на обекта;

Методи:

void readCommand()-главен метод на класа, записващ нововъведените данни от потребителя(Име на планетата, брой, тип) за обекта. Този метод създава „популация“ на съответната планета

int getNumber()const-константен get-ер, връщащ подаденото число от потребителя.

string getName()const-константен get-ер, връщащ подаденото име на Планетата от потребителя.

EntityType getType()const-константен get-ер, връщащ подадения тип от потребителя.

Клас Scene:

Полета:

vector<Planet* > planet- тук се запазват създадените планети

string planetName-име на планетата

EntityType type-вид на Entity(entity, animal, human, god)

int number-брой, подаден от потребителя.

Методи:

void setPlanetName(string)-Задава име на планетата

void setEntity(EntityType)-Задава тип на Entity

void setNumber(int)-задава цяло число

int getPlanetSize()const-константен get-ер, връщащ размера на вектора с планети

void createPlanets()-Създава планета, използвайки RandomGenerator, с който задава името ѝ, и го вмъква във вектора planet

void erasePopulation()-Проверява дали съществуват планети и ако да, изтрива популацията им с помощта на функцията erasePopulation() от клас Planet

void destroyPlanet()-Проверява дали съществуват планети и ако да, ги унищожава с помощта на erase() –векторна функция

void addEntity()-създава Entities на съответващата планета по тип и брой.

void getStatistics()-get-ер, връщащ информация за Планетите относно техния пореден номер, име и популация

void sceneUpdate()-функция, която се извършва на заден план. Тя непрекъснато променя позициите на обектите от вектора planet.

int check()-проверява съществуват ли създадени планети

Конструктори и деструктори:

Scene ()-конструктор по подразбиране

~Scene()-деструктор по подразбиране

Клас Simulator:

Полета:

God* m_player- адрес към БОГ

Command Manager* command-адрес към командния мениджър

Scene* sc-адрес към действията(Scene)

Методи:

void Menu()-извиква валидните команди

void Update()-на отделна нишка извиква Update1 за всяка една от планетите

void Run()-извиква Menu() и RecognizeCommand метода на CM

Конструктори и деструктори:

Simulator ()-конструктор по подразбиране, инициализиращ указателите

Simulator(Simulator&)-копи конструктор, присвояващ стойностите на полетата към нов обект на същия клас.

~Simulator()-деструктор по подразбиране, извикващ delete за указателите