



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

# God

Благой Симеонов Анастасов, Факултетен № 61799

## Съдържание

1. Въведение .....	2
2. Описание на приложените алгоритми .....	2
3. Описание на програмния код.....	2
4. Използвани технологии .....	2

1. Въведение
2. Описание на приложените алгоритми
3. Описание на програмния код
4. Използвани технологии

### Въведение:

Програмата симулира Бог, който може да създава и унищожава планети и същества, които ги обитават, както и във всеки един момент да получи информация на коя планета колко и какви същества има. Самите същества биват няколко вида: Entity, Animal, Human, God. Те могат да извършват няколко действия като основните са движение по планетата и атака на противник.

### Описание на програмния код:

Клас Point2D:

Полета:

double x, double y – определят положението на съществата

Методи:

double GetX() – връща стойността на x на обекта

double GetY() – връща стойността на y на обекта

void SetX(double x) – задава стойността на x  
void SetY(double y) – задава стойност на y  
double distance(Point2D& p) – връща разстоянието до дадения обект

Конструктори и деструктори:

Point2D() – по подразбиране

Point2D(double x, double y) – задаване стойности на x и y

~Point2D() – деструктор по подразбиране

## Клас Entity:

Полета:

string name, double energy, weight, strength, size. Те определят името, енергията, теглото, силата и размера на съществото.

State state – състояние от enum State

Point2D position – чрез класа Point2D определяме позиция.

RandomGenerator randomPosition – randomPosition от класа RandomGenerator

Методи:

void SetName(string name); void SetEnergy(double energy); void SetSize(double size); void SetWeight(double weight); void SetStrength(double strength); void SetState(State state); void SetPosition(Point2D position); - задават име, енергия, размер, тегло, сила, състояние(което е в enum) и позиция.

string GetName(); double GetEnergy(); double GetSize(); double GetWeight(); double GetStrength(); State GetState(); Point2D GetPosition(); - връщат име, енергия, размер, тегло, сила, състояние(което е в enum) и позиция.

virtual void Attack(Entity& enemy); virtual void Move(); virtual void Eat(){} virtual void Sleep(){} virtual void SearchForFood(){} virtual void Analyze(){} virtual void DoAction(); - виртуални функции, които имплементират какво правят и какво им коства на съществата да го правят, когато те са в състояние, носещо името на съответната функция.

Конструктори и деструктори:

Entity() – default

Entity(string name, double energy, double size, double weight, double strength, State state, Point2D position) - задават се съответните стойности

~Entity() – default

### Клас Animal (наследява Entity):

Методи:

void Eat(), void Sleep(), void SearchForFood() – извършвайки едно от тези неща, Animal променя своята енергия/сила/тегло/размер.

void DoAction() – избира се random действие на Animal

Конструктори и деструктори:

Animal() – default

Animal(string name, double energy, double size, double weight, double strength, State state, Point2D position) – задава съответните стойности

~Animal() – default

### Клас Human(наследява Animal):

Методи:

void Analyze() – Human може да извършва това действие за разлика от Animal

void DoAction() – действие на случаен принцип

Конструктори и деструктори:

Human() – default

Human(string name, double energy, double size, double weight, double strength, State state, Point2D position) – задава съответните стойности

~Human() – default

### Клас God(наследява Human):

Методи:

void AddPlanets(Scene\*); void DeletePopulation(Scene\*, string); void ErasePlanet(Scene\*, string); void AddEntities(Scene\*, EntityType, string,int); void FetchStatistics(Scene\*); - създава се указател към Scene извикала тези функции, които са съответно за добавяне

на random планети, изтриване на население, унищожаване на планета, добавяне на население и вадене на статистика, и се запазва адреса ѝ.

Конструктори и деструктори:

God() – default

God(string name, double energy, double size, double weight, double strength, State state, Point2D position) – задава съответните стойности

~God () – default

Клас Planet:

Полета:

string name – запазва името на планетата

vector<Entity\*> entities – всички същества от планетата се запазват в този вектор

Методи:

void setName(string name); - [ИМЕТО НА ПЛАНЕТАТА](#)

int getEntities(); - връща съществата

void setEntities(vector<Entity\*>); - задава съществата

void addEntities(EntityType, int); - добавя същества

void DeletePopulation(); - трие същества

void moveEntities(); - движи същества

void detectCollisions(); - мери разстоянието между две същества и проверя дали е възможна атака между тях.

Конструктори и деструктори:

Planet(string) – задава име на планета

Клас RandomGenerator:

Полета:

vector<string> randomNames; - random имената от тип низ се пазят в този вектор

Commented [V1]:

Методи:

string SetRandomName(), double SetRandomX(),SetRandomY(),SetRandomDamage() – задават съответно random име на планета, random координати на съществата и random damage.

Конструктори:

RandomGenerator() – в него се пълни векторът с имена на планети

## Клас CommandManager:

Полета:

string planet – пази името на планетата

int num – броят на съществата

EntityType entity – тип същество(трябва да е от enum-а EntityType)

Методи:

string getPlanet() – връща планетата

int getNum() – връща броя същества

EntityType getEntity() – връща типа същества

void processCommand() – тази функция прави възможна команда от типа “add Earth 5 animal” като запазва броя на съществата, събира ги и ги добавя към съответните планети.

## Клас Scene:

Полета:

Vector <Planet\*> planets – в този вектор ще се запазват всички планети

Методи:

void GeneratePlanets(); void DeletePopulation(string); void ErasePlanet(string); void AddEntities(EntityType, string,int); void FetchStatistics(); void Update(); - функции за добавяне планета, триене населението ѝ, унищожаване планета, добавяне същества, изкарване на статистика, обновяване позициите на съществата

int getPlanets() – връща броя планети;

void setPlanets() – задава планетите;

Конструктори и деструктори:

Scene() – default;

~Scene() – трие планети;

## Клас Simulator:

Полета:

God\* m\_player – член-променлива указател към God, която представлява играча

Scene\* space – член-променлива указател към Scene, която представява галактиката

CommandManager\* command – член-променлива указател към CommandManager, която представява командата

Методи:

void Run() – извиква Menu();

void Menu() – изкарва смислено меню за потребителя

void Update() – обновява състоянието на галактиката

Конструктори и деструктори:

Simulator() – инициализира полетата

~Simulator() – default

IDE: Dev-C++ 5.7.1

