

Техническая документация к программному продукту





ИМИТАЦИОННАЯ ОБЪЕКТНО-ОРИЕНТИРОВАННАЯ МОДЕЛЬ

«ЭХО ПЛУТОНА»

Листов 59

Технический писатель:

Чайко
Владимир Иванович
2 ноября 2023 г.

					Имитационная объектно-ориентированная модель «Эхо Плутона»		
Изм.	Лист	№ докум	Подпись	Дата			
Разраб	Чайко В.И.		02.11.23	ИООМ	Лит.	Лист	Листов
Пров	Чайко В.И.		02.11.23			1	59
Н. Контр.	Чайко В.И.		02.11.23		Г. Новокузнецк		
Утв	Чайко В.И.		02.11.23				

Автор: Чайко В.И.

Этот документ является официальной технической документацией к имитационной объектно-ориентированной модели «Эхо Плутона».

Имитационная объектно-ориентированная модель «Эхо Плутона»:
техническая документация/ Чайко В.И. – г. Новокузнецк, 2023 г. – 59 с., ил.

Данная техническая документация прилагается к имитационной объектно-ориентированной модели «Эхо Плутона» и описывает ее полностью. Она предназначена для тех, кто хочет разобраться в этой модели и имитационном объектно-ориентированном моделировании.

Данная техническая документация может распространяться любыми методами с обязательным указанием ссылки на данный документ.

					Имитационная объектно-ориентированная модель «Эхо Плутона»	Лист
Изм.	Лист	№ докум	Подпись	Дата		2

Содержание

Содержание		3
Введение		5
1. Модель		6
1.1 Изображения.....		6
1.2 Звук.....		6
1.3 Генераторы случайных чисел.....		6
1.4 Устройство колонии на Плуtone.....		7
1.5 Колонист.....		7
1.6 Место взятия пробы.....		9
1.7 Жилой модуль.....		10
1.8 Модуль связи.....		10
1.9 Ядерный реактор.....		11
1.10 Солнечная батарея.....		12
1.11 Кратер.....		13
1.12 3 режима работы модели.....		14
2. Моделирование		16
2.1 Выбор места взятия пробы колонистом.....		16
2.2 Моделирование усталости колониста.....		16
2.3 Моделирование эмоционального состояния колониста.....		21
2.4 Моделирование отдыха.....		23
2.5 Взаимное влияние усталости и состояния.....		24
2.6 Пространство и время в модели.....		25
2.7 Имитация взятия пробы.....		25
2.8 Имитация отдыха.....		26
2.9 Имитация складирования проб.....		26
2.10 Влияние полигонов на колониста.....		26
3. Самообучающийся алгоритм		29
3.1 Самообучающийся алгоритм.....		29
3.2 Сложность алгоритма $O(N)$		29
3.3 Накопление опыта алгоритмом.....		29
3.4 Эффективность.....		31
3.5 Работа алгоритма в потоковом режиме.....		31
4. Нейронная сеть		32
4.1 Нейрон и нейронная сеть.....		32
4.2 Механизм принятия решения нейронной сетью.....		34
4.3 Функция нейронов нейросети модели.....		35
4.4 Механизм генерации начальных значений весов.....		36
4.5 Механизм обучения.....		37
4.6 Технология генеративно-сопоставительных сетей (GAN)		39
4.7 Реализация нейронной сети.....		40
4.8 Алгоритм работы нейронной сети.....		41
5. Результаты моделирования		43
5.1 Результаты.....		43

5.2	Запись в файл CSV.....	43
6.	Обработка результатов.....	46
	Заключение.....	47
	Список источников.....	48
	Приложение А. Список используемых формул.....	50
	Приложение Б. Минимальные системные требования.....	53
	Приложение В. Файлы модели.....	54
	Приложение Г. Возможные ошибки и проблемы.....	56

Введение

Разработана имитационная объектно-ориентированная модель создания колоний на других небесных телах «Эхо Плутона».

В данной модели моделируется создание небольшой колонии на Плутоне для проверки эффективности ее конфигурации.

Данная модель доказывает возможность и эффективность проверки существующих технологий постройки колоний на других небесных телах при помощи имитационного объектно-ориентированного моделирования.

В данной модели имитируется деятельность колониста, живущего на поверхности Плутона, осуществляющего взятие проб поверхности (почвы). В результате своей деятельности колонист устает и нуждается в отдыхе в жилом модуле.

Данная модель имитирует точное протекание процессов, в результате чего результат моделирования складывается из совокупного течения процессов. Предсказать точный конечный результат моделирования невозможно, т.к. он есть результат протекания процессов.

Данная модель доказала, что для осуществления проверки знаний и технологий, в области колонизации небесных тел, возможно применение технологий искусственного интеллекта (самообучающиеся алгоритмы, нейронные сети).

Модели, созданные по такому методу, позволят проверить на пригодность знания о колонизации небесных тел прежде, чем они будут испытаны физически. Это приведет к уменьшению количества затрат человечества на проверку всех накопленных технологий.

Данная техническая документация будет интересна всем занимающимся проблемами освоения космоса, постройки колоний на небесных телах и просто моделированием.

					Имитационная объектно-ориентированная модель «Эхо Плутона»	Лист
Изм.	Лист	№ докум	Подпись	Дата		5

Модель

1.1 Изображения

Для формирования внешнего вида модель использует цветные изображения в формате BMP и глубиной цвета 24 бит.

Формат изображения BMP является простым и распространенным стандартом растровой 2D графики, с которым могут работать все электронные устройства. [1]

Глубина цвета 24 бит обеспечивает передачу 16777216 цветов.

Изображения блоков (построек) колонии, за исключением жилого модуля, были созданы при помощи нейронной сети «Realistic Vision 4.0». [2]

Изображение жилого модуля было создано в графическом редакторе.

Поверхность Плутона и кратеры были созданы на основе реальных фотографий Плутона, сделанных автоматической межпланетной станцией NASA «Новые горизонты» в 2015 году.

1.2 Звук

Модель имеет звуковое сопровождение. Все звуки, используемые моделью, хранятся в формате WAV.

WAV (WAVE) – формат файла-контейнера для хранения звука в цифровом формате без сжатия. Сам же звук хранится в виде импульсно-кодовой модуляции. Данный формат является распространенным. С ним могут работать любые цифровые устройства. [3]

1.3 Генераторы случайных чисел

Для точной имитации, каких либо жизненных процессов, требуется наличие некоторого количества случайности, как и в реальной жизни.

В различных процессах модели требуется генерировать псевдослучайные числа трех типов:

1. Целочисленные
2. Действительные

					Имитационная объектно-ориентированная модель «Эхо Плутона»	Лист
Изм.	Лист	№ докум	Подпись	Дата		6

3. Целочисленные с нормальным распределением.

Для этого модель использует в качестве генераторов случайности 3 генератора случайных чисел:

1. «RAND» - стандартный генератор псевдослучайных целых чисел языка программирования Blitz BASIC.

2. «RND» - стандартный генератор псевдослучайных действительных чисел языка программирования Blitz BASIC.

3. «GENERATOR» - специально созданный в виде функции «generator()» генератор целых псевдослучайных чисел с нормальным распределением. В основе данного генератора лежит стандартный генератор «RAND».

Для генерации разных последовательностей псевдослучайных чисел в качестве начального значения генератора (семени) используется системное время. [4]

1.4 Устройство колонии на Плутоне

Колония состоит из 4-х блоков:

1. Жилой модуль - модуль колонии, в котором осуществляется складирование собранных образцов и отдых колониста.

2. Модуль связи - модуль, осуществляющий функцию связи с Землей.

3. Ядерный реактор - модуль, предназначенный для обеспечения колонии электроэнергией.

4. Солнечная батарея - модуль, предназначенный для обеспечения колонии электроэнергией.

Состав колонии был разработан совместно с нейронной сетью «YaGPT» от отечественной компании «Яндекс». [5]

1.5 Колонист

Колонист - объект, имитирующий человека-колониста. Данный объект описывается 11 переменными, описанными в таблице 1.

					Имитационная объектно-ориентированная модель «Эхо Плутона»	Лист
Изм.	Лист	№ докум	Подпись	Дата		7

Таблица 1. Переменные, описывающие человека

№	Название	Смысл	Принимаемые значения
1	«Внешний вид»	Хранит адрес ячейки памяти, в которой содержится изображение	$\in \mathbb{N}$
2	«X»	Координата по оси X	$0 < X < 801 \in \mathbb{N}$
3	«Y»	Координата по оси Y	$0 < X < 601 \in \mathbb{N}$
4	«Выбор»	Хранит информацию о том, какое место взятия пробы выбрал человек	$0 \leq X \leq 30 \in \mathbb{N}$
5	«Статус»	Хранит информацию о том, идет ли колонист за пробой, или несет его в жилой модуль	$0 \leq X \leq 1 \in \mathbb{N}$
6	«Проб»	Хранит информацию о количестве взятых проб колонистом	$0 \leq X \leq 30 \in \mathbb{N}$
7	«Сила»	Хранит информацию о количестве сил колониста	$0 \leq X \leq 100 \in \mathbb{R}$
9	«Сила N»	Хранит информацию о количестве сил человека	$0 \leq X \leq 100 \in \mathbb{N}$
10	«Психическая сила»	Хранит информацию о количестве психических сил человека	$0 \leq X \leq 100 \in \mathbb{R}$
11	«Психическая сила N»	Хранит информацию о количестве психических сил человека	$0 \leq X \leq 100 \in \mathbb{N}$

Колонист, в результате моделирования, принимает решения на основе всех описанных выше 11 переменных.

В результате моделирования все эти переменные заменяют свои значения. Их изменение зависит как от внешних факторов, влияющих на колониста, так и от внутренних (самого человека). Изменяя свои значения, они «ставят колониста» в различные ситуации, в результате чего он принимает решение о каком-либо действии.

В модели колонист обозначен красным квадратом, как показано

на рисунке 1.



Рисунок 1. Внешний вид колониста

1.6 Место взятия пробы

«Место взятия пробы» - полигон, имитирующий место, в котором планируется взятие пробы. Данный объект описан 3 переменными, представленными в таблице 2.

Таблица 2. Переменные, описывающие место взятия пробы

№	Название	Смысл	Принимаемые значения
1	«X»	Хранит координату дерева по оси X	$0 < X < 700 \in \mathbb{N}$
2	«Y»	Хранит координату дерева по оси Y	$0 < Y < 500 \in \mathbb{N}$
3	«Взято»	Хранит информацию о том, была ли взята проба с данного места.	$0 \leq X \leq 1 \in \mathbb{N}$

Внешний вид места взятия пробы обозначается флагом и представлено на рисунке 2.



Рисунок 2. Место взятия пробы.

Проба считается взятой, если человек выбрал место взятия пробы, пришел на него и взял пробу. Если колонист графически наложился на флаг, проходя мимо, это место считается нетронутым.

1.7 Жилой модуль

«Жилой модуль» – полигон, эмитирующий жилой модуль, в котором живет колонист и хранит пробы. Данный объект описывается 5 переменными, представленными в таблице 3.

Таблица 3. Переменные, описывающие дом.

№	Название	Смысл	Принимаемые значения
1	«Вид»	Хранит адрес ячейки памяти, в которой содержится изображение	$\in \mathbb{N}$
2	«X»	Хранит координату левой верхней точки полигона по оси X	$0 \leq X \leq 696 \in \mathbb{N}$
3	«Y»	Хранит координату левой верхней точки полигона по оси Y	$0 \leq X \leq 511 \in \mathbb{N}$
4	«Вход X»	Хранит координату X входа в жилой модуль.	$0 \leq X \leq 696 \in \mathbb{N}$
5	«Вход Y»	Хранит координату Y входа в жилой модуль.	$0 \leq X \leq 511 \in \mathbb{N}$

Данный полигон используется объектом «колонист» для того, чтобы отдохнуть (восполнить силы), улучшить эмоциональное состояние и хранения взятых образцов. Для того чтобы колонист начал восстанавливать силы и уровень эмоционального состояния, он должен находиться в пределах полигона.

Внешний вид жилого модуля представлен на рисунке 3.



Рисунок 3. Внешний вид модуля.

1.8 Модуль связи

«Модуль связи» – полигон, эмитирующий модуль связи. Данный полигон описывается 3 переменными, представленными в таблице 4.

Таблица 4. Переменные, описывающие модуль связи.

№	Название	Смысл	Принимаемые значения
1	«Вид»	Хранит адрес ячейки памяти, в которой содержится изображение	$\in \mathbb{N}$
2	«X»	Хранит координату левой верхней точки полигона по оси X	$0 \leq X \leq 745 \in \mathbb{N}$
3	«Y»	Хранит координату левой верхней точки полигона по оси Y	$0 \leq X \leq 550 \in \mathbb{N}$

Данный полигон выступает помехой на пути колониста, которую он должен обойти.

Внешний вид жилого модуля представлен на рисунке 4.



Рисунок 4. Внешний вид модуля связи.

1.9 Ядерный реактор

«Ядерный ректор» - полигон, эмитирующий ядерный реактор. Данный полигон описывается 3 переменными, представленными в таблице 5.

Таблица 5. Переменные, описывающие ядерный реактор.

№	Название	Смысл	Принимаемые значения
1	«Вид»	Хранит адрес ячейки памяти, в которой содержится изображение	$\in \mathbb{N}$
2	«X»	Хранит координату левой верхней точки полигона по оси X	$0 \leq X \leq 700 \in \mathbb{N}$
3	«Y»	Хранит координату левой верхней точки полигона по оси Y	$0 \leq X \leq 487 \in \mathbb{N}$

Данный полигон выступает помехой на пути колониста, которую он должен обойти.

Внешний вид жилого модуля представлен на рисунке 5.

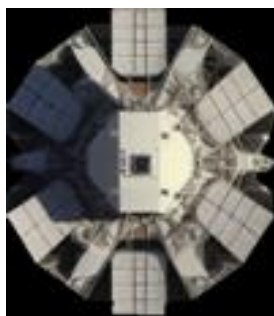


Рисунок 5. Внешний вид ядерного реактора.

1.10 Солнечная батарея

«Солнечная батарея» - полигон, эмитирующий солнечную батарею. Данный объект описывается 3 переменными, представленными в таблице 6.

Таблица 6. Переменные, описывающие солнечную батарею.

№	Название	Смысл	Принимаемые значения
1	«Вид»	Хранит адрес ячейки памяти, в которой содержится изображение	$\in \mathbb{N}$
2	«X»	Хранит координату левой верхней точки полигона по оси X	$0 \leq X \leq 696 \in \mathbb{N}$
3	«Y»	Хранит координату левой верхней точки полигона по оси Y	$0 \leq Y \leq 511 \in \mathbb{N}$

Данный полигон выступает помехой на пути колониста, которую он должен обойти.

Внешний вид солнечной батареи представлен на рисунке 6.

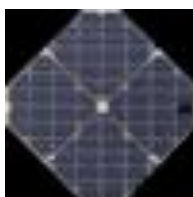


Рисунок 6. Внешний вид солнечной батареи.

1.11 Кратер

«Кратер» – полигон, эмитирующий кратер на поверхности Плутона. Данный объект описывается 3 переменными, представленными в таблице 7. Количество кратеров, при одном проигрывании, варьируется от 0 до 10.

Таблица 7. Переменные, описывающие солнечную батарею.

№	Название	Смысл	Принимаемые значения
1	«Вид»	Хранит адрес ячейки памяти, в которой содержится изображение	$\in \mathbb{N}$
2	«X»	Хранит координату левой верхней точки полигона по оси X	$0 \leq X \leq 700 \in \mathbb{N}$
3	«Y»	Хранит координату левой верхней точки полигона по оси Y	$0 \leq X \leq 500 \in \mathbb{N}$

Данный полигон выступает помехой для колониста. Во время преодоления кратера он (колонист) замедляется и испытывает большую нагрузку. Это влияет на время выполнения задания и усталость (как физическую, так и психическую).

Кратер представляется одним из 5 различных изображений, представленными на рисунках 7, 8, 9, 10 и 11.



Рисунок 7. Кратер (Тип 1).



Рисунок 8. Кратер (Тип 2).



Рисунок 9. Кратер (Тип 3).



Рисунок 10. Кратер (Тип 4) .



Рисунок 11. Кратер (Тип 5) .

1.12 3 режима работы модели

В модель «Эхо Плутона» предусмотрено 3 режима работы:

1. «Человек» – расстановкой построек занимается пользователь модели при помощи компьютерной мышки. Внешний вид режима «человек» представлен на рисунке 12.

2. «Самообучающийся алгоритм» – расстановкой построек занимается самообучающийся алгоритм автоматически. Действия пользователя не требуется. Внешний вид режима «человек» представлен на рисунке 13.

3. «Нейронная сеть» – расстановкой жилого модуля занимается нейросеть, а все остальные модули расставляются случайно. Внешний вид режима «человек» представлен на рисунке 14.

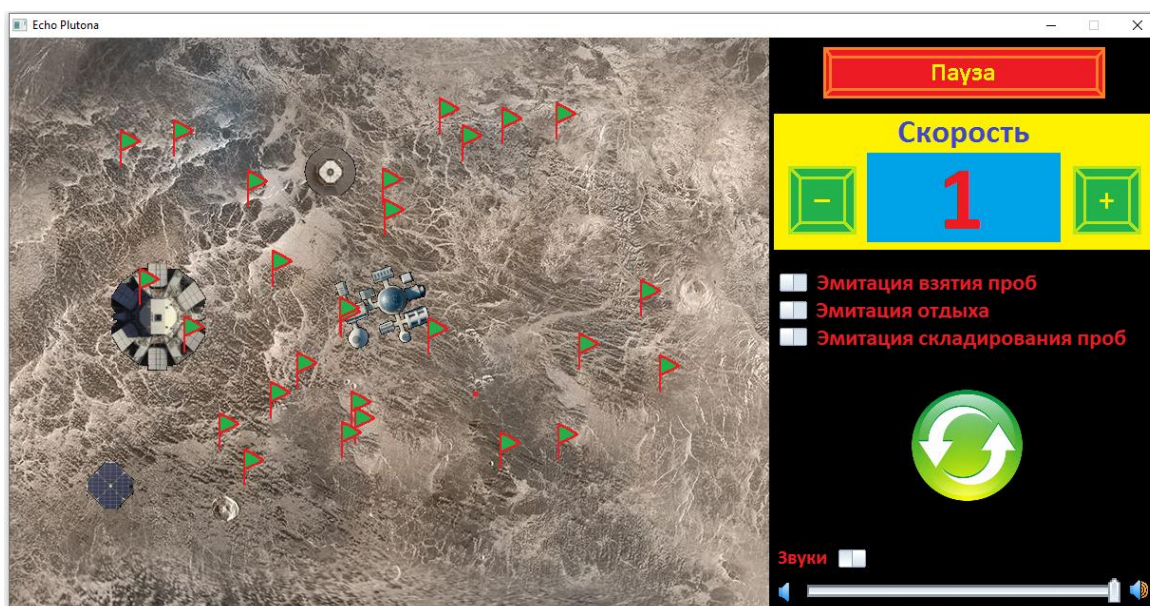


Рисунок 12. Внешний вид режима «Человек»

Изм.	Лист	№ докум	Подпись	Дата

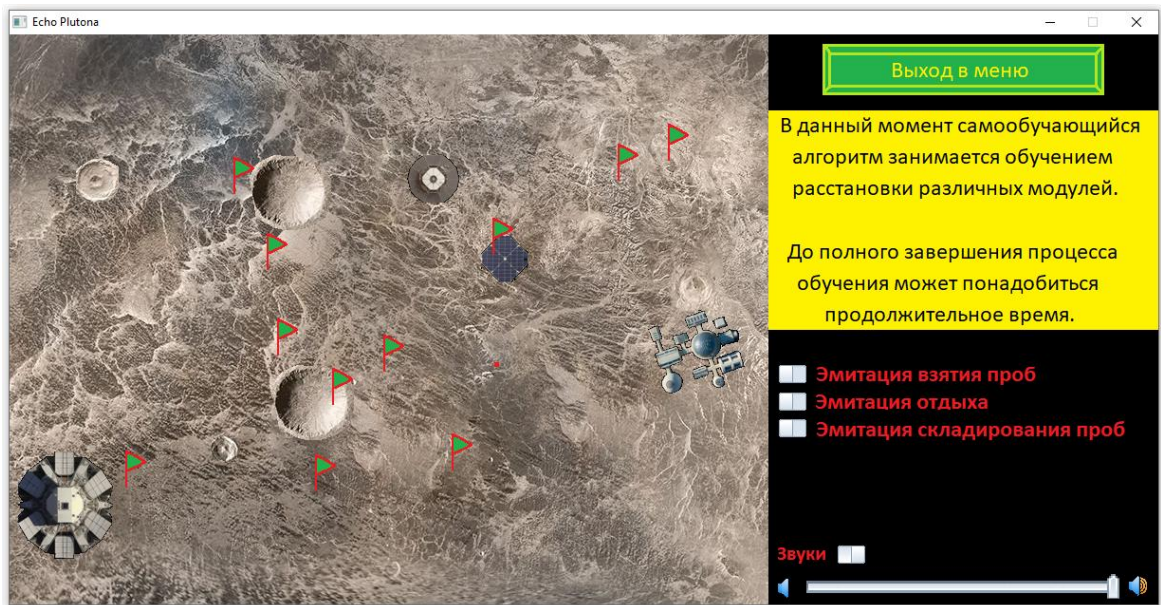


Рисунок 13. Внешний вид режима «самообучающийся алгоритм»

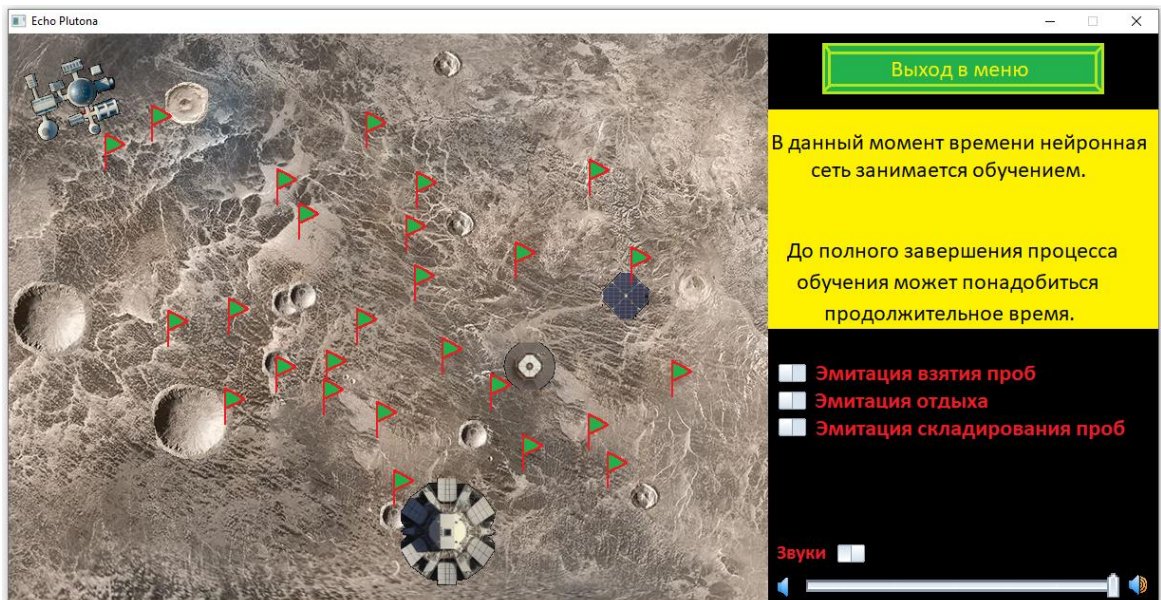


Рисунок 14. Внешний вид режима «нейронная сеть».

Изм.	Лист	№ докум	Подпись	Дата

Моделирование

2.1 Выбор места взятия пробы колонистом

Человек во время моделирования выбирает, в каком месте ему взять пробу. Это осуществляется алгоритмом на основе генератора псевдослучайных чисел GENERATOR. Наглядно данный алгоритм представлен в виде схемы на рисунке 15.

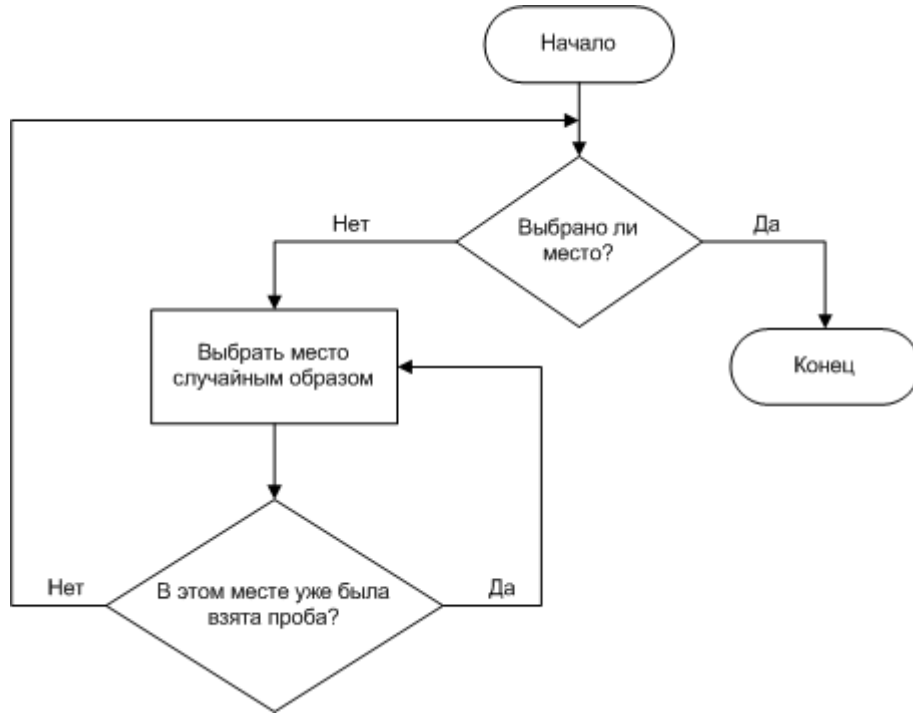


Рисунок 15. Алгоритм выбора места взятия пробы колонистом

2.2 Моделирование усталости колониста

Ключевым параметром, влияющим на деятельность колониста, является его усталость – снижение «запаса сил» во время физических нагрузок.

Сегодня наиболее распространена двухпараметрическая формула критической мощности. Она представляет собой гиперболическую зависимость между мощностью и временем с критической мощностью в виде степенной асимптоты и константы кривизны (1).

$$P = M + \frac{W}{t} \quad (1)$$

где: P – уровень запаса энергии, M – расход сил без физической активности, W – объем работы, который может быть выполнен человеком, t – время выполнения работы.

Изм.	Лист	№ докум	Подпись	Дата

График данной функции представлен на рисунке 16.

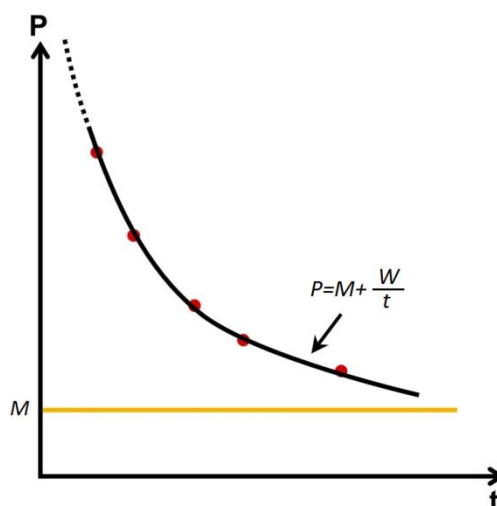


Рисунок 16. График функции (1).

Данная функция очень проста в использовании и поэтому используется чаще всего, в том числе и на международных спортивных соревнованиях. [6]

Смысл данной функции прост: человек, когда не отдыхает, он устает, даже если ничего не делает. После начала какой-либо деятельности, величина расхода энергии увеличивается и зависит от сложности выполняемой работы. Чем сложнее работа – тем выше величина расхода энергии.

Использование в имитационной объектно-ориентированной модели использование формулы (1) невозможно связи с тем, что в ней нет дифференциации труда по уровню сложности. Связи с этим, данная формула требует иного представления.

Для дифференциации работ по уровню энергозатратности, энергозатратность каждой из данных работ вычисляется отдельно по формуле (2).

$$P_n = k_n t_n \quad (2)$$

где: P_n – энергозатратность работы n , k_n – коэффициент энергозатратности работы n , t_n – время, на протяжении которого совершалась работа n .

График функции (2) представлен на рисунке 17.

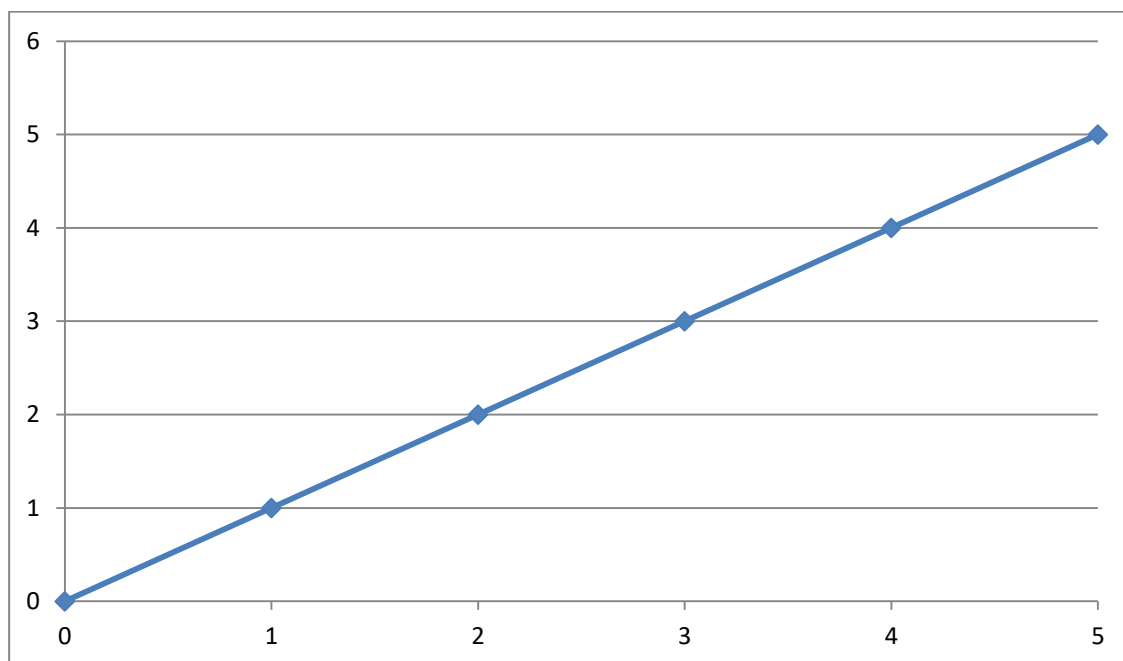


Рисунок 17. График функции (2)

Учитывая, что величина расходуемой энергии (k_n) растет во время выполнения работы (в начале работы человек устает медленнее, чем через некоторое время) формулу (2) требуется дополнить многочленом, позволяющим учитывать эту зависимость (3).

График функции (3) представлен на рисунке 18.

$$P_n = (k_n + (k_f t_n))t_n = k_n t_n + k_f t_n^2 \quad (3)$$

где: P_n – энергозатратность работы n , k_n – коэффициент энергозатратности работы n , k_f – коэффициент увеличения энергозатратности работы n со временем, t_n – время, на протяжении которого совершалась работа n .

Данная функция показывает идеальный график увеличения энергозатратности. В реальной жизни все процессы имеют некоторую случайность и не могут быть представлены в виде четкой линии и зависимости $f(x)$. Чтобы имитационная модель могла имитировать реальный процесс увеличение энергозатратности, коэффициенты k_n и k_f должны быть не константами, а случайным числом из определенного диапазона. Именно поэтому в данной модели, вместо констант используется результат, выданный генератором действительных псевдослучайных чисел «RND».

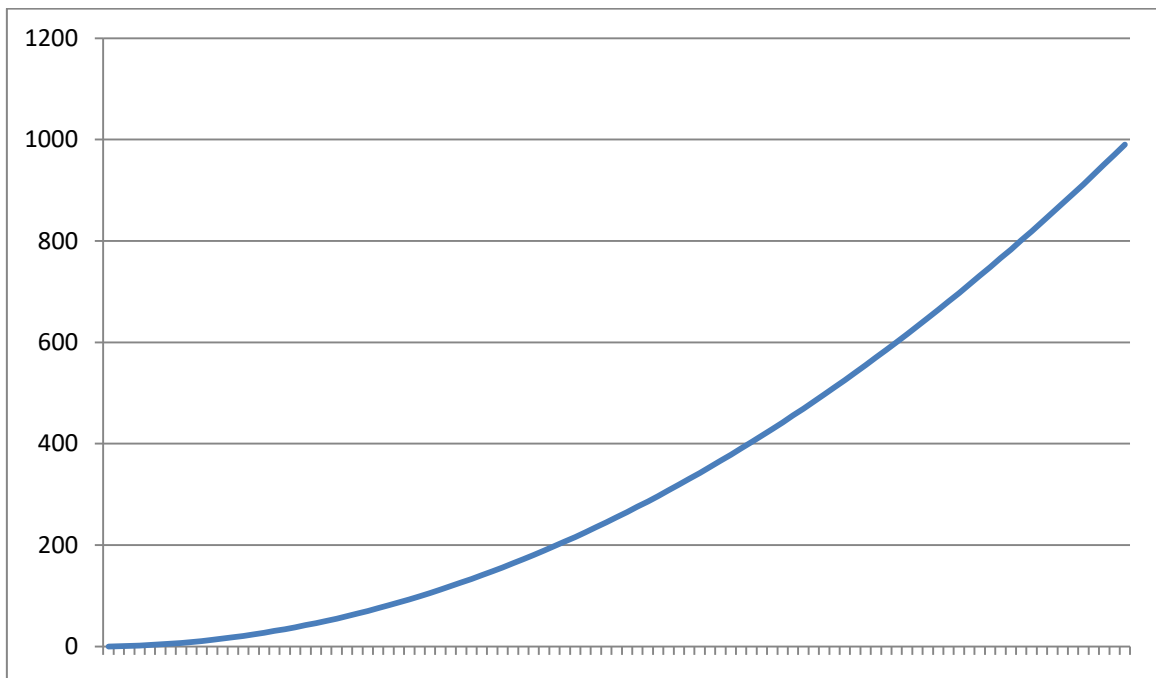


Рисунок 18. График функции (3)

Использование генератора псевдослучайных чисел приводит к тому, что функция не имеет одного графика и каждый раз выдает новый (4). Два получившихся графика представлены на рисунках 19 и 20.

$$P_n = (x + (yt_n))t_n = xt_n + yt_n^2 \quad (4)$$

где:

P_n – энергозатратность работы n ,

x – k_n в диапазоне $a_1 \leq X \leq b_1$,

y – k_f в диапазоне $a_2 \leq Y \leq b_2$,

t_n – время, на протяжении которого совершалась работа n .

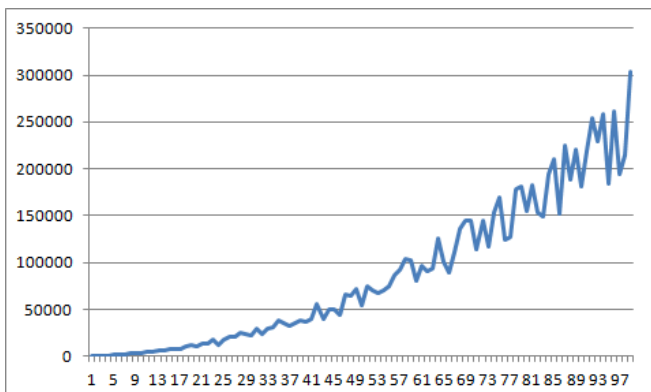


Рисунок 19.
Первый график

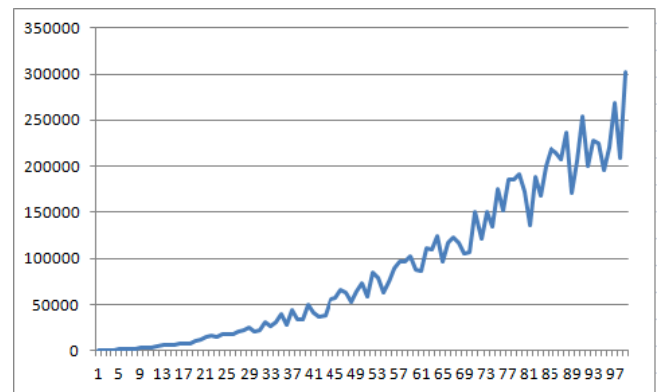


Рисунок 20.
Второй график

Теперь данный график имитирует увеличение усталости работы

Изм.	Лист	№ докум	Подпись	Дата

так, как оно есть в реальной жизни. Колонист, выполняя одну и ту же работу, устает приблизительно одинаково. Именно такое положение вещей и показывают данные графики: они приблизительно одинаковы (имеют небольшие различия). Каждый раз функция будет выдавать новый график.

Модель рассматривает силы колониста как шкалу от 0 до 100%. Таким образом, каждое действие человека отнимает от 100% ту энергию, которую «потратил» человек.

Учитывая, что у нас есть расход сил без физической активности (M), а колонист может выполнять несколько работ параллельно (нести пробы + преодолевать кратер), то общая формула расчета потраченной энергии (5) имеет следующий вид:

$$P = M + P_1 + P_2 + \dots + P_n \quad (5)$$

где:

- P – общий расход сил,
- M – расход сил без физической активности,
- P_1 – трата сил на 1 работу,
- P_2 – трата сил на 1 работу,
- P_n – трата сил на n работу.

Учитывая, что каждая P в формуле (5) является формулой (4), мы имеем следующую формулу расчета усталости (6).

$$P = M + (x_1 t_1 + y_1 t_1^2) + (x_2 t_2 + y_2 t_2^2) + \dots + (x_n t_n + y_n t_n^2) \quad (6)$$

где:

- P – Общее количество энергозатратности,
- M – расход сил без физической активности,
- X_1 – k_1 в диапазоне $a_1 \leq X \leq b_1$,
- X_2 – k_2 в диапазоне $a_2 \leq X \leq b_2$,
- X_3 – k_3 в диапазоне $a_3 \leq X \leq b_3$,
- Y_1 – k_{f1} в диапазоне $a_4 \leq Y \leq b_4$,
- Y_2 – k_{f2} в диапазоне $a_5 \leq Y \leq b_5$,
- Y_3 – k_{f3} в диапазоне $a_6 \leq Y \leq b_6$,
- t_1 – время, на протяжении которого совершалась работа 1,
- t_2 – время, на протяжении которого совершалась работа 2,
- t_n – время, на протяжении которого совершалась работа n .

На рисунках 21 и 22 представлен график расхода силы

колониста, рассчитанный по формуле (6), при условии, что он сначала шел, потом взял пробу (продолжая идти), а затем начал преодолевать кратер.

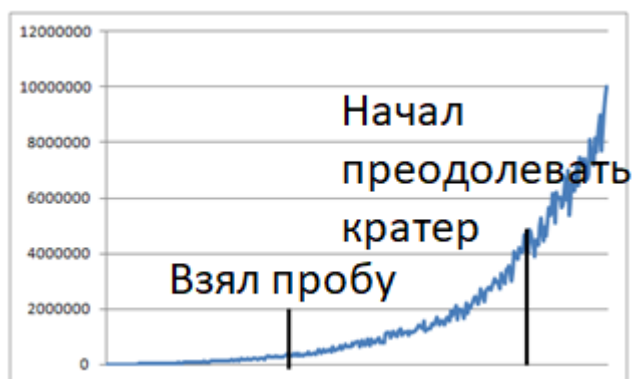


Рисунок 21.
Первый график



Рисунок 22.
Второй график

Для вычисления падения сил колониста модель отнимает от уже имеющихся сил ту, которая требуется ему для выполнения совершаемой работы. Для этого используется формула (7).

$$A = A - P \quad (7)$$

где:

A – количество оставшихся сил,

P – количество расходуемых сил.

График формулы (7) представлен на рисунках 23 и 24.

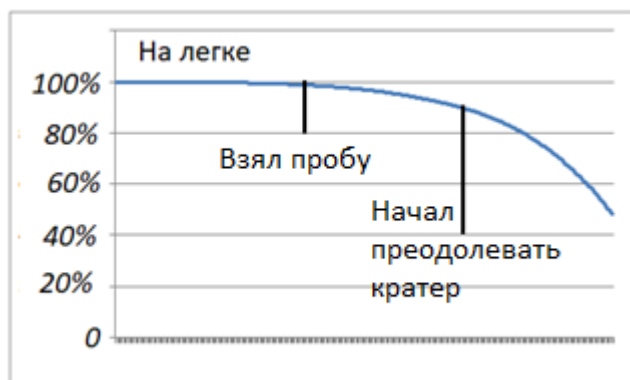


Рисунок 23.
Первый график

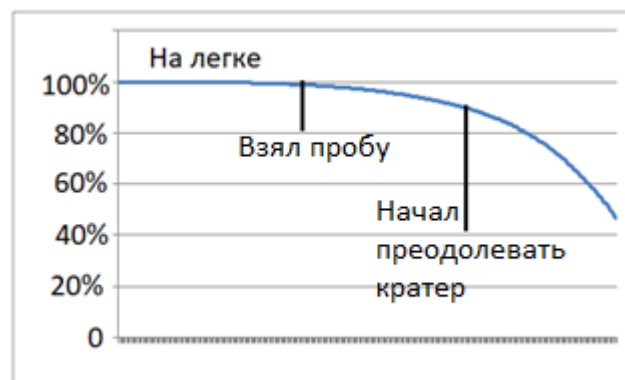


Рисунок 24.
Второй график

2.3 Моделирование эмоционального состояния колониста

Данная модель рассчитывает не только физическое, но и эмоциональное состояние колониста. В данной модели оно (психическое состояние) воспринимается как «запас

Изм.	Лист	№ докум	Подпись	Дата
------	------	---------	---------	------

психической энергии» и уменьшается в результате деятельности человека – затрат психической силы на деятельность.

Учитывая, что суть запасов энергии и запасов психической силы одинакова, ее моделирование (психической силы, эмоционального состояния) моделируется аналогично моделированию запасов энергии.

Формула расчета затрат психической энергии, при совершении конкретной работы, (8) аналогична формуле (4):

$$P_n = (x + (yt_n))t_n = xt_n + yt_n^2 \quad (8)$$

где:

P_n – затраты психической силы при выполнении работы n ,

x – k_n в диапазоне $a_1 \leq X \leq b_1$,

y – k_f в диапазоне $a_2 \leq Y \leq b_2$,

t_n – время, на протяжении которого совершалась работа n .

Два получившихся графика данной функции представлены на рисунках 25 и 26.

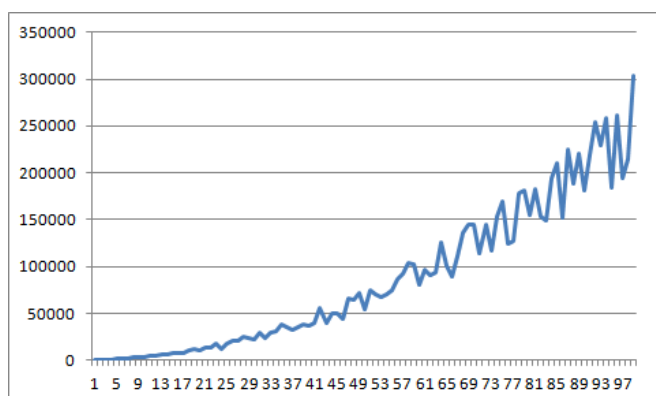


Рисунок 25.
Первый график

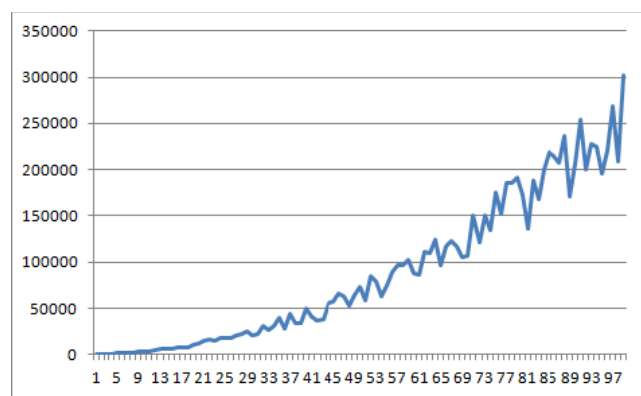


Рисунок 26.
Второй график

Таким образом, общие затраты психической энергии рассчитываются по формуле (9), аналогичной формуле (6).

Для вычисления падения психических сил колониста модель отнимает от уже имеющихся психических сил ту, которая требуется ему для выполнения совершаемой работы. Для этого используется формула (10).

График формулы (7) представлен на рисунках 27 и 28.

$$P = M + (x_1 t_1 + y_1 t_1^2) + (x_2 t_2 + y_2 t_2^2) + \dots + (x_n t_n + y_n t_n^2) \quad (9)$$

где:

P – Общее количество энергозатратности,

M – расход сил без физической активности,

X_1 – k_1 в диапазоне $a_1 \leq X \leq b_1$,

X_2 – k_2 в диапазоне $a_2 \leq X \leq b_2$,

X_3 – k_3 в диапазоне $a_3 \leq X \leq b_3$,

Y_1 – k_{f1} в диапазоне $a_4 \leq Y \leq b_4$,

Y_2 – k_{f2} в диапазоне $a_5 \leq Y \leq b_5$,

Y_3 – k_{f3} в диапазоне $a_6 \leq Y \leq b_6$,

t_1 – время, на протяжении которого совершалась работа 1,

t_2 – время, на протяжении которого совершалась работа 2,

t_n – время, на протяжении которого совершалась работа n .

$$A = A - P \quad (10)$$

где:

A – количество оставшихся сил

P – количество расходующихся сил

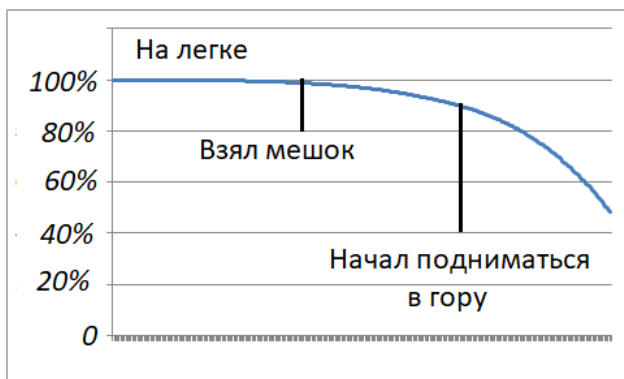


Рисунок 27.
Первый график

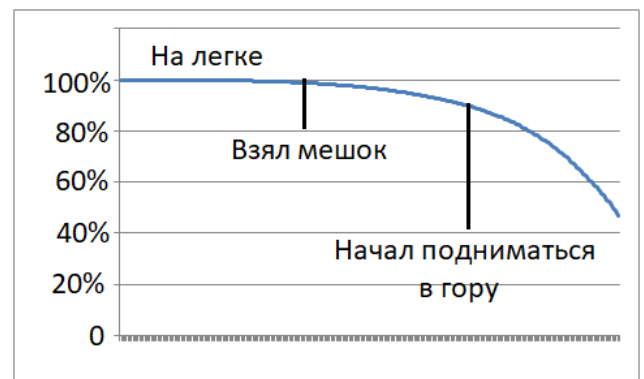


Рисунок 28.
Второй график

2.4 Моделирование отдыха

Отдых в данной модели рассматривается как процесс, противоположный усталости, как физической, так и психической. Таким образом, «отдых» вычисляется по формуле (11).

Два графика формулы (11) представлены на рисунках 29 и 30.

$$A = A + xt_n + yt_n^2 \quad (11)$$

где:

A – количество физической или психической силы человека,

x – k_n в диапазоне $a_1 \leq X \leq b_1$,

y – k_f в диапазоне $a_2 \leq Y \leq b_2$,

t_n – продолжительность отдыха.

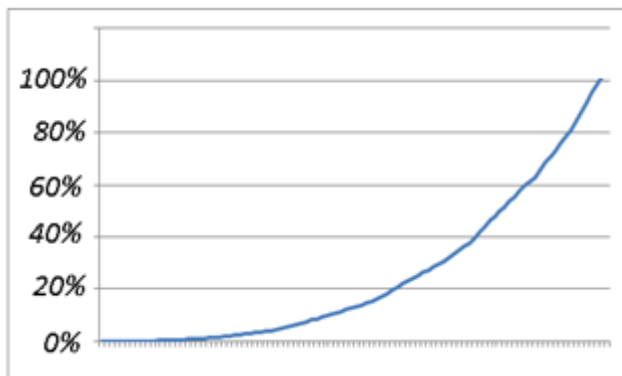


Рисунок 29.
Первый график

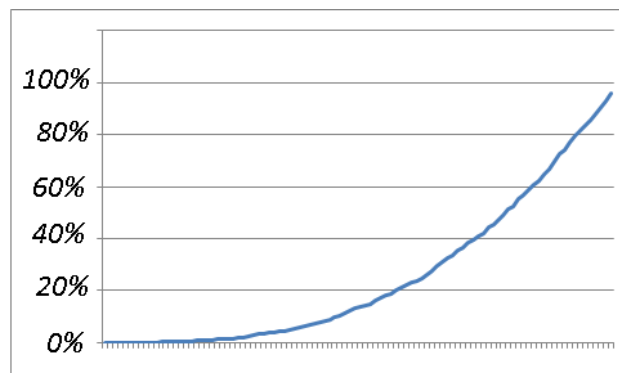


Рисунок 30.
Второй график

2.5 Взаимное влияние усталости и состояния

В реальной жизни физическая и эмоциональная усталости влияют друг на друга:

1. Чем сильнее физически устал человек, тем сильнее расходуется его эмоциональная сила.
2. Чем сильнее психологически устал человек, тем сильнее расходуется его физическая сила.
3. Чем меньше устал физически человек, тем быстрее восстанавливается его психологическая сила.
4. Чем меньше психически устал человек, тем быстрее восстанавливается его физическая сила.

В модели данные зависимости установлены через влияние на коэффициенты-переменные x и y в формулах (4), (6), (8), (9) и (11), как показано на формуле (12).

Значение переменных f_1 и f_2 , для физической усталости, зависит от уровня психической силы («эмоционального состояния»).

Значение переменных f_1 и f_2 , для психической усталости, зависит от уровня физической силы.

Изм.	Лист	№ докум	Подпись	Дата
------	------	---------	---------	------

$$a + f_1 \leq x \leq b + f_2 \quad (12)$$

где:

x – коэффициент, генерируемый генератором «RND»

a – минимальное значение генерируемого коэффициента

b – максимальное значение генерируемого коэффициента

f_1 – число, на которое изменяется a

f_2 – число, на которое изменяется b

2.6 Пространство и время в модели

В данной модели пространство и время связаны между собой следующим образом:

1 единица пространства = 1 метру в реальной жизни,

1 единица времени = 2 секундам в реальной жизни,

Скорость взятия пробы – 5 минут (300 секунд).

Расчет времени, в данной модели, происходит по формуле (13).

$$t = 2T \quad (13)$$

где:

t – количество секунд,

T – количество итераций основного цикла модели.

2.7 Имитация взятия пробы

В данной модели есть возможность включать и выключать имитацию процесса взятия пробы.

При включенной имитации производится процесс взятия пробы дерева так, как если бы он проходил на самом деле. Это приводит к тому, что пользователь данной модели вынужден ждать завершения этого процесса.

При выключенной имитации процесс взятия пробы заменяется быстрым просчетом, что ведет к «быстрому скачку» на 5 минут вперед. При этом в результатах отчета моделирования данный скачок будет виден, т.к. время в модели и запас силы человека (как физической, так и психической) изменятся моментально.

2.8 Имитация отдыха

В данной модели есть возможность включать и выключать имитацию процесса отдыха колониста в жилом модуле.

При включенной имитации производится процесс отдыха так, как если бы он проходил на самом деле. Это приводит к тому, что пользователь данной модели вынужден ждать завершения этого процесса продолжительное время.

При выключенной имитации процесс отдыха заменяется быстрым просчетом, что ведет к «быстрому скачку» вперед по времени и «быстрому скачку» увеличения сил (как физических, так и психологических). При этом в результатах отчета моделирования данный скачок будет виден.

2.9 Имитация складирования проб

В данной модели есть возможность включать и выключать имитацию процесса складирования проб в жилом модуле.

При включенной имитации производится процесс складирования проб так, как если бы он проходил на самом деле. Это приводит к тому, что пользователь данной модели вынужден ждать завершения этого процесса продолжительное время.

При выключенной имитации процесс складирования заменяется быстрым просчетом, что ведет к «быстрому скачку» вперед по времени и «быстрому скачку» уменьшения сил (как физических, так и психологических). При этом в результатах отчета моделирования данный скачок будет виден.

2.10 Влияние полигонов на колониста

Во время своей деятельности, колонист взаимодействует с полигонами «жилой модуль», «модуль связи», «ядерный реактор», «солнечная батарея» и «кратер». Каждый из этих полигонов оказывает свое влияние на человека.

Полигон «кратер» оказывает следующее влияние на человека:

1. Уменьшает скорость движения на 30% из-за сложности передвижения по нему.
2. Увеличивает расход физических и психологических сил колониста в 2 раза.

					Имитационная объектно-ориентированная модель «Эхо Плутона»	Лист
Изм.	Лист	№ докум	Подпись	Дата		26

Влияние происходит за счет уменьшения переменной «скорость колониста» и внесение дополнительного многочлена (4), описывающего влияние на человека, в функцию расчета энергозатратности (6).

Алгоритм осуществления влияния полигона «кратер» на колониста представлен на рисунке 31.

Полигон «жилой модуль» влияет на человека путем увеличения его физического и психического запасов сил. Однако простого нахождения на полигоне недостаточно, человек должен принять решение отдохнуть. Алгоритм влияния полигона «жилой модуль» на человека представлен на рисунке 32.

Полигоны «модуль связи», «ядерный реактор», «солнечная батарея» являются помехой, которую колонист должен обойти. При этом, во время обхода, он скрывается под крышей постройки, т.ч. может показаться, что он прошел сквозь нее.

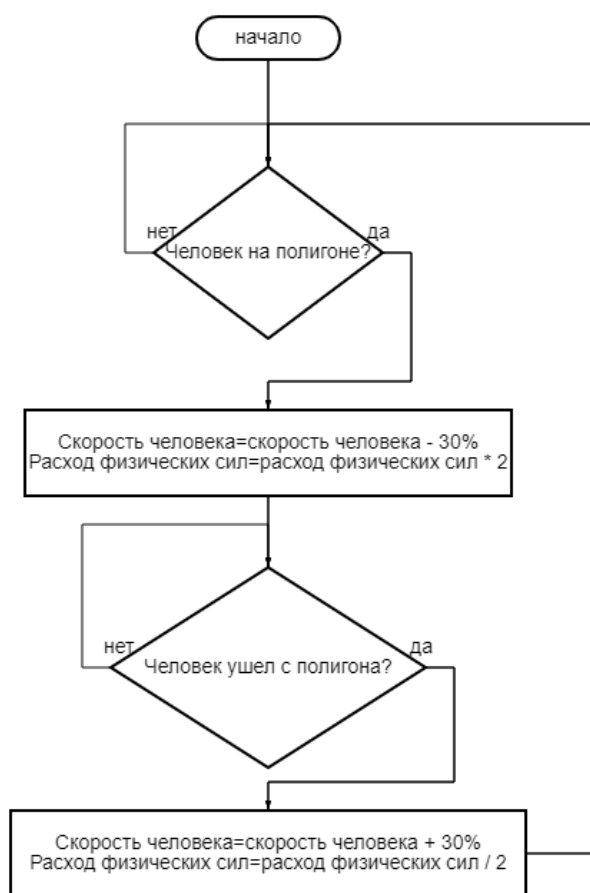


Рисунок 31. Схема алгоритма влияния полигона «кратер» на человека

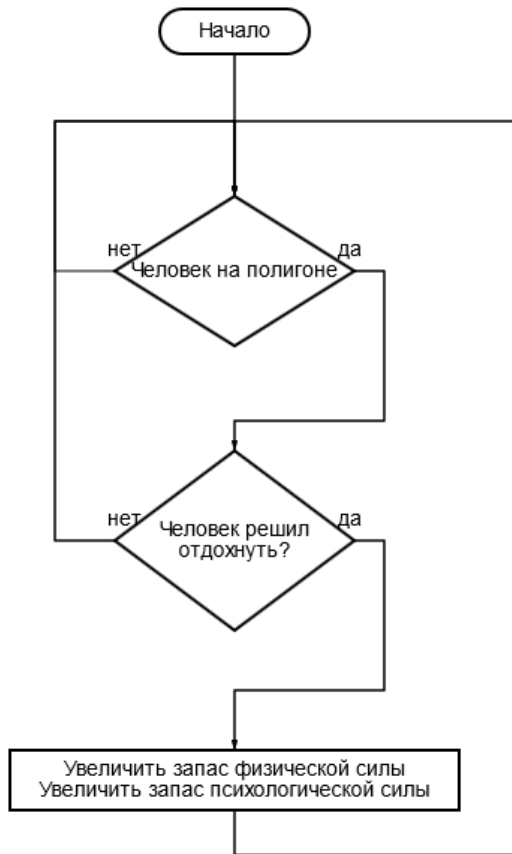


Рисунок 32. Алгоритм влияния полигона «Жилой модуль».

Самообучающийся алгоритм

3.1 Самообучающийся алгоритм

Самообучающийся алгоритм – это алгоритм, осуществляющий перебор всех вариантов решения задачи из множества возможных решений с накоплением статистических данных (опыта).

В модели статистика собирается в двоичном файле «algorithm.model» виде двоичных данных.

Расширение «.model» является уникальным, разработано автором модели и не принадлежит к какой-либо программе.

3.2 Сложность алгоритма $O(N)$

Данный алгоритм является линейным, а его сложность рассчитывается по формуле (14). [7]

$$O(n) = n^i \quad (14)$$

где:

n – количество возможных позиций каждого элемента,

i – количество элементов.

Расчет сложности самообучающегося алгоритма, встроенного в модель, приведен ниже:

$$O(n) = (800 \cdot 600)^{(4+30)} \approx 7 \cdot 10^{198}$$

3.3 Накопление опыта алгоритмом

Накопление опыта алгоритмом происходит в результате накопления данных об эффективности принятых алгоритмом решений.

Общий принцип работы алгоритма:

1. Реализует один из возможных вариантов расстановки построек колонии.
2. Проигрывает модель, определяя эффективность данной расстановки.

3. Сравнивает полученную эффективность с наиболее лучшей из ранее полученных.

4. Если только что проигранный на модели вариант расстановки более эффективен, то модель сохраняет в двоичном формате информацию о новом варианте.

5. Если только что проигранный на модели вариант расстановки менее эффективен, то модель не запоминает его и переходит к пункту 1 данного алгоритма.

Схема работы алгоритма представлена на рисунке 33.

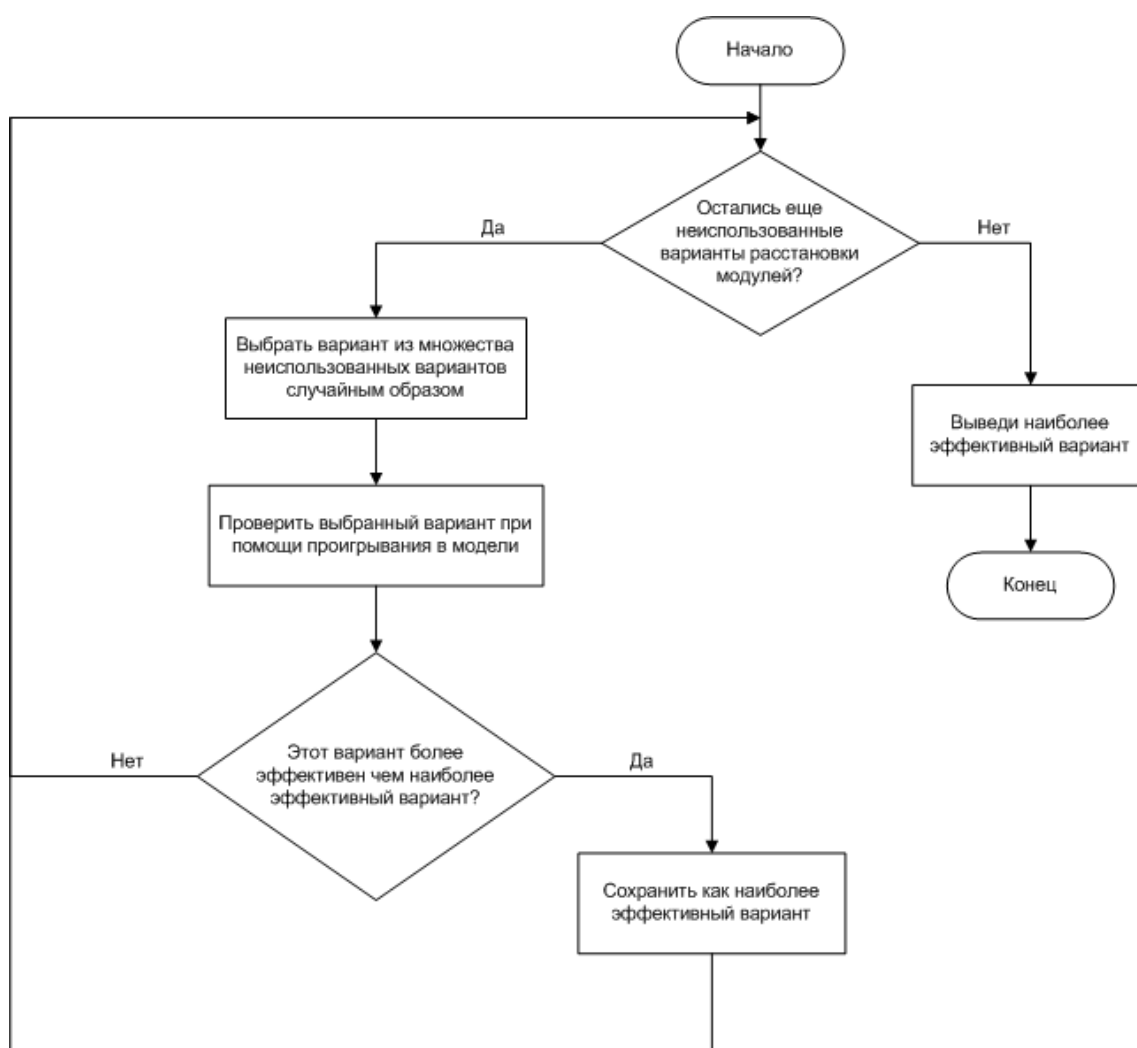


Рисунок 33. Схема работы самообучающегося алгоритма.

В результате перебора всего множества вариантов решений самый лучший вариант расстановки будет сохранен как «наиболее эффективный вариант» в файле «algorithm.csv».

3.4 Эффективность

Цель работы самообучающегося алгоритма – найти наиболее эффективный вариант расстановки строений колонии.

Под эффективностью понимается время, которое было затрачено на выполнение задачи. Чем больше затрачено времени, тем меньше эффективность.

Под наиболее эффективным вариантами расстановки построек понимается такая расстановка, при которой колонист затрачивает наименьшее количество времени на выполнение задачи.

Учитывая, что каждый раз колонисту приходится брать пробы из разного количества мест, в качестве затрачиваемого времени принимается среднее время (15).

$$E = \frac{t}{k} \quad (15)$$

где:

E – эффективность,

t – время, затраченное на взятие всех проб,

k – количество проб.

Чем меньше E , тем более эффективно была осуществлена расстановка построек.

3.5 Работа алгоритма в потоковом режиме

Самообучающийся алгоритм работает параллельно с моделью, получая данные от нее данные.

Таким образом, передача данных организована в виде потока – данные передаются сразу же, а не после окончания работы модели.

Данный метод работы алгоритма называется потоковым режимом.

Нейронная сеть

4.1 Нейрон и нейронная сеть

Нейронная сеть – компьютерная программа, имитирующая работу головного мозга с целью получить его возможности (интеллект).

Под интеллектом понимается способность принимать обоснованные решения на основе имеющегося опыта.

Основной принцип работы нейронной сети, как и головного мозга, заключается в получении сигналов, его обработка и выдача указания.

Обработка сигналов заключается в их поэтапном прохождении слоев, состоящих из нейронов. [8]

Нейрон – минимальная частица головного мозга (или нейросети), способная осуществлять элементарные преобразования сигналов.

Нейроны объединяются между собой, образуя сеть, способную обрабатывать сложную информацию и решать сложные логические задачи.

Биологический нейрон состоит из 3-х частей:

1. Ядро – основание нейрона,
2. Тело – ядро в оболочке,
3. Дендриты – связи, по которым нейрон принимает сигналы
4. Аксон – выход из нейрона (он один)
5. Терминали – ответвления от аксона для подключения к множеству других нейронов. [9]

Наглядно устройство нейрона представлено на рисунке 34.

Общий механизм работы нейрона следующий:

1. Получить все входящие по дендритам данные
2. Суммировать все вошедшие данные
3. Преобразовать сумму данных в зависимости от функции
4. Выдать результат, который будет передан на другие нейроны.

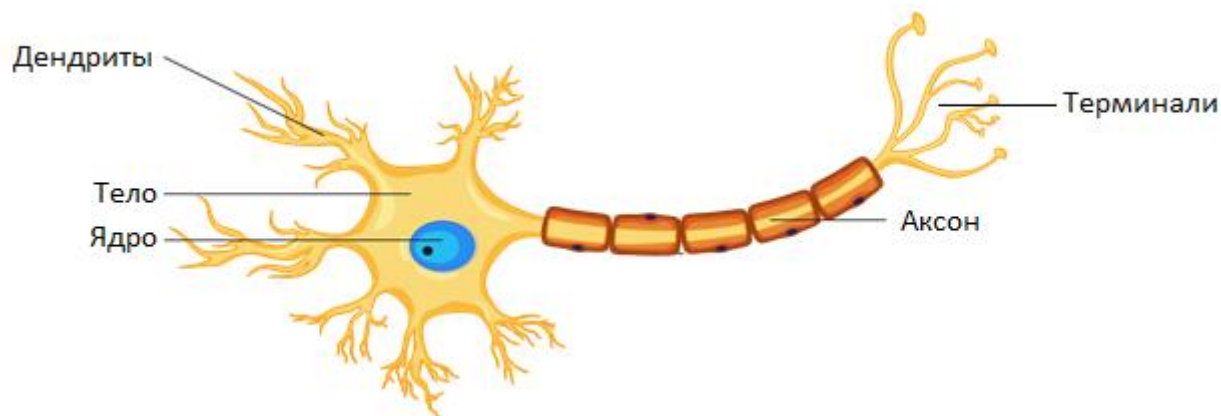


Рисунок 34. Устройство нейрона

В жизни, в качестве данных, природой используются нервные импульсы – электрические сигналы различной величины (разное напряжение).

Для реализации тех же функций, что и у биологического нейрона, не требуется повторять его строение полностью.

В качестве данных, в программном нейроне, используются числовые данные. Величина числа, в данном случае, заменяет силу сигнала.

В нейронных сетях «программный нейрон» состоит всего лишь из 3-х частей:

1. Ядра
2. Дендритов
3. Аксонов

Реализуется «программный нейрон» еще проще – через переменную. [8] Так, чтобы эмитировать работу нейрона, нужно осуществить следующий алгоритм:

1. Взять значения переменных, сложить, сохранить сумму в переменную (вход данных через дендриты от предыдущих нейронов и их суммирование)
2. Преобразовать полученную сумму, приняв ее за аргумент функции. Значение сохранить в эту же переменную. (Обработка нейроном данных).
3. Использовать эту переменную для суммирования с другими переменными и сохранения результата в следующей переменной (передача данных через аксон и терминали).

Учитывая, что нейрон, по своей сути, является простой переменной, то вся нейронная сеть представляет собой массив переменных.

Наглядно устройство программного нейрона изображено на рисунке 35.

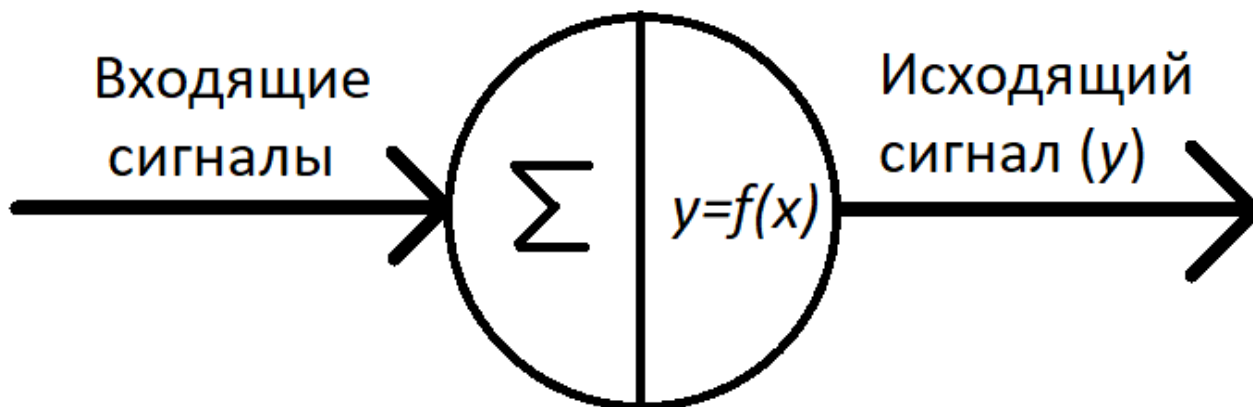


Рисунок 35. Устройство программного нейрона

Каждый нейрон, после суммирования, осуществляет обработку полученной суммы через математическую функцию ($y = f(x)$). Существует большое количество различных функций для нейронов. Наиболее популярными из них являются сигмоид (16), арктангенс (17), функция ошибки (18) и функция Гуддермана (19).

$$y = \frac{1}{1 + e^{-x}} \quad (16)$$

$$y = \arctg(x) \quad (17)$$

$$\frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \quad (18)$$

$$\int_0^x \frac{1}{\cosh(t)} dt \quad (19)$$

4.2 Механизм принятия решения нейронной сетью

После того как исходные данные поступают на вход нейронов первого слоя, начинается принятие решения. Процесс принятия решений происходит благодаря пропусканию данных через нейроны по очереди (очередность устанавливается связями). Проходя через каждый нейрон эти данные преобразуются. Полученные на последнем слое данные и есть решение.

В программном виде это выглядит как подача на вход нейронной сети числовых данных, которые преобразуются в выходные числовые данные. Числовые данные на выходе – это и есть решение. Наглядно данный процесс показан на 3-х слойной нейронной сети из 9 нейронов на рисунке 36.

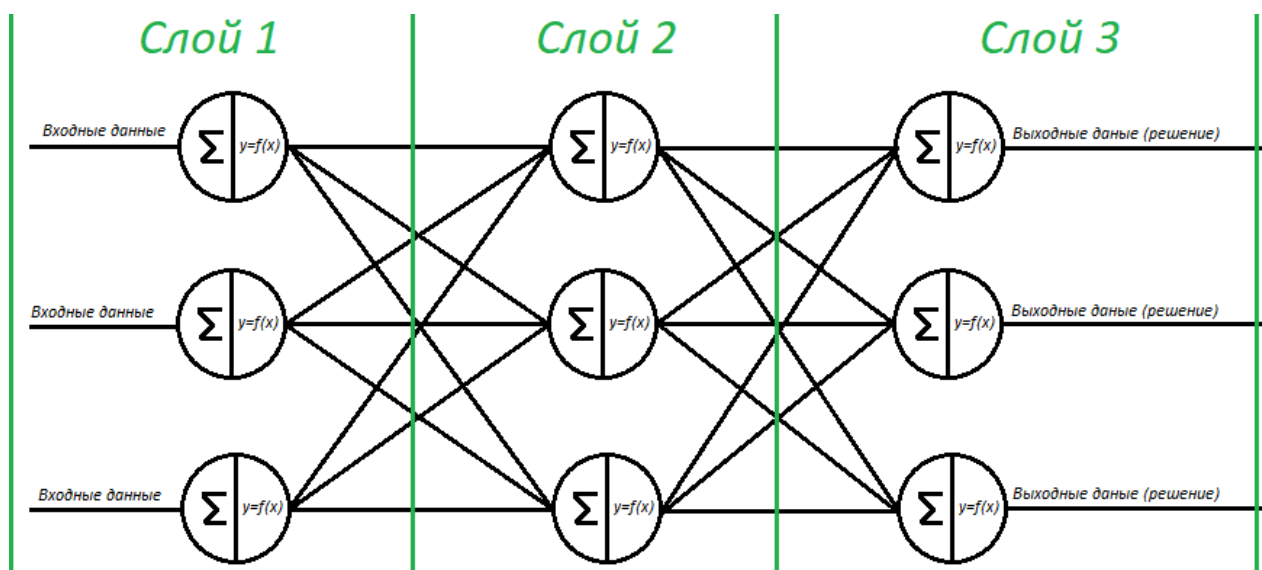


Рисунок 36. Пример принятия решения нейронной сетью

4.3 Функция нейронов нейросети модели

Создаваемая нейронная сеть выдает значения координат жилого модуля. Учитывая, что координаты находятся в диапазоне от 0 до 745 по x ($0 \leq x \leq 745 \in \mathbb{N}$) и от 0 до 550 по y ($0 \leq y \leq 550 \in \mathbb{N}$). Это означает, что реализуемая нейронами функция должна иметь возможность выдавать значения в этих диапазонах. Функции (1), (2), (3) и (4) не подходят, т.к. их значения находятся в диапазоне от 0 до 1 ($0 \leq x \leq 1 \in \mathbb{R}$). Связи с этим, для реализации нейронной сети была использована функция (20):

$$y = kx \quad (20)$$

где:

- y – выходной сигнал,
- k – коэффициент,
- x – сумма входящих сигналов.

Эта функция очень проста, способна к обучению, а ее значения могут находиться в нужных нам диапазонах.

4.4 Механизм генерации начальных значений весов

Перед первым запуском нейронной сети необходимо выставить все коэффициенты k , в функциях нейронов (20), самостоятельно. Значения данных весов должны быть разными.

Задание значений коэффициентов, при первом запуске нейронной сети, в модели «Эхо Плутона», осуществляется через генератор случайных чисел RND.

Диапазон генерирования значений коэффициентов играет весьма значительную роль. Наиболее эффективным является задание коэффициентов в диапазоне $0 \leq X \leq b \in \mathbb{R}$.

Максимальное значение (b) устанавливается по формуле (21) [8].

$$b = \frac{1}{\sqrt{x}} \quad (21)$$

где:

y – максимальное значение диапазона,

x – количество нейронов на предыдущем уровне

На практике применение формулы (21) привело к тому, что нейронная сеть выдает значения координат гораздо больше необходимого диапазона (выходит за пределы игрового поля). Таким образом, чтобы скорректировать диапазон выдаваемых нейронной сетью значений данная формула была модернизирована: введен подстроечный коэффициент k (22).

$$b = \frac{1}{\sqrt{x} \cdot k} \quad (22)$$

где:

y – максимальное значение диапазона,

x – количество нейронов на предыдущем уровне

k – подстроечный коэффициент.

Значение данного коэффициента было установлено опытным путем и равно 5.

Итоговая формула определения значения b приведено в формуле (23):

$$b = \frac{1}{\sqrt{x} \cdot 5} \quad (23)$$

где:

y – максимальное значение диапазона,

x – количество нейронов на предыдущем уровне

Общее математическое описание нахождения значения каждой переменной (нейрона) приведено в формуле (24):

$$n_{(i,j)} = \sum_{j=1}^j n_{(i-1,j)} \cdot k_{(i,j)} \quad (24)$$

где $n_{(i,j)}$ – нейрон j слоя i, $k_{(i,j)}$ – коэффициент выходной связи нейрона j слоя i.

4.5 Механизм обучения

В результате работы нейронной сети накапливается опыт, влияющий на принятие решений – происходит обучение.

Реализуется это за счет изменения коэффициентов k в функции нейронов (20).

Алгоритм, осуществляющий изменение коэффициентов, называется функцией обучения. [8]

Наиболее распространенным методом обучения является алгоритм обратного распространения ошибки. Суть данного метода заключается в следующем: вычисляется ошибка, после чего она «распространяется» на все нейроны – изменяет их значение. Ошибка должна быть дифференцирована: в ней виноваты все слои и все нейроны, поэтому она должна быть учтена каждым нейроном в соответствии с его значимостью. [10]

Данный механизм весьма сложен для понимания и реализации, поэтому данный алгоритм был изменен (упрощен) автором модели.

Для реализации механизма обучения был реализован метод «минимальной добавочности». Суть метода заключается в следующем: корректировка коэффициентов осуществляется при помощи прибавления к нему некоторого небольшого числа (α). Значение данного числа одинаково для всех нейронов. Знак

этого числа (+ или -) определяется знаком вычисленной ошибки.

Алгоритм реализованного метода:

1. Вычислить ошибку нейронной сети
2. Если ошибка положительная и не равна 0, то прибавляемое число должно быть отрицательным ($-\alpha$)
3. Если ошибка отрицательная и не равна 0, то прибавляемое число должно быть положительным ($+\alpha$)
4. Прибавить прибавляемое число ко всем коэффициентам.

Ошибка нейронной сети вычисляется по формуле (25) [8]:

$$O = c - r \quad (25)$$

где:

O - ошибка,

c - правильный ответ,

r - ответ нейронной сети.

Математическое описание метода представлено в формуле (26):

$$\begin{cases} k_{(i,j)} = k_{(i,j)} + (+\alpha), & \text{если } O < 0 \\ k_{(i,j)} = k_{(i,j)} + (-\alpha), & \text{если } O > 0 \end{cases} \quad (26)$$

где: $n_{(i,j)}$ - нейрон j слоя i , $k_{(i,j)}$ - коэффициент выхода нейрона j слоя i , α - коэффициент минимальной добавочности, O - ошибка.

Прибавляемое число (α) определяется опытным путем. От его значения зависит 2 фактора: скорость и точность обучения.

Чем больше это число, тем быстрее скорость обучения, но меньше его точность. Чем меньше это число, тем меньше скорость обучения, но точнее результат.

В модели «Эхо Плутона» данное число было установлено опытным путем и равно 0,01.

Данный алгоритм освобождает от дифференцирования ошибки.

Схема работы алгоритма представлена на рисунке 37:

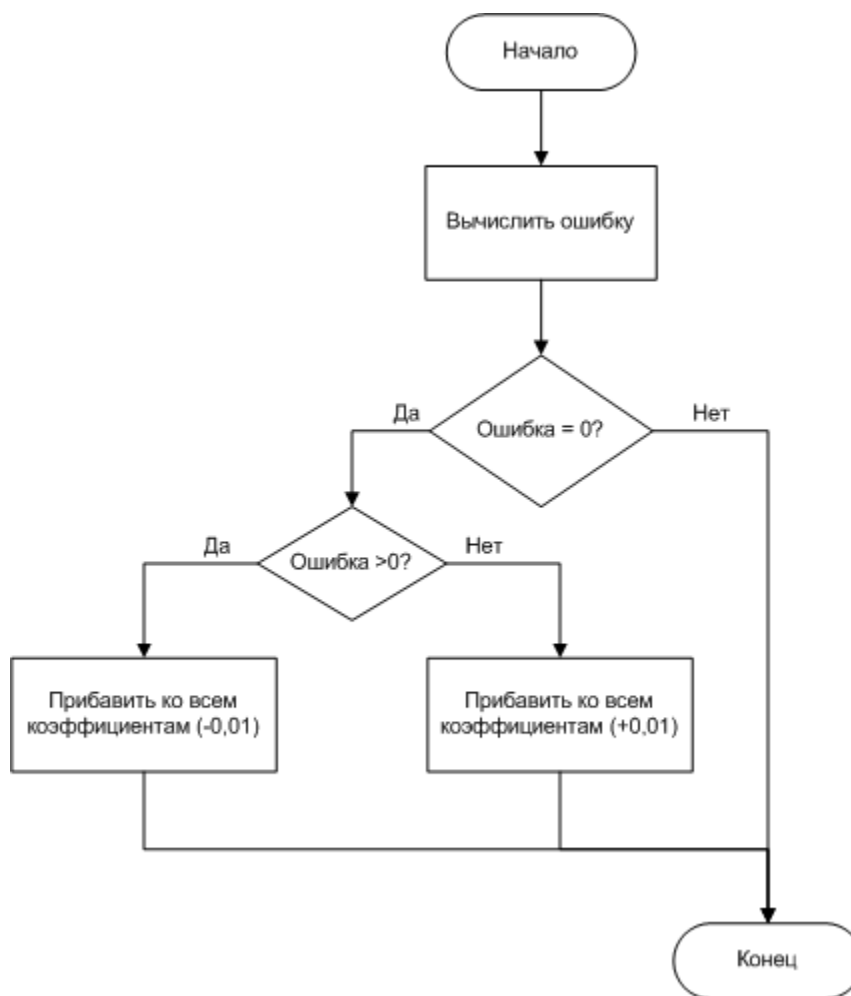


Рисунок 37. Схема работы алгоритма.

4.6 Технология генеративно-сопоставительных сетей (GAN)

Технология генеративно-сопоставительных сетей заключается в идее объединить нейронную сеть с генератором заданий. Генератор создает задания для нейронной сети, которая учится, выполняя их. Таким образом, из процесса образования исключается человек. Это приводит к значительному ускорению обучения.

Генеративно-сопоставительная сеть (GAN) состоит из двух программ:

1. Генератор – программа, генерирующая задания.
2. Дискриминатор – нейронная сеть, которая обучается, выполняя задания. [11]

Использование в модели «Эхо Плутона» нейронной сети реализует эту технологию: сама модель является программой,

которая дается задание для решения (генератором), а нейронная сеть обучается на них (дискриминатор).

4.7 Реализация нейронной сети

Реализация нейронной сети в модели «Эхо Плутона» была осуществлена по принципу «не усложняй» (KISS principle). [12]

На первый слой нейронной сети подаются значения координат всех мест, где требуется осуществить взятие пробы. Максимальное число таких мест – 30. Каждое из них имеет 2 координаты: X и Y. Таким образом на вход нейронной сети подается не более 60 значений. Этим и обусловлен тот факт, что первый слой имеет 60 нейронов.

Ответ нейронной сети состоит из 2-х чисел: координат X и Y жилого модуля колониста. Этим обусловлен тот факт, что количество нейронов выходного слоя равно 2.

Чтобы снизить количество вычислений, и сложность нейронной сети, было решено сделать каждый последующий слой в 2 раза меньше предыдущего. Каждый последующий слой должен быть меньше предыдущего не более чем в 2 раза, чтобы не снизить эффективность работы нейронной сети.

Минимальное необходимое количество слоев нейронной сети вычисляется по формуле (27):

$$N = \log_n m \quad (27)$$

где:

N – минимальное необходимое количество слоев,
n – во сколько раз каждый слой будет меньше предыдущего,
m – число нейронов в первом слое.

Учитывая, что количество нейронов на первом слое равно 60 ($m=60$), а коэффициент уменьшения был выбран 2 ($n=2$), количество слоев должно быть не менее 6 (28).

Таким образом, нейронная сеть состоит из 6 слоев по 60, 30, 15, 8, 4 и 2 нейрона соответственно. Общее количество нейронов – 119.

Реализована нейронная сеть в виде 19 одномерных массивов

$$N = \log_2 60 \cong 5,9 \approx 6$$

(28)

где:

N – минимальное необходимое количество слоев,
 n – во сколько раз каждый слой будет меньше предыдущего,
 m – число нейронов в первом слое.

разной размерности, состоящих из переменных типа Float.

Чертеж нейронной сети представлен на рисунке 38.

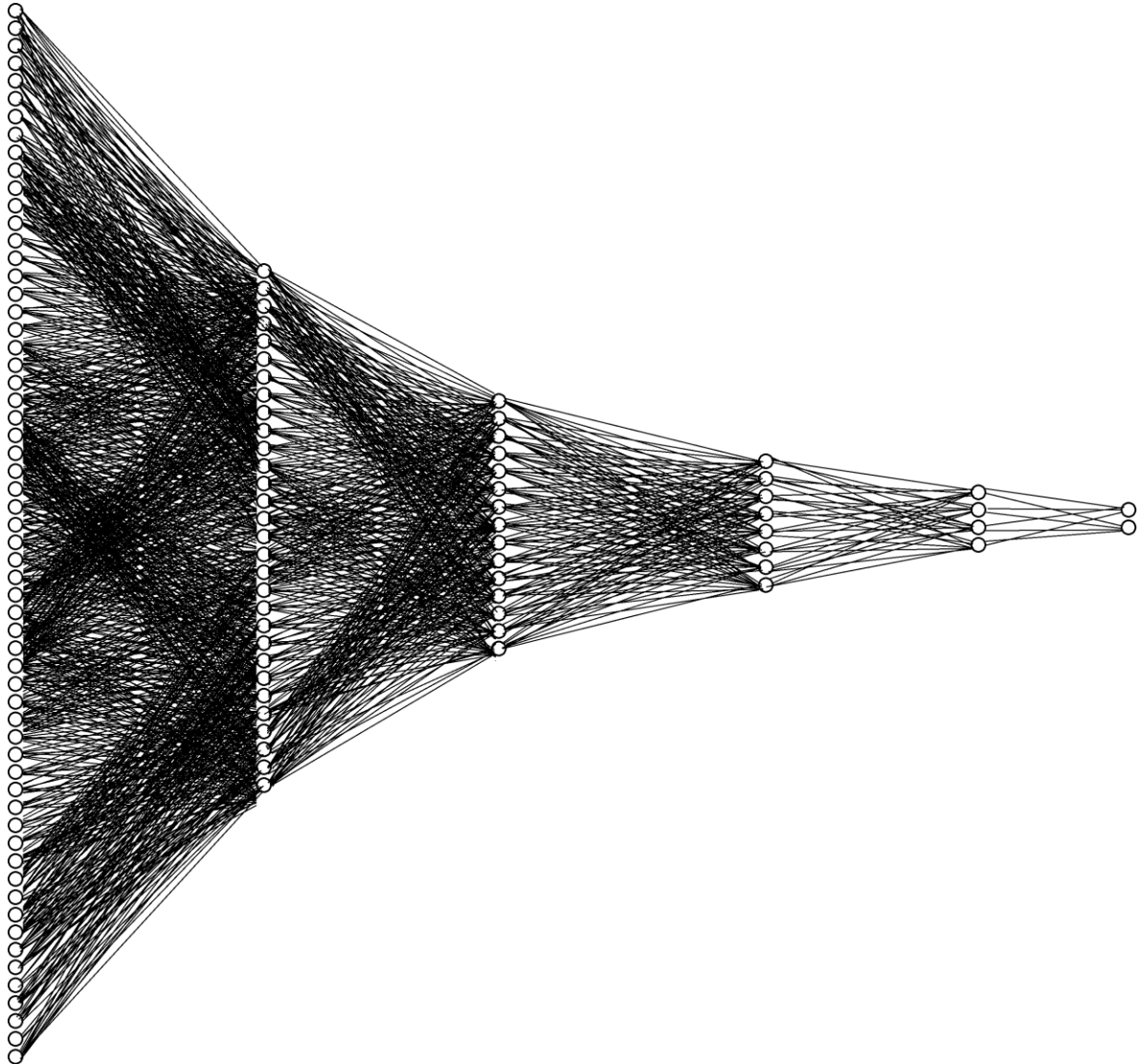


Рисунок 38. Чертеж нейронной сети.

4.8 Алгоритм работы нейронной сети

Во время моделирования нейронная сеть осуществляет накопление опыта, на основании которого делает предположения о том, как лучше расставлять постройки

колонии. После она проверяет эти предположения на эффективность при помощи модели. Результат моделирования становится новым опытом. Этот процесс циклический и повторяется до тех пор, пока нейронная сеть не выработает правила правильной расстановки модулей относительно ландшафта.

Алгоритм данного процесса представлен на рисунке 39.



Рисунок 39.

По своей сути, нейронная сеть решает задачу регрессии – взять координаты всех мест, где требуется взять пробу, и выдать координаты, где лучше всего поставить жилой модуль колониста. [13] Для решения этой задачи используется метод градиентного спуска. [14]

Изм.	Лист	№ докум	Подпись	Дата

Результаты моделирования

5.1 Результаты

Результатом моделирования является не только итоговая картинка, изображенная на экране, но и цифровые данные. Список собираемых цифровых данных представлен в таблице 8.

Таблица 8. Список собираемых данных

Параметр	Значение	Единица
Режим «Человек»		
Итерации	$\in \mathbb{N}$	Штук
Время	$\in \mathbb{N}$	Секунды
Сила	$\in \mathbb{N}$	Проценты
Сила	$\in \mathbb{R}$	Проценты
Психическая сила	$\in \mathbb{N}$	Проценты
Психическая сила	$\in \mathbb{R}$	Проценты
Количество взятых проб	$\in \mathbb{N}$	Штуки
Режим «Самообучающийся алгоритм»		
Эффективность	$\in \mathbb{N}$	Единиц
Время	$\in \mathbb{N}$	Секунд
Координаты всех объектов	$\in \mathbb{N}$	X и Y
Режим «Нейронная сеть»		
Значение коэффициентов нейронов	$\in \mathbb{R}$	число

5.2 Запись в файл csv

Все результаты моделирования автоматически записываются в соответствующие файлы:

- Режим «Человек» - файл «oth.csv».
- Режим «самообучающейся алгоритм» - файл «algorithm.csv».
- Режим «нейронная сеть» - файл «neiro.csv».

Все они находятся в одной папке с файлом запуска модели

Изм.	Лист	№ докум	Подпись	Дата
------	------	---------	---------	------

«echo_pkutona.exe».

Формат хранения файлов CSV является общедоступным форматом хранения табличных данных и может быть открыт в любой аналитической программе, например «Excel», «Google Sheets» и «Calc», или в какой-либо СУБД (базе данных), в том числе «MySQL», «1C» и «Access».

Такой способ хранения данных позволяет свободно их передавать в виде файла и открывать в любой удобной для конкретного пользователя программе для их анализа. [15]

Все данные записаны в следующем формате: каждая строка таблицы записывается в виде новой строчки, а ее разделение на столбцы осуществляется знаком «точка с запятой» («;»). Наглядно формат записи изображен на рисунке 40. Результат открытия данного формата в программе «Excel» изображен на рисунке 41.

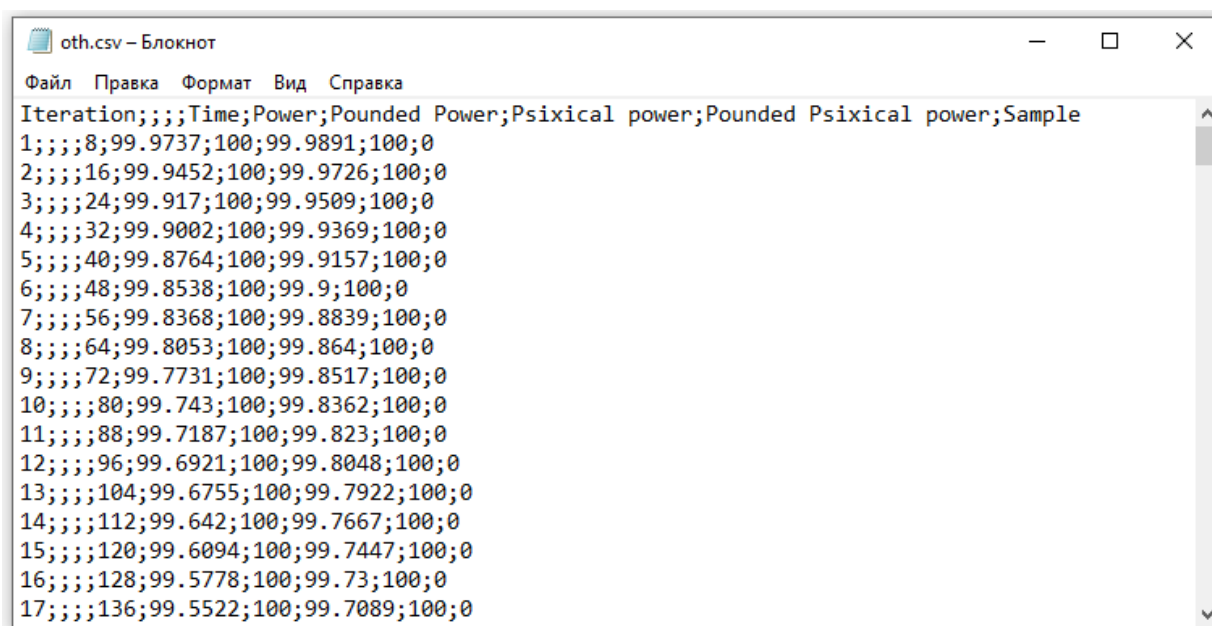


Рисунок 40. Наглядное изображение формата записи результатов

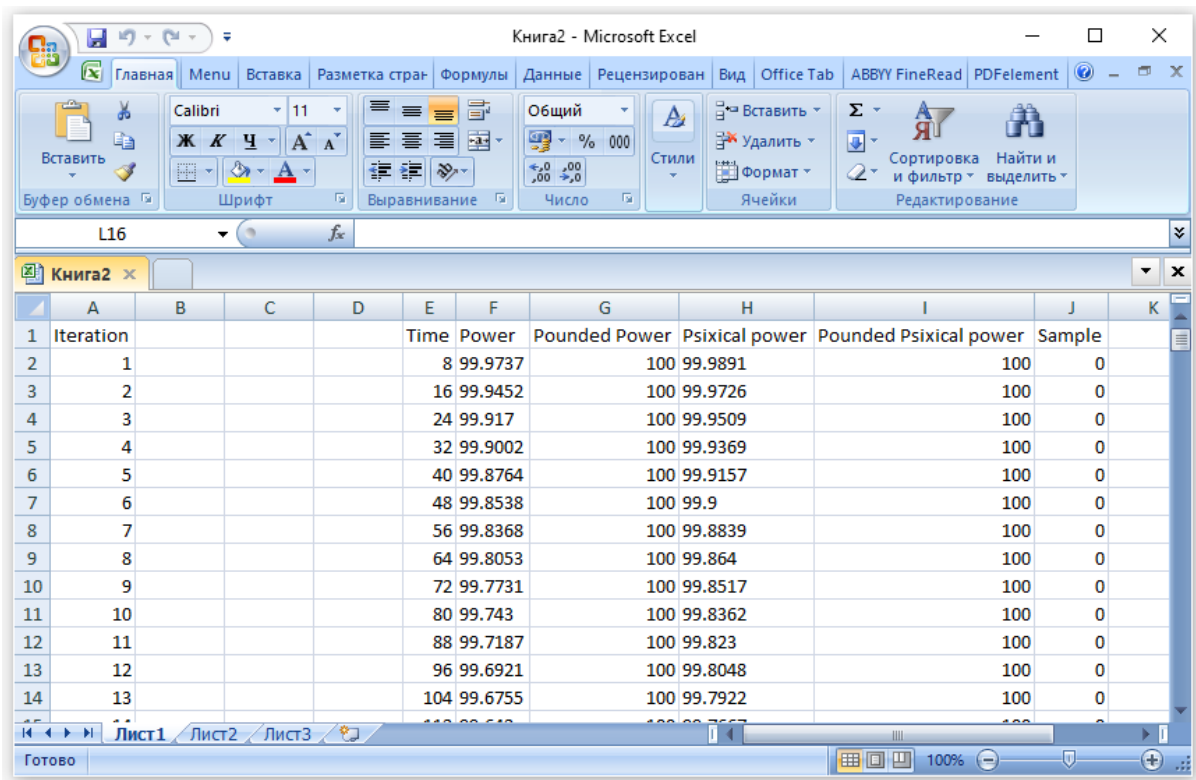


Рисунок 41. Результат открытия данного формата в программе «Excel»

Обработка результатов

Для обработки результатов необходимо импортировать CSV файл в любую удобную пользователю программу анализа данных.

После импорта пользователь может использовать абсолютно весь инструментарий его аналитической программы.

					Имитационная объектно-ориентированная модель «Эхо Плутона»	Лист
Изм.	Лист	№ докум	Подпись	Дата		46

Заключение

В данной технической документации описана имитационная объектно-ориентированная модель «Эхо Плутона».

Данная модель была создана для того, чтобы доказать пригодность использования имитационного объектно-ориентированного моделирования, самообучающихся алгоритмов и нейронных сетей для проверки существующих и разработки новых технологий колонизации небесных тел.

Учитывая, что модель имитирует все процессы так, как они протекали бы в реальной жизни, чтобы узнать конечный результат, ее нужно «проиграть» полностью. Результат моделирования, как и в реальной жизни, складывается из совокупного течения процессов, поэтому невозможно предсказать, каким образом они повлияют друг на друга и к чему приведут.

Основная цель данной документации – дать достаточно полное описание созданной модели, ее принципах и методах.

					Имитационная объектно-ориентированная модель «Эхо Плутона»	Лист
Изм.	Лист	№ докум	Подпись	Дата		47

СПИСОК ИСТОЧНИКОВ

1. Bitmap Image File (BMP), Version 5. // Sustainability of Digital Formats: Planning for Library of Congress Collection. URL: <https://www.loc.gov/preservation/digital/formats/fdd/fdd000189.shtml> (дата обращения: 24.11.2022.).
2. Realistic Vision 4 & Stable Diffusion: Creating Striking AI Images (Tutorial Guide) | ChatBots.top. – Текст : электронный // chatGPT SHOP : [сайт]. – URL: <https://chatgptshop.org/blog/realistic-vision-4-stable-diffusion-creating-striking-ai-images-tutorial-guide-chatbots-top/#:~:text=Realistic%20Vision%204%20is%20an,your%20AI-generated%20images%20to%20life> (дата обращения: 23.10.2023).
3. Audio File Format Specifications. // Telecommunications & Signal Processing Laboratory. URL: <https://www-mmssp.ece.mcgill.ca/Documents/AudioFormats/WAVE/WAVE.html> (дата обращения: 24.11.2022.).
4. Sethi M. Game programming. – Second edition. – THOMSON course technology, 2003.
5. YandexGPT 2. – Текст : электронный // YandexGPT : [сайт]. – URL: <https://ya.ru/gpt/2> (дата обращения: 23.10.2023).
6. Sreedhara V.S.M. Gregory M.M., Hutchison R.E. A survey of mathematical models of human performance using power and energy. // Sports medicine - Open. – 2019. – №54. URL: <https://doi.org/10.1186/s40798-019-0230-z> (дата обращения: 24.11.2022).
7. Knuth, Donald. The Art of Computer Programming. "Section 6.1: Sequential Searching". Sorting and Searching. / Donald Knuth. – 3rd ed. – : Addison-Wesley, 1998. – 780 с. – Текст : непосредственный.
8. Rashid, Tariq Make Your Own Neural Network / Tariq Rashid. – : CreateSpace, 2016. – 222 с. – Текст : непосредственный.
9. Нервная ткань. Нейрон. Синапс. Нервы. – Текст : электронный // ЯКласс : [сайт]. – URL: <https://www.yaklass.ru/p/biologia/8-klass/nervnaia-reguliatciia-16071/stroenie-nervnoi-sistemy-i-ee-znachenie-16072/re-85e19745-be2f-4f82-9208-060f40f25bff> (дата обращения: 31.10.2023).

					Имитационная объектно-ориентированная модель «Эхо Плутона»	Лист
Изм.	Лист	№ докум	Подпись	Дата		48

10. Werbos, P. J. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences / P. J. Werbos. – Cambridge: Harvard University, 1974. – 453 с. – Текст : непосредственный.
11. Goodfellow, I. J. Generative Adversarial Nets / I. J. Goodfellow. – Текст : непосредственный // University of Montreal. – 2014.
12. The Unix Philosophy in One Lesson. – Текст : электронный // Applying the Unix Philosophy. : [сайт]. – URL: <http://www.catb.org/~esr/writings/taoup/html/ch01s07.html> (дата обращения: 24.10.2023).
13. McCaffrey, J. Тесты - Регрессия нейронной сети. / J. McCaffrey. – Текст : электронный // Microsoft Ignite. : [сайт]. – URL: <https://learn.microsoft.com/ru-ru/archive/msdn-magazine/2016/march/test-run-neural-network-regression> (дата обращения: 24.10.2023).
14. Lemarechal, C. Cauchy and the Gradient Method. / C. Lemarechal. – Текст : непосредственный // Documenta Math. – 2012. – № . – С. 251-254.
15. Clive H., Chambers M., Keast A. CSV-1203. CSV File Format Specification. – First Edition. – Published in the United Kingdom: mastpoint.com, 2013. – 31 с. URL: https://arquivo.pt/wayback/20160305185525mp_/http://mastpoint.curzonnassau.com/csv-1203/csv-1203.pdf (дата обращения: 24.11.2022).

Приложение А. Список используемых формул.

№	Функция
(1)	$P = M + \frac{W}{t}$
(2)	$P_n = k_n t_n$
(3)	$P_n = (k_n + (k_f t_n)) t_n = k_n t_n + k_f t_n^2$
(4)	$P_n = (x + (y t_n)) t_n = x t_n + y t_n^2$
(5)	$P = M + P_1 + P_2 + \dots + P_n$
(6)	$P = M + (x_1 t_1 + y_1 t_1^2) + (x_2 t_2 + y_2 t_2^2) + \dots + (x_n t_n + y_n t_n^2)$
(7)	$A = A - P$
(8)	$P_n = (x + (y t_n)) t_n = x t_n + y t_n^2$
(9)	$P = M + (x_1 t_1 + y_1 t_1^2) + (x_2 t_2 + y_2 t_2^2) + \dots + (x_n t_n + y_n t_n^2)$
(10)	$A = A - P$
(11)	$A = A + x t_n + y t_n^2$
(12)	$a + f_1 \leq x \leq b + f_2$
(13)	$t = 2T$
(14)	$O(n) = n^i$

(15)	$E = \frac{t}{k}$
(16)	$y = \frac{1}{1 + e^{-x}}$
(17)	$y = \arctg(x)$
(18)	$\frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$
(19)	$\int_0^x \frac{1}{\cosh(t)} dt$
(20)	$y = kx$
(21)	$b = \frac{1}{\sqrt{x}}$
(22)	$b = \frac{1}{\sqrt{x} \cdot k}$
(23)	$b = \frac{1}{\sqrt{x} \cdot 5}$
(24)	$n_{(i,j)} = \sum_{j=1}^j n_{(i-1,j)} \cdot k_{(i,j)}$
(25)	$O = c - r$

(26)	$\begin{cases} k_{(i,j)} = k_{(i,j)} + (+\alpha), & \text{если } O < 0 \\ k_{(i,j)} = k_{(i,j)} + (-\alpha), & \text{если } O > 0 \end{cases}$
(27)	$N = \log_n m$
(28)	$N = \log_2 60 \cong 5,9 \approx 6$

Приложение Б. Минимальные системные требования

Характеристика	Значение
CPU	Intel Pentium G4400 OEM (3,3 ГГц) AMD FX-4300 BOX (3,8 ГГц) или более поздние
ОС	Windows 95/98/ME/2000/XP/Vista/7/8/10
RAM	256 Мбайт
HDD	18 Мбайт
Видеокарта	GeForce 210 (1 Гбайт ОЗУ) или более поздняя
Разрешение экрана	1280x1024 или выше
DirectX	9.0 или выше
Звуковая карта	16 бит (48 кГц) или выше
Контроллеры	Мышь, клавиатура
Дополнительно	Наушники или колонки

Изм.	Лист	№ докум	Подпись	Дата

Имитационная объектно-ориентированная
модель «Эхо Плутона»

Лист

53

Приложение В. Файлы модели

Название	Тип	Расширение	Объем (Кбайт)
echo_plutona.exe	Исполнимый файл	exe	1696
block.bmp	Изображение	bmp	28
chel.bmp	Изображение	bmp	1
crat1.bmp	Изображение	bmp	21
crat2.bmp	Изображение	bmp	25
crat3.bmp	Изображение	bmp	5
crat4.bmp	Изображение	bmp	7
crat5.bmp	Изображение	bmp	4
flag.bmp	Изображение	bmp	3
knopka_zastav.bmp	Изображение	bmp	28
menu.bmp	Изображение	bmp	739
menu_vibor.bmp	Изображение	bmp	1191
menu_vibor_knop1.bmp	Изображение	bmp	19
menu_vibor_knop2.bmp	Изображение	bmp	18
menu_vibor_knop3.bmp	Изображение	bmp	82
menu_vibor_knop4.bmp	Изображение	bmp	82
menu_vibor_knop5.bmp	Изображение	bmp	28
menu_vibor_knop6.bmp	Изображение	bmp	28
menu_vibor2.bmp	Изображение	bmp	1737
menu2.bmp	Изображение	bmp	739
menu3.bmp	Изображение	bmp	739
menu4.bmp	Изображение	bmp	739
paus1.bmp	Изображение	bmp	48
paus2.bmp	Изображение	bmp	48
paus3.bmp	Изображение	bmp	48
paus4.bmp	Изображение	bmp	48
perekluch1.bmp	Изображение	bmp	2
perekluch2.bmp	Изображение	bmp	2
polzunok.bmp	Изображение	bmp	2
powerx.bmp	Изображение	bmp	1531
reactor.bmp	Изображение	bmp	34
scorost.bmp	Изображение	bmp	168
scorost_max1.bmp	Изображение	bmp	16
scorost_max2.bmp	Изображение	bmp	16
scorost_min1.bmp	Изображение	bmp	16
scorost_min2.bmp	Изображение	bmp	16
scorost1.bmp	Изображение	bmp	7
scorost2.bmp	Изображение	bmp	8
scorost3.bmp	Изображение	bmp	8
sonlc_panel.bmp	Изображение	bmp	8
svaz.bmp	Изображение	bmp	9

Изм.	Лист	№ докум	Подпись	Дата
------	------	---------	---------	------

text.bmp	Изображение	bmp	77
text_algoritm.bmp	Изображение	bmp	279
text_neiro.bmp	Изображение	bmp	279
text_zastav.bmp	Изображение	bmp	360
verx.bmp	Изображение	bmp	238
vixod1.bmp	Изображение	bmp	48
vixod2.bmp	Изображение	bmp	48
zanovo1.bmp	Изображение	bmp	41
zanovo2.bmp	Изображение	bmp	44
zastav.bmp	Изображение	bmp	1055
zelob.bmp	Изображение	bmp	13
zvuk_minus.bmp	Изображение	bmp	1
zvuk_plus.bmp	Изображение	bmp	2
zvuki.bmp	Изображение	bmp	3
knop_menu_vibor.wav	Звуковой файл	wav	49
molot.wav	Звуковой файл	wav	36
pause.wav	Звуковой файл	wav	349
perekluchatel.wav	Звуковой файл	wav	14
stroika.wav	Звуковой файл	wav	457
zanovo.wav	Звуковой файл	wav	266
zapret.wav	Звуковой файл	wav	159
zastav.wav	Звуковой файл	wav	827
Общий вес:			14606,75

Изм.	Лист	№ докум	Подпись	Дата

Имитационная объектно-ориентированная
модель «Эхо Плутона»

Лист

55

Приложение Г. Возможные ошибки и проблемы

Ошибка/Сложность	Описание и решение
<p>Модель, в результате запуска, сама открылась и закрылась несколько раз.</p>	<p>Причина :</p> <p>Антивирусная программа запустила модель в «безопасном режиме», параллельно проверяя ее на вирусы.</p> <p>Исправление :</p> <p>После проверки антивирус убедится, что в модели нет вирусов, и Вы можете закрыть и открыть модель заново – больше она сама открываться и закрываться не будет.</p>
<p>Модель не запускается, выдавая окно ошибки с надписью «Stream does not exist».</p>	<p>Причина :</p> <p>У Вас открыт один из CSV файлов. В то время как он открыт, модель не может обратиться к нему и записать туда результаты своего моделирования.</p> <p>Исправление :</p> <p>Закройте файл и запустите модель заново.</p>
<p>В открытом CSV файле неправильные данные, ошибки, некорректные данные из-за чего невозможно построить графики и анализировать результаты.</p>	<p>Причина :</p> <p>В результате открытия был изменен формат данных, изменены сами данные (округлились числа) либо данные считались некорректно.</p>

Изм.	Лист	№ докум	Подпись	Дата

Исправление :

Некоторые программы, например «Excel», могут автоматически при открытии файлов «подгонять» их под формат, округляя и изменяя данные. Чтобы избежать такого, необходимо открывать файл путем его импорта:

«Данные» - «Получить внешние данные» -> «Из текста»

Data -> Get External Data -> From Text

Во время моделирования модель, иногда, подвисает.

Причина :

Генератор псевдослучайных чисел не может генерировать большое количество случайных чисел с такой скоростью, как требуется модели. Это приводит к тому, что модель вынуждена ждать, когда он выдаст ей необходимое случайное число.

Исправление :

Данную проблему можно решить, используя процессоры с более быстрыми генераторами псевдослучайных чисел. Однако задержки модели не приводят к искажению результатов моделирования, поэтому кроме некоторого неудобства данная проблема не принесет.

Изм.	Лист	№ докум	Подпись	Дата

Во время моделирования, иногда, кажется, что колонист не взял пробу.

Причина :

Память видеокарты работает не так быстро, как этого требует модель. Это приводит к тому, что она не успевает отобразить на экране удаление флага.

Исправление :

Данную проблему можно решить, используя видеокарты с более быстрой памятью. Однако данная проблема не сказывается на результатах моделирования, поэтому кроме некоторого неудобства данная проблема не принесет.

Во время моделирования колонист проходит сквозь постройки колонии.

Причина :

На самом деле колонист обходит эти постройки. Он так близко к стенке, что скрывается под крышей строения, поэтому его не видно.

Исправление :

Это не ошибка.

График, построенный на основе данных CSV файла неправильный.

Причина :

Программа по оси X указывает не количество секунд, а номер строки записи файла CSV.

Исправление :

Для построения графика воспользуйтесь точечным типом диаграммы, указав

Изм.	Лист	№ докум	Подпись	Дата

первым аргументом значение столбца «Time».

					Имитационная объектно-ориентированная модель «Эхо Плутона»	Лист
Изм.	Лист	№ докум	Подпись	Дата		59