

## **Изменения:**

### **1. Инфекция стала 3-х ступенчатой:**

Стадия 1 (латентная):

- Опухоли маленькие, не видны
- Очень низкая передача ( $\times 0.1$ )
- Низкая смертность (2% максимум)
- Большинство выживает

Стадия 2 (инфекционная):

- Опухоли видны, развиваются
- Высокая передача ( $\times 5.0$ ), особенно половая
- Умеренная смертность (10% максимум)
- Значительная часть умирает

Стадия 3 (терминальная):

- Опухоли обширные, изъязвленные
- Максимальная передача ( $\times 10.0$ )
- Высокая смертность (20% ежедневно + 100% в конце)
- Практически все умирают

Общая выживаемость за 240 дней: <5% (реалистично для DFTD)

### **2. Исправления смерти:**

1. Исправлена смертность в стадии 2 - теперь использует **stage2\_duration** вместо **stage1\_duration**
2. Добавлена стадия 3 с отдельной логикой смертности
3. Унифицирована терминология (стадии вместо фаз)

4. Разделены параметры для каждой стадии (инфекционность и смертность)
5. Добавлены флаги контроля передачи по путям и стадиям

## По файлам:

### 1. ФАЙЛ: constants.py

#### НОВЫЕ КОНСТАНТЫ ИНФЕКЦИОННОСТИ:

```
python
# ---- INFECTION TRANSMISSION PARAMETERS ----
INFECTIVITY1: float = 0.3      # Базовая вероятность половой передачи (дни 0-10)
INFECTIVITY2: float = 0.01     # Базовая вероятность неполовой передачи

# ---- STAGE-SPECIFIC TRANSMISSION MULTIPLIERS ----
STAGE1_TRANSMISSION_MULTIPLIER: float = 0.1    # Латентная стадия (очень низкая)
STAGE2_TRANSMISSION_MULTIPLIER: float = 5.0     # Инфекционная стадия
STAGE3_TRANSMISSION_MULTIPLIER: float = 10.0    # Терминальная стадия

# ---- TRANSMISSION CONSTRAINTS ----
STAGE1_CAN_TRANSMIT_SEXUAL: bool = False       # Латентная НЕ передает половым путем
STAGE1_CAN_TRANSMIT_CONTACT: bool = True        # Латентная передает контактным (с очень низкой вероятностью)
STAGE2_CAN_TRANSMIT_SEXUAL: bool = True          # Инфекционная передает всеми путями
STAGE2_CAN_TRANSMIT_CONTACT: bool = True
STAGE3_CAN_TRANSMIT_SEXUAL: bool = True          # Терминальная передает всеми путями

STAGE3_CAN_TRANSMIT_CONTACT: bool = True
```

#### НОВЫЕ КОНСТАНТЫ СМЕРТНОСТИ:

```
python
# ---- DISEASE MORTALITY FACTORS BY STAGE ----
DISEASE_MORTALITY_FACTOR_STAGE1: float = 0.02   # 2% максимум для латентной стадии
DISEASE_MORTALITY_FACTOR_STAGE2: float = 0.10    # 10% максимум для инфекционной стадии
```

```
DISEASE_MORTALITY_FACTOR_STAGE3: float = 0.20 # 20% максимум для терминальной  
стадии (плюс 100% в конце)
```

## НОВЫЕ ИМЕНА КОНСТАНТ ДЛЯ СТАДИЙ БОЛЕЗНИ:

```
python
```

```
# ---- INFECTION STAGES ----  
INFECTION_STAGE_HEALTHY: int = 0 # Здоров  
INFECTION_STAGE_LATENT: int = 1 # Латентная (0.5-1.0 года, нет передачи)  
INFECTION_STAGE_INFECTIOUS: int = 2 # Инфекционная (0.25-1.0 года, передача)  
INFECTION_STAGE_TERMINAL: int = 3 # Терминальная (0.5-1.0 года, 100% смертность)
```

## НОВЫЕ КОНСТАНТЫ ДЛИТЕЛЬНОСТЕЙ СТАДИЙ:

```
python
```

```
# ---- STAGE DURATIONS (в днях) ----  
STAGE1_DURATION_MIN: int = 60 # 0.5 года = 60 дней  
STAGE1_DURATION_MAX: int = 120 # 1.0 года = 120 дней  
STAGE2_DURATION_MIN: int = 30 # 0.25 года = 30 дней  
STAGE2_DURATION_MAX: int = 120 # 1.0 года = 120 дней  
STAGE3_DURATION_MIN: int = 60 # 0.5 года = 60 дней  
STAGE3_DURATION_MAX: int = 120 # 1.0 года = 120 дней
```

## НОВЫЕ КОНСТАНТЫ УСИЛЕНИЯ ПЕРЕДАЧИ:

```
python
```

```
# ---- TRANSMISSION BOOSTS ----  
STAGE2_TRANSMISSION_BOOST: float = 5.0 # Инфекционная стадия в 5 раз заразнее  
STAGE3_TRANSMISSION_BOOST: float = 10.0 # Терминальная стадия в 10 раз заразнее
```

## 2. ФАЙЛ: infection.py

**НОВАЯ ФУНКЦИЯ:** calculate\_transmission\_probability

**Формула:**

text

Вероятность = StageMultiplier × DistanceFactor × (SexualComponent + NonsexualComponent)

## Аргументы:

python

```
def calculate_transmission_probability(  
    state: SimulationState,  
    transmitter_indices: Tensor,  
    susceptible_indices: Tensor,  
    dist_sq_matrix: Tensor,  
    day_in_year: int,  
    infectivity1: float = INFECTIVITY1,      # 0.3  
    infectivity2: float = INFECTIVITY2,      # 0.01  
    stage1_multiplier: float = STAGE1_TRANSMISSION_MULTIPLIER, # 0.1  
    stage2_multiplier: float = STAGE2_TRANSMISSION_MULTIPLIER, # 5.0  
    stage3_multiplier: float = STAGE3_TRANSMISSION_MULTIPLIER, # 10.0  
    breeding_days: int = BREEDING_DAYS,  
    device: th.device = DEVICE  
  
) -> Tensor:
```

## Компоненты формулы:

### 1. DistanceFactor:

python

```
distance_factor = th.clamp((MAX_RADIUS_SQ - dist_sq_matrix) / MAX_RADIUS_SQ, 0,  
                           1)  
    ○ Если расстояние = 0 → factor = 1  
    ○ Если расстояние = MAX_RADIUS → factor = 0  
    ○ Линейная зависимость от расстояния
```

### 2. StageMultiplier:

- Стадия 1 (латентная): ×0.1
- Стадия 2 (инфекционная): ×5.0
- Стадия 3 (терминальная): ×10.0

### 3. SexualComponent:

- Только в сезон размножения (дни 0-10)
- Только между взрослыми (STATUS\_ADULT)
- Зависит от STAGE\*\_CAN\_TRANSMIT\_SEXUAL флагов
- Значение: infectivity1 (0.3)

### 4. NonsexualComponent:

- Всегда доступна
- Для всех резидентов
- Зависит от STAGE\*\_CAN\_TRANSMIT\_CONTACT флагов
- Значение: infectivity2 (0.01)

## **ОБНОВЛЕННАЯ ФУНКЦИЯ: infection\_spread**

### **Новая логика:**

1. Прогрессия болезни: Переходы между стадиями 1→2→3
2. Инициализация длительностей: Для новых зараженных
3. Определение кто может передавать: Все резиденты на стадиях 1-3
4. Расчет вероятностей: Через calculate\_transmission\_probability
5. Применение заражения: Для каждого восприимчивого берется максимальная вероятность

### **Ключевое изменение:**

```
python

# Старая логика:
if day_in_year < breeding_days:
    # Сложная смесь половой и контактной передачи
else:
    # Только контактная передача

# Новая логика:
prob_matrix = calculate_transmission_probability(...) # Единая формула
```

### **НОВАЯ ЛОГИКА СТАДИЙ:**

#### **Стадия 1 → Стадия 2:**

```
python
transition_to_stage2 = stage1_mask & (state.age_of_disease[:n] >= state.stage1_duration[:n])
```

#### **Стадия 2 → Стадия 3:**

```
python
transition_to_stage3 = stage2_mask &
    (state.age_of_disease[:n] >= (state.stage1_duration[:n] + state.stage2_duration[:n]))
```

### **3. ФАЙЛ: simulation\_core.py (функция process\_all\_deaths)**

#### **ПОЛНОСТЬЮ ПЕРЕПИСАННАЯ ФУНКЦИЯ process\_all\_deaths**

**Новая сигнатура:**

```
python
def process_all_deaths(
    state: SimulationState,
    day_in_year: int,
    base_mortality: float = MORTALITY,
    disease_mortality_factor_stage1: float = DISEASE_MORTALITY_FACTOR_STAGE1, # 0.02
    disease_mortality_factor_stage2: float = DISEASE_MORTALITY_FACTOR_STAGE2, # 0.10
    disease_mortality_factor_stage3: float = DISEASE_MORTALITY_FACTOR_STAGE3, # 0.20
    dispersal_deadline: int = DISPERSAL_DEADLINE,
    maturity_age: int = AGE_JUVENILE_TO_ADULT,
    semelparous_death_day: int = 10,
    device: th.device = DEVICE
) -> None:
```

#### **РАЗДЕЛЬНАЯ ОБРАБОТКА СТАДИЙ:**

##### **1. Стадия 1 (латентная):**

```
python
# Используем stage1_duration в формуле
disease_death_rate = disease_mortality_factor_stage1 / (1.0 + th.exp(sigmoid_arg))

# Максимум: 2% в день
```

##### **2. Стадия 2 (инфекционная):**

```
python
# Используем stage2_duration (исправлено!)
age_in_stage2 = age_of_disease - stage1_duration # Возраст в текущей стадии
disease_death_rate = disease_mortality_factor_stage2 / (1.0 + th.exp(sigmoid_arg))

# Максимум: 10% в день
```

##### **3. Стадия 3 (терминальная):**

```
python
```

```
# Две компоненты:  
# 1. Ежедневная вероятность (до 20%)  
disease_death_rate = disease_mortality_factor_stage3 / (1.0 + th.exp(sigmoid_arg))  
  
# 2. 100% смертность в конце стадии  
total_duration = stage1_duration + stage2_duration + stage3_duration  
  
death_at_end = age_of_disease >= total_duration # Гарантированная смерть
```

## ФОРМУЛА СИГМОИДЫ (одинаковая для всех стадий):

```
text
```

```
sigmoid_arg = 10.0 * ((duration/2.0) + (duration/3.0) - age_in_stage) / duration  
  
disease_death_rate = StageFactor / (1.0 + exp(sigmoid_arg))
```

Где:

- duration = длительность текущей стадии
  - age\_in\_stage = возраст в текущей стадии
  - StageFactor = DISEASE\_MORTALITY\_FACTOR\_STAGE\* (0.02, 0.10, 0.20)
- 

## 4. ФАЙЛ: state.py

### НОВЫЕ ПОЛЯ В SimulationState:

```
python
```

```
# Индивидуальные длительности стадий (ранее были глобальные)  
stage1_duration: Tensor # Длительность латентной стадии (60-120 дней)  
stage2_duration: Tensor # Длительность инфекционной стадии (30-120 дней)  
  
stage3_duration: Tensor # Длительность терминальной стадии (60-120 дней)
```

### ОБНОВЛЕННЫЕ ИМЕНА ПОЛЕЙ:

```
python
```

```
stage1_duration: Tensor # Для согласованности с новой терминологией  
stage2_duration: Tensor  
  
stage3_duration: Tensor # Новое поле
```

---

## 5. ФАЙЛ: main.py

### ОБНОВЛЕННЫЕ ВЫЗОВЫ:

#### Infection spread:

```
python  
  
infection_spread(  
    simulation_state,  
    distance_sq=distance_sq,  
    day_in_year=day_in_year,  
    infectivity1=INFECTIVITY1,      # 0.3 (параметр)  
    infectivity2=INFECTIVITY2,      # 0.01 (параметр)  
    stage1_multiplier=STAGE1_TRANSMISSION_MULTIPLIER, # 0.1  
    stage2_multiplier=STAGE2_TRANSMISSION_MULTIPLIER, # 5.0  
    stage3_multiplier=STAGE3_TRANSMISSION_MULTIPLIER, # 10.0  
    breeding_days=BREEDING_DAYS,  
    device=DEVICE  
  
)
```

#### Death processing:

```
python  
  
process_all_deaths(  
    state=simulation_state,  
    day_in_year=day_in_year,  
    base_mortality=MORTALITY,  
    disease_mortality_factor_stage1=DISEASE_MORTALITY_FACTOR_STAGE1, # 0.02  
    disease_mortality_factor_stage2=DISEASE_MORTALITY_FACTOR_STAGE2, # 0.10  
    disease_mortality_factor_stage3=DISEASE_MORTALITY_FACTOR_STAGE3, # 0.20  
    dispersal_deadline=DISPERSAL_DEADLINE,  
    maturity_age=AGE_JUVENILE_TO_ADULT,  
    semelparous_death_day=SEMELPAROUS_DEATH_DAY,  
    device=DEVICE  
  
)
```

---

## 6. ФАЙЛЫ БЕЗ ИЗМЕНЕНИЙ:

### genetics.py:

- Без изменений - логика наследования хромосом осталась прежней
- Функции `calculate_chrom` и `replication` работают как и раньше

### physics.py:

- Без изменений - физика движения, отталкивания, притяжения к территории
- Функции `move`, `calculate_full_distance_matrix` и т.д. не изменились

### visualisation.py:

- Временные изменения вернулись к оригиналу
  - Масштаб и легенда восстановлены как было
- 

## 7. НОВАЯ МАТЕМАТИЧЕСКАЯ МОДЕЛЬ В ЦЕЛОМ:

### ИНФЕКЦИОННОСТЬ:

text

$$P(\text{заражения}) = \text{StageMultiplier} \times \text{DistanceFactor} \times (\text{SexualComponent} + \text{NonsexualComponent})$$

Где:

- `StageMultiplier`  $\in \{0.1, 5.0, 10.0\}$  (по стадии)
- `DistanceFactor`  $= (\text{MAX_RADIUS}^2 - \text{distance}^2) / \text{MAX_RADIUS}^2 \in [0, 1]$
- `SexualComponent`  $= \text{infectivity1} \times [\text{day\_in\_year} < \text{breeding\_days}] \times [\text{оба взрослые}]$
- `NonsexualComponent`  $= \text{infectivity2} \times [\text{резиденты}]$

### СМЕРТНОСТЬ:

text

$$P(\text{смерти\_от\_болезни}) = \text{StageFactor} / (1 + \exp(10 \times ((d/2) + (d/3) - a) / d))$$

Где:

- StageFactor  $\in \{0.02, 0.10, 0.20\}$  (по стадии)

- d = длительность текущей стадии

- a = возраст в текущей стадии

+ Для стадии 3: 100% смертность в конце

---