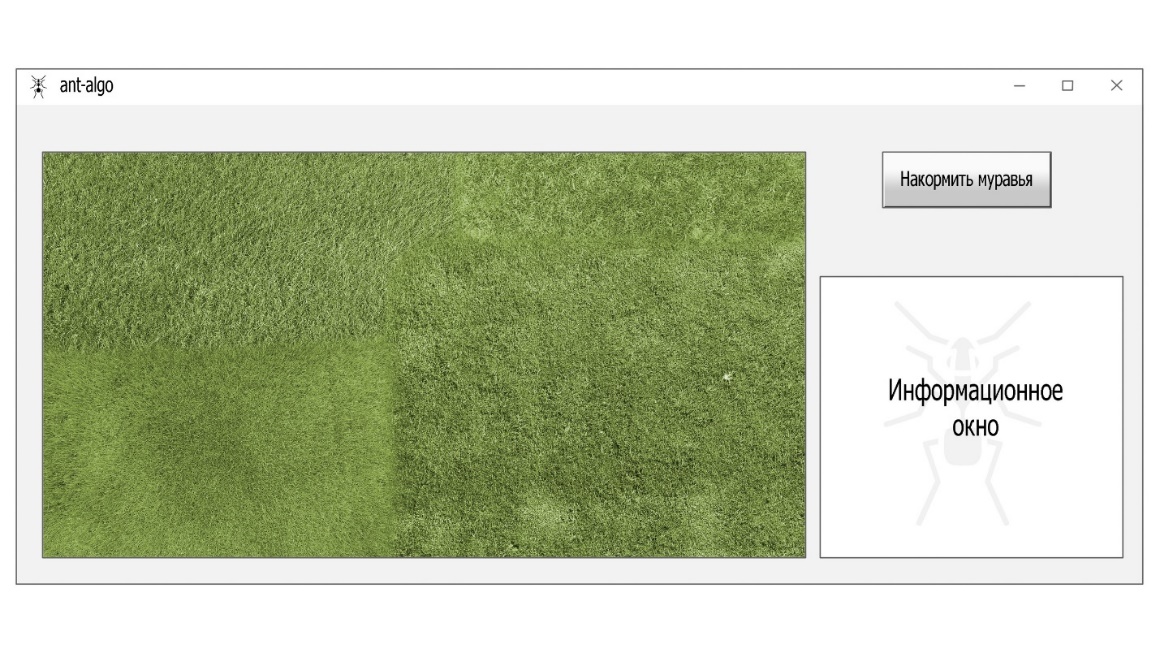
В пространстве имен core объявлено состояние (state) приложения, которое может быть:

1. state\_started – приложение только что запущенно
2. state\_nodes – состояние выбора положения узлов
3. state\_ants – состояние выбора количества муравьев
4. state\_execution – состояние выполнения алгоритма

Алгоритм работы:

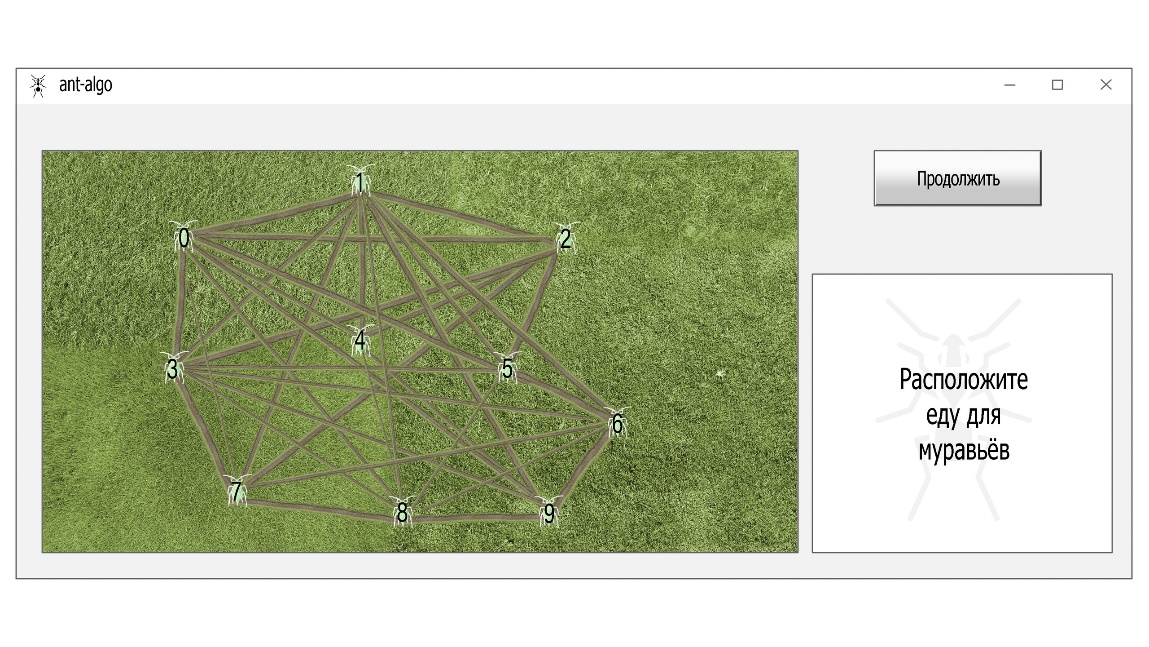
1. Отрисовывается окно с начальным состоянием - drawWindow(state\_started).
2. Запускается waitForEvent(), который ждет событие внутри окна.

Возможные события: закрыть окно (возвращает AppEvent::EXIT), нажатие на кнопку «Накормить муравья» (возвращает AppEvent::FEED)



1. Если функция вернула AppEvent::FEED, кнопка накормить муравья меняется на «Продолжить», происходит выход из текущего цикла, пользовательский интерфейс обновляется – drawWindow(state\_nodes), запускается новый цикл

который ожидает от функции waitForEvent() закрытия программы (AppEvent::EXIT), нажатия на поле – выбора положения узла (возвращает AppEvent::DRAWNODE) или нажатия на “продолжить” – переход к выбору кол-ва муравьев (возврат AppEvent::CHOOSEANTS)



1. Если waitForEvent() вернул AppEvent::DRAWNODE

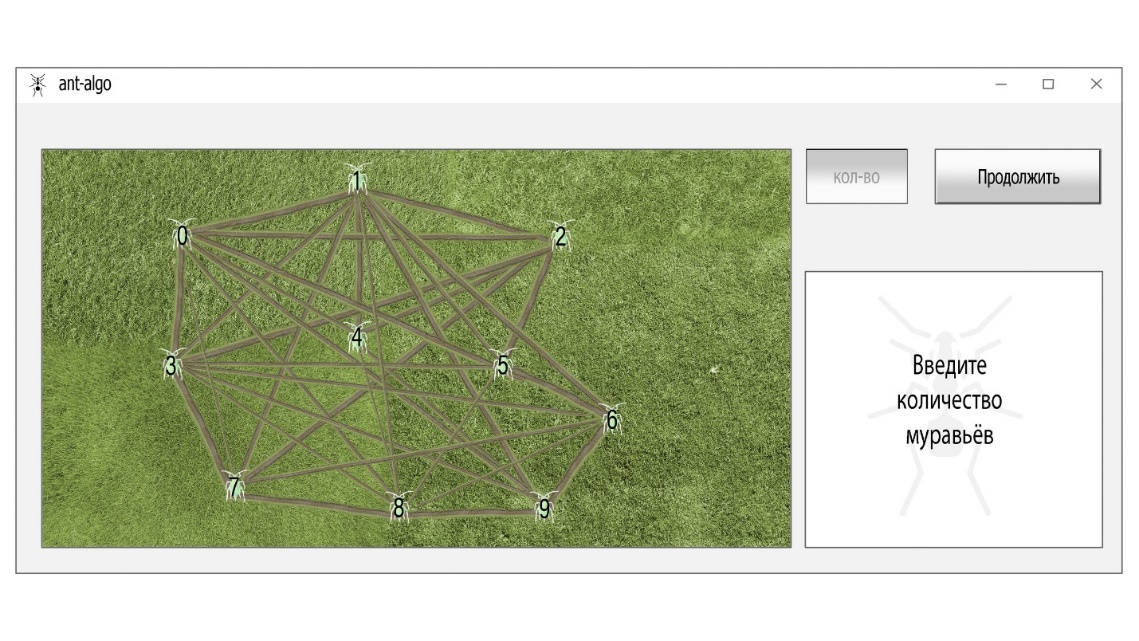
а) рисуется узел на поле

б) вызывается Visualization::drawPath(tempX, tempY, …) со всеми уже существующими узлами (в цикле вызвать со всеми элементами из core::nodes)

г) в nodes добавляется новый Node

1. Если waitForEvent() вернул AppEvent::CHOOSEANTS – выход из цикла.

Обновляется пользовательский интерфейс – drawWindow(state\_ants). Запускается новый цикл, в котором обрабатываются следующие события (waitForEvent()): AppEvent::EXIT, AppEvent::LAUNCH.



1. При нажатии на кнопку «Продолжить», внутри слушателя событий считывается введенное количество муравьев и кладется в core::numberOfAnts, а следом функцией возвращается событие AppEvent::LAUNCH (запуск алгоритма).
2. При событии AppEvent::LAUNCH происходит выход из текущего цикла, обновляется пользовательский интерфейс – drawWindow(state\_execution), начинается выполнение алгоритма…

