Задания в совокупности дают более чем 10 баллов. Обязательны к выполнению только 1, 2, 4, 5. Остальные можете пропускать, если хотите.

1. В *окне* отображается поле из (N=15)x(M=20) клеток. Каждая клетка может быть в одном из трёх основных состояний: богата ресурсами (зелёная), истощена (чёрная), содержит группу кроликов (зелёная с красным эллипсом внутри).
2. Каждые T=1000ms происходит шаг симуляции: текущая клетка кроликов становится истощённой, а сами кролики пытаются перейти на соседнюю свободную богатую ресурсами клетку (в порядке *приоритета*: верхнюю, правую, нижнюю либо левую)
3. \*\* Сделайте перемещение не по приоритету, а случайным среди доступных клеток.
4. Если кролики находятся на истощённой клетке, то они погибают (исчезают с поля).
5. Если все кролики погибли, симуляция считается завершённой. Выведите сообщение о завершении симуляции.
6. В углу окна выведите текст, показывающий номер текущего шага симуляции.
7. \* Используйте двойную буферизацию.
8. Окно содержит *меню*, через которое можно вызвать *диалог*.
9. В диалоге есть два поля (*текст-бокса*) для ввода и *кнопки* «OK» и «Отмена».
10. В поля вводятся два числа: I (0 <= I < N) и J (0 <= J < M). При нажатии OK в клетку с координатами (I, J) добавляется кролик и диалог закрывается. При нажатии на «Отмену» диалог закрывается.
11. \* Сделайте так, что симуляция замораживается, пока диалог открыт.
12. \* При нажатии на ОК проверяйте числа на корректность диапазона и при необходимости выводите соответствующее сообщение.
13. \* Кнопка OK должна быть доступна, только если в текст-боксы введены числа (а не буквы или пустая строка).
14. При нажатии левой кнопкой мыши идёт перерождение клетки под курсором из истощённой в богатую ресурсами.
15. При наведении курсора на истощённую клетку она должна подсвечиваться.
16. При наведении курсора на истощённую клетку он принимает вид «рука» (hand).
17. \* Сделайте так, что возможно восстановить не более одной клетки за время шага T.
18. Нажатие на клавишу [R] полностью сбрасывает симуляцию.