

Библиотека SFML. Graphical User Interface. Задачи.

- **Перетаскивание:** В папке `1movable/` содержится заготовка исходного кода для этого задания. Эта программа просто рисует прямоугольник на экране. Сделайте его перетаскиваемым мышью. При нажатии на него и последующим движении мыши он должен начать двигаться вместе с курсором. При отпускании мыши должен остаться на месте.
- **Карточки:** В этом же проекте создайте 20 прямоугольников случайного цвета, но одинакового размера, так чтобы все они были перетаскиваемыми.
- **Выбор и удаление:** В папке `2move_and_delete/` содержится заготовка исходного кода для этого задания. В этой программе есть несколько объектов(кругов), которые можно выделять. Выделение происходит по нажатию левой клавиши мыши. Зажав клавишу `Ctrl` можно выделить несколько объектов. Также в программе реализован прямоугольник выделения (но он пока не выбирает объекты). Нажатием правой кнопки мыши можно создать случайный круг. Добавьте следующие возможности в программу:

- Перемещение всех выделенных объектов при зажатии левой клавиши мыши и её движениях. Перемещаться должны все выделенные объекты параллельно (также как перемещаются несколько выделенных значков на рабочем столе). Прямоугольник выделения при этом рисоваться не должен.
- Выделение объектов с помощью прямоугольника выделения. Прямоугольник выделения должна рисоваться только если нажатие мыши произошло вне кругов. Все объекты, находящиеся внутри прямоугольника выделения, на момент отпускания левой кнопки мыши должны выделяться.
- При нажатии клавиши `Delete`, все выделенные объекты должны удаляться. Чтобы удалить элемент из `std::list` используйте итераторы и метод `erase`. Чтобы вспомнить как это работает – пример работы с методом `erase` (удаляем все отрицательные числа):

```
std::list<int> numbers = {5, -4, 6, 41, 64, -10, 16};
for (std::list<int>::iterator it = numbers.begin(); it != numbers.end(); it++)
{
    if (*it < 0)
        it = numbers.erase(it);
    // erase возвращает правильный итератор
    // Если удалять так numbers.erase(it) - цикл не сможет выполнить it++
}
```

- Задание случайного цвета. При нажатии клавиши пробел цвет всех выделенных шаров должен меняться на случайный. Для этого понадобится добавить поле `color` в класс `Ball`.
- **Класс кнопки:** В папке `3button/` содержится заготовка исходного кода для этого задания. Логика работы должна этой кнопки аналогичной логике работы обычной кнопки в ОС Windows. При нажатии на прямоугольник он немного меняет цвет. При отпускании мыши, если курсор всё ещё находится на прямоугольнике, срабатывает некоторое действие (печать в консоль).
 - Создайте 1 круг. Сделайте так, чтобы при нажатии на кнопку цвет круга менялся бы на случайный.
 - **Переключатель** - в этом же файле создайте класс под названием `ToggleButton`, который будет описывать поведение переключателя.



Переключатель может находиться в 2-х состояниях (ON/OFF). При нажатии мышью на переключатель он должен менять состояние. Графическое оформление – на ваше усмотрение.

- Создайте 1 экземпляр переключателя. Сделайте так, чтобы круг перекрашивался в зелёный цвет, если переключатель находится во включённом состоянии и так, чтобы круг перекрашивался в красный цвет, если переключатель выключен.

- **Флажки:** В папке `4checkbox/` содержится заготовка исходного кода для этого задания. Описан класс `Checkbox` - флажка. Измените программу так, чтобы при нажатии на кнопку на экран печатались все города, у которых выделены флажки.
- **Контекстное меню:** В папке `5context_menu/` содержится пример, реализующий простейшее контекстное меню на SFML.
 - Создайте круг. И добавьте опции в контекстное меню так, чтобы можно было менять цвет и размер круга.
 - Добавьте к задаче **Выбор и удаление** контекстное меню. С его помощью нужно создавать объекты, удалять их, изменять их цвет и радиус.