

На основе материалов лабораторных работ № 3 (преимущественно) и 2 реализовать ядро гоночной игры. В рамках гонки определенный (введенный) пользователем набор транспортных средств (ТС) должен проехать заданное количество “кругов” определенной длины. Допускается считывать характеристики ТС и параметры гонки из файла входных данных.

1. Внутри ядра организовать глобальный цикл продолжительностью до финиша последнего не сошедшего с дистанции ТС с шагом по времени 1 сек.
2. На каждой итерации цикла для каждого не сошедшего с дистанции ТС вычислять:
 - скорость на текущей итерации. Считать, что скорость набирается ТС моментально;
 - новое положение на треке в момент окончания итерации цикла;
 - суммарный пробег в рамках текущей гонки в момент окончания итерации цикла;
 - остаток топлива в баке в момент окончания итерации цикла;
 - пробег колеса в момент окончания итерации цикла;
 - вероятность повреждения каждого колеса. Если какое-либо колесо меняет свой статус на “повреждено”, то новая текущая скорость ТС принимается равной 75% от предыдущей. При повреждении большего количества колес процедура повторяется. Считать, что повреждение происходит в момент начала следующей итерации цикла. Повреждения колес - независимые события.
3. Общая логика прохождения дистанции ТС:
 - с момента старта проходит дистанцию круг за кругом;
 - в конце каждого круга проверяет необходимость заезда в бокс для дозаправки или смены колеса (колес). Условия заезда в бокс придумать самостоятельно. Время заправки зависит от объема бака и текущего остатка топлива. Считать, что все команды заливают топливо с одинаковой скоростью, которую необходимо вынести в константу. Аналогично, скорость замены колеса - отдельная константа, идентичная для всех ТС. Время проведенное в боксе прибавляется ко времени прохождения

трассы ТС. Заправка и замена колос производится одновременно;

- если у ТС произошло повреждение колеса на круге, оно обязано заехать в бокс для замены колеса в конце текущего круга;
- условия схождения с дистанции: повреждение всех колес при прохождении дистанции или нулевой остаток топлива в баке.

4. Дополнительные требования:

- стандартные требования к меню и вводу информации;
- хранение объектов классов осуществлять в контейнерах типа `vector`;
- при работе со строками использовать класс `string`;
- для ввода\вывода информации по объектам перегрузить соответствующие операторы;
- предусмотреть обработку исключений в наиболее очевидных местах;

Дата сдачи - 17 неделя.