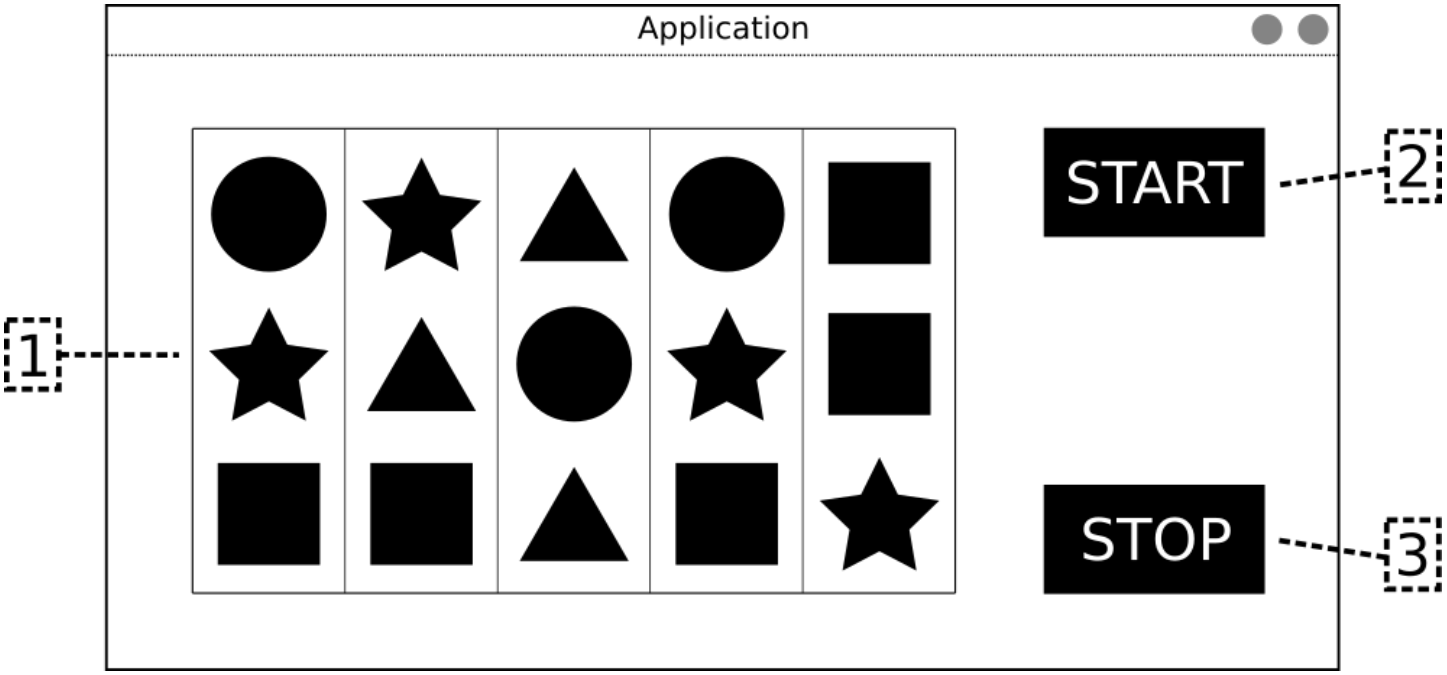
Тестовое задание на позицию
Junior C++ Game Developer

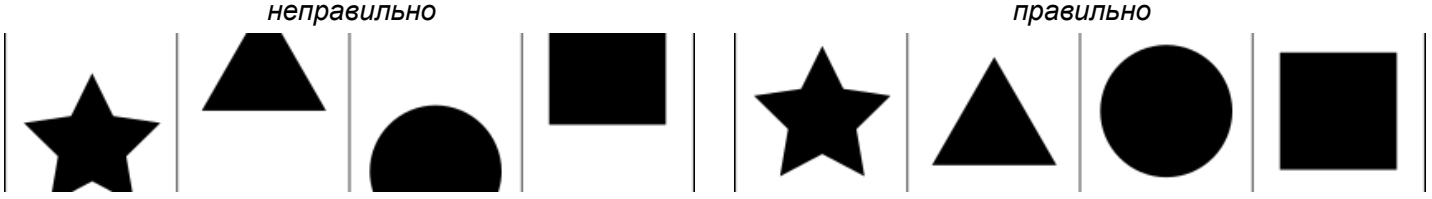
Для выполнения этого задания Вам необходимо создать приложение, которое будет эмулировать вращающиеся барабаны слот-машины. Приложение должно содержать 3 обязательных графических элемента, показанных на схематичном изображении:



1) барабаны слот-машины;
2) кнопка **Start** для запуска вращения барабанов;
3) кнопка **Stop** для остановки вращения барабанов.

Кроме обязательных элементов приложение должно также иметь:

1. Автоматическую остановку барабанов по таймеру, если кнопка **Stop** не была нажата.
2. Плавную смену состояний барабанов, то есть барабаны не должны резко начинать и заканчивать вращение. Также стоит отметить, что мы не требуем сверхточную симуляцию вращения на основе реальных физических законов.
3. Остановку барабанов только в таких позициях, в которых все символы на барабанах находятся на одной линии, более наглядно это показано на рисунке ниже:



4. Как минимум 3 барабана, каждый из которых должен иметь как минимум по 3 символа.
5. Реализация смены режимов игры должна быть основана на машине состояний (state machine).

Для обязательных графических элементов Вы вольны использовать любые графические стили, менять их расположение, а также включать или выключать отображение кнопок при необходимости. Кроме того, Вы можете добавлять дополнительные графические элементы, эффекты и другие улучшения внешнего вида приложения.

После остановки барабанов приложение должно осуществить подсчёт выигрыша по любой известной вам логике. Например, он может быть основан на подсчёте количества одинаковых символов в одной линии.

Необходимо без использования сторонних библиотек реализовать циклическую машину состояний, которая должна:

- содержать в себе как минимум три состояния (state):
  - ожидание действия игрока, в котором приложение ожидает нажатия на кнопку **Start**;
  - показ выигрыша, в котором необходимо продемонстрировать расчитанный выигрыш;
  - вращение барабанов, это состояние активно пока другие два неактивны, при желании можно разбить это состояние на более мелкие.
- переходить из одного состояния в другое согласно данной последовательности:
  - ожидание действия игрока;
  - вращение барабанов;
  - показ выигрыша;
  - ожидание действия игрока (новый цикл);
  - и так далее…
- использовать полиморфизм в реализации, то есть классы всех состояний должны иметь общий интерфейс.

При необходимости вы вольны добавлять новые состояния в эту машину, также как и разбивать описанные выше состояния, меняя тем самым описанную последовательность. Данную машину состояний необходимо оформить в виде отдельного класса в составе проекта, также в виде отдельных классов стоит оформить и состояния машины состояний. Как это уже было сказано ранее, необходимо использовать данную машину состояний в реализованном приложении.

Для реализации приложения Вы должны использовать язык C++ в стандарте не выше C++17. Разрешается использовать стандартную библиотеку языка, а также любые другие open-source библиотеки. В случае использования сторонних библиотек следует убедиться в том, что они либо встраиваются в приложение, либо они предоставлены вместе с приложением, и приложение линкуется с ними. Не следует использовать платформо- и компиляторо-зависимые расширения языка.

В качестве системы сборки Вы можете использовать:

- для реализации на Linux: make, ninja, а также все системы, способные их сгенерировать;
- для реализации на Windows: nmake, проект для Visual Studio, а также все системы, способные их сгенерировать.

Стоит отметить, что Вы практически не ограничены в реализации графического представления: для этого можно использовать как низкоуровневые библиотеки (например OpenGL), так и более высокоуровневые библиотеки и фреймворки (например SFML, SDL). Тем не менее, использовать полноценные игровые движки не следует, так как они имеют намного большую функциональность, чем требуется для решения этой задачи.

Графические ресурсы, которые будут использованы в приложении, никак не регламентированы. Но все ресурсы, которые необходимы для работы приложения, должны быть также предоставлены вместе с приложением.

По итогу выполнения этого задания мы надеемся получить результат в одном из двух видов, который более удобен для Вас:

- архив, содержащий итоговый бинарный файл приложения (и библиотек, если имеются), а также исходный код этого приложения;
- ссылку на определённый релиз в веб-сервисе для хранения репозиториев (GitHub, GitLab, Bitbucket), в описании которого будет добавлен итоговый бинарный файл приложения, естественно, в этом случае исходный код должен быть открыт для просмотра.

Для обоих способов необходимо наличие файла для системы сборки, кроме того, необходимо также добавить файл-описание того, как производить сборку. Также мы настоятельно советуем убедиться в том, что после описанной процедуры сборки полученное приложение способно полностью функционировать.

Рекомендуемое время выполнения данного тестового задания - 1 неделя, но торопиться отсылать решение на проверку не стоит, так как качество кода оценивается в первую очередь. При возникновении любых непредвиденных трудностей в процессе решения Вы можете сообщить нам о них по электронной почте info.spb@octaviangaming.com.