

`target_vector, 2f)` поворачивает ракету в направлении цели, `game Object.transform.Translate (Vector3.up*0,050f)` перемещает ракету в направлении астероида.

Создание собственной игры – это исключительный опыт, который помог нам освоить азы движка Unity и дал нам толчок для развития в дальнейшем.

УДК 004.588

Студ. В.А. Белашков

Науч. рук. ст. преп. А. С. Наркевич
(кафедра программной инженерии, БГТУ)

САМОУЧИТЕЛЬ АНГЛИЙСКОГО ЯЗЫКА

Программа является консольным приложением, написанным на языке программирования C++. Проект состоит из 7 файлов: *main.cpp*, *stdafx.h*, *Leaing.cpp*, *Text.cpp*, *Functions.cpp*, *Dictionary.h*, *Dictionary.cpp*.

Файл *main.cpp* содержит главную функцию, которая вызывает остальные функции. Заголовочный файл проекта *stdafx.h* содержит заголовки используемых библиотек, прототипы функций, определение структур (*Settings* – настройки для изучения, *Grammar* – массивы со словами для изучения грамматики) и глобальных переменных.

Learning.cpp содержит функции для блока изучения: *learningWord Call* (нужна для вызова функции *learningWord* и передачи в нее параметров изучения), *learning Word* (изучение слов), *learning Gram* (изучение грамматики), *correctEnding* (т.к. слова для изучения грамматики используются из словаря, их нужно привести к нужной форме или числу, что и делает эта функция).

Файл *Text.cpp* (файл с информационными функциями) содержит все функции *menu* и функцию *grammarAid*. *Functions.cpp* (файл со вспомогательными функциями): *start* (считывание информации из файлов (словарь и настройки) и заполнение ей соответствующих структур), *end* (заполнение файлов (словарь и настройки) из соответствующих структур), *comparisonStr* (сравнение двух строк на равенство, без учета регистра).

Словарь реализован в виде пространства имен (*Dictionary*) и состоит из двух файлов. Первый *Dictionary.h* содержит определение структуры словаря и прототипы функций *add Word* (добавление слова в словарь), *del Word* (удаление слова из словаря), *check Dictionary* (просмотр словаря). *Dictionary.cpp* содержит определение функций, объявленных в *Dictionary.h*.

Принцип работы: при запуске приложения вызывается функция *start*, запускающая бесконечный цикл, в котором выводится текст меню и с помощью конструкции *switch* выбирается один из основных блоков программы (1 - словарь, 2 - грамматика, 3 - проверка знаний, 4 - настройки), так же из пункта меню для завершения работы.

Первый пункт содержит четыре подпункта: добавить слово в словарь (*add Word*), удалить слово (*del Word*), изменить слово (с помощью *add Word*) и просмотреть словарь (*check Dictionary*).

Второй пункт состоит из списка грамматических правил (*grammar Aid*), при выборе какого-либо пункта на экране отобразится соответствующее грамматическое правило.

Третий пункт содержит два подпункта: словарный запас и грамматика. Изучение слов происходит следующим образом (*learning Word Call, learning Word, comparison Str, словарь*): случайным образом из словаря выбирается слово и отображается на экране.

Пользователю нужно ввести перевод слова (язык перевода зависит от настроек пользователя). При изучении грамматики (*learning Gram, correct Ending, comparison Str, словарь*) изначально нужно выбрать грамматическое правило, которое хотите изучить.

Далее на экране появляются предложения на русском языке, состоящие из слов, которые пользователь добавил в словарь. Пользователю нужно ввести перевод предложения, учитывая изучаемое время и числа существительного (то есть предложение должно быть грамматически верным на английском языке).

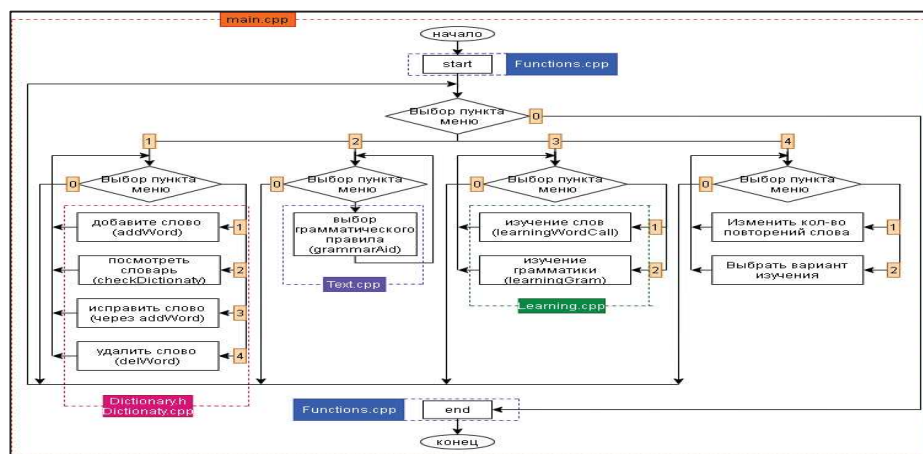


Рисунок 1 – Схема приложения

Последний блок приложения – настройки. В нем два подпункта: изменение количества повторений слова для вычеркивания его из списка изучаемых и выбор варианта изучения для словаря (ввод английского перевода, ввод русского перевода или комбинированный вариант). Перед завершением работы вызывается функция *end*. На рисунке представлена схема приложения.