

Программное обеспечение для проектирования объектов и программирования их поведения в задачах компьютерной графики

Разоренов Н.А., Фролов О.М.

Белорусский национальный технический университет

Современное программное обеспечение неразрывно связано с графическими интерфейсами. Так же существует отдельная категория программ, которая работает только с графикой (например, игры). Для разработки существуют разнообразные наборы библиотек, которые снимают с разработчика необходимость разрабатывать все с нуля. Эти наборы библиотек называются графическими движками.

Целями любого графического движка являются: 1) уменьшить время создания приложения; 2) легкая расширяемость базового функционала на основе существующей платформы.

Целью работы была разработка части функционала графического движка, для уменьшения дублирования кода. Так же легкого создания и расширения функциональной части объектов, которые используются для представления базовых примитивов. Для добавления расширяемости объектов и компоновки их в реальном времени, в зависимости от различных условий, были использованы возможности .net для создания динамических объектов. Для контролирования динамического поведения, объект должен реализовывать интерфейс IDynamic Meta Object Provider и возвращать реализацию объекта, унаследованного от Dynamic Meta Object. Это позволяет расширять функциональную составляющую объектов в реальном времени.

Для создания возможности динамически расширять функционал объектов были использованы Expressions, которые позволяют удобным образом создавать динамические методы и компилировать их в реальном времени.

Объединяя эти возможности, можно спрятать от пользователя внутреннюю реализацию системы, и упростить работу. Одной из главных идей, которые преследовались во время написания проекта, была реализация функциональной части подписываний на вызов метода. Следует заметить, что функционал подписки позволяет так же удобно распараллелить процесс работы приложения в целом. Для этого использовалась библиотека TPL(taskparallel library). Вызов подписанных методов осуществляется после завершения работы основного метода. Для реализации этого функционала использовался AOP(aspectoriented programming) подход. В частности был создан постобработчик на библиотеке Microsoft.Ccil для изменения созданных сборок, после из компиляции.