БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики и информатики

Кафедра технологий программирования

Трофимчик Анастасия Дмитриевна

**РЕФЕРАТ**

**НА ТЕМУ “С/С++. Возможности создания игровых приложений”**

Студентки 1 курса 8 группы

**Преподаватель**

*Солодовникова*

*Марина Леонидовна*

ст. Преподаватель кафедры ТП

Минск, 2019

**1.Игровые приложения**

Ультрасовременные компьютерные игры не уступают голливудским блокбастерам по визуальным эффектам , саундтреку и градусу напряжения . Но игры совершенно особый вид развлечений; хорошая игра может заставить геймера часами просиживать у монитора . Основная изюминка компьютерных игр , которая делает их столь увлекательными, — это интерактивность. Компьютерная игра - это не фильм, где вы наблюдаете за приключениями героя , побеждающего несмотря ни на что, - в игре вы сами перевоплощаетесь в героя.

Такая интерактивность достигается прежде всего с помощью программирования . Именно программа оживляет монстра , эскадрилью штурмовиков или целую армию , заставляет эти виртуальные сущности по-разному реагировать на действия пользователя в зависимости от ситуации . Благодаря программе игровой сюжет может принимать новые обороты . На самом деле именно на уровне программной логики игра может отвечать на действия пользователя так, что удивляет даже собственных создателей.

Хотя в настоящее время существует огромное количество языков программирования, важнейшим из них в игровой индустрии является язык C ++ . Если в ближайшем универмаге вы зайдете в отдел компьютерных игр и наугад возьмете диск с полки вероятно что вся игра от первой до последней строчки будет написана на С ++ . Мораль : если хотите профессионально заниматься программированием игр , выучите C ++ .

**2.С++ - мощный инструмент создания приложений**

С++ — это объектно-ориентированный язык программирования, который сочетает в себе возможности низкоуровневого и высокоуровневого языков, иногда его даже называют из-за этого языком среднего уровня. Если проанализировать долю использования объектно-ориентированных языков, то согласно тому же (приведенному выше) ресурсу, она превышает 50-процентный рубеж, обгоняя функциональные, логические и процедурные языки.

С++ — это мощный инструмент для создания различных приложений, от небольших до достаточно объемных и трудоемких. Главным преимуществом этого языка является сочетание в нем возможности работы на низком и высоком уровнях. Таким образом, код создаваемый программистом может быть максимально адаптирован к системным возможностям той платформы, на которой он будет работать. А именно это свойство дает неоспоримое преимущество в оптимальном использовании ресурсов вычислительной системы (компьютера или другого устройства, например, мобильного телефона или игровой приставки).

С++ - это язык программирования общего назначения, хорошо известный своей эффективностью, экономичностью, и переносимостью. Указанные преимущества С++ обеспечивают хорошее качество разработки почти любого вида программного продукта. Использование С++ в качестве инструментального языка позволяет получать быстрые и компактные программы. Во многих случаях программы, написанные на С++, сравнимы по скорости с программами, написанными на языке ассемблера.

C++ продолжает оставаться наиболее распространенным языком для разработки игр, и на то есть свои причины. До сих пор большинство заказных проектов пишется на нем.  
C++ покрывает нужды, но не соответствует желаниям. Он сложный. Несмотря на наличие хороших инструментов, на нём легко допускать ошибки. Кроме того, по сравнению с C, он медленно компилируется. C++ имеет высокую производительность и предоставляет возможности, которых нет в C.  
C# и Java имеют схожие проблемы. Они многословны и сложны, а C++ прост. Оба языка направляют программиста прямиком в пучину ООП. Как и в большинстве высокоуровневых языков, многие сложные вещи тут скрываются так, что ничего не мешает случайно выстрелить себе в ногу.

**2.Почему C — лучший выбор**  
Хоть C и опасен, зато надёжен. Это очень острый нож, способный отрубить пальцы так же легко, как нарезать овощи. Но он прост, и научиться правильно его использовать труда не составит.   
  
Он быстрый. А когда дело доходит до компиляции, сложно представить, что какой-то язык способен сделать это быстрее.   
Есть возможность писать код так, чтобы он работал везде. И обычно это относительно несложно. Трудно представить, что однажды это изменится.  
Есть отличная поддержка библиотек и инструментов.  
Хоть это и несколько печалит, но C до сих пор остаётся лучшим языком.  
Так что да, C — лучший выбор.

**3. Общая схема создания компьютерных игр**

Для создания любой компьютерной видеоигры сегодня используются игровые движки. Игровой движок – базовое программное обеспечение компьютерной игры. В общем случае, термин «игровой движок» применяется для того программного обеспечения, которое пригодно для

повторного использования, и тем самым может быть рассмотрено как основание для разработки множества различных игр без существенных изменений. Стоит заметить, что игровых движков сегодня довольно много. Некоторые из них полностью закрыты и используются только внутри компаний. Некоторые открыты вплоть до исходных кодов программ.

Большая часть игровых движков написана на языке С++: ORX, GoldSrc, Blender, Panda3D, Mar-malade, Anura, WorldForge, Visual Pinball, Unreal Engine, Unity, Unigine, Spring, Source, OGRE Horde3D, Dark Engine и т. д.

То, что игровые движки написаны на С++, заставляет гейм-программистов писать свои игры на этом же языке.

С помощью игровых движков создается оболочка компьютерной видеоигры: игровые карты, сцены, анимация движений персонажей и т. д. В общем, движки отвечают за реализацию механики, графики, анимации. Для реализации физических явлений используются физические движки. Например, NVIDIA PhysX, который, в свою очередь, тоже написан на С++. При создании серьезной видеоигры придется работать с освещением, тенями, частицами, разрушаемостью — всем, что важно для конкретной игры. Один человек не может разработать все это за год или даже два, поэтому такое под силу только крупным студиям. Конечно, многие из них трудятся над проектами в движках, но данная работа все равно колоссальна.

Однако без движков тоже можно обойтись: например, World of Warcraft был написан исклю-чительно на C++. В общем, при разработке компьютерной игры работа с игровым движком – лишь подготовительный этап. После того как в редакторе была создана готовая сцена, приходит время написания кода. Код приводит в действие созданные в игровом движке объекты. Напри-мер, движок отвечает за создание анимации для какого-то объекта, а вот код нужен для того, чтобы эта анимация появлялась или исчезала в определенных условиях игры.

**4.Роль языков С/С++ в разработке игр**

Для разработки игры можно выбрать любой язык. Но ориентироваться лучше на то, что создателю игры хочется реализовать и на какой платформе:

Swift – игры на iOS или MacOS

PHP и JavaScript – браузерные игры

C# – игры на Unity

С или C++ – большие требовательные игры

В большинстве случаев используются языки, позволяющие работать с большой нагрузкой – а это языки С или С++.

Причем некоторые программисты используют процедурный язык С так же успешно, как и объектно-ориентированный язык С++.

Дело в том, что большое количество программистов считают С++ слишком сложным и из-за этого в нем легко допускать неприятные ошибки. Кроме того, по сравнению с С, он медленно компилируется. С++ предоставляет возможности, которых нет в С, но достигается это ценой большой сложности.

Однако С++ при разработке больших и серьезных компьютерных игр используется более широко, чем С. Дело в том, что первым достоинством объектно-ориентированного программирования является то, что оно обеспечивает значительное сходство между объектами реального мира, моделируемыми программой, и объектами классов С++. Объекты классов могут представлять такие реально существующие вещи, как игральные карты, геометрические фигуры, человеческие персонажи и т. д. Это облегчает концептуализацию решаемой задачи. Программист выделяет в задаче те элементы, которые целесообразно представлять в виде объектов, а затем помещает все данные и функции, связанные с этим объектом, внутрь самого объекта. Если объект представляет игральную карту, то с помощью описания класса программист наделяет объект полями для хранения достоинства карты, ее масти, а также функциями, выполняющими действия по заданию этих величин, их изменению, выводу на экран, сравнению с заданными значениями и т. д.

В процедурном С, напротив, моделирование объектов реального мира возлагается на глобальные переменные и функции. Они не способны вместе образовать легко воспринимаемую модель, что является чрезвычайно важным для эффективного создания компьютерных игр.

Хотя в настоящее время существует огромное количество языков программирования, важнейшим из них в игровой индустрии является язык С++. Существует ряд причин, по которым разработчики игр активно пользуются языком С++:

1) Он быстр. Грамотно написанные программы на С++ могут работать просто молниеносно. Одной из основных проектных характеристик языка С++ была высокая производительность. Если в программе требуется добиться очень высокой производительности, то С++ позволяет работать и с ассемблером – самым низкоуровневым языком программирования, который способен про-честь программист.

2) Он гибок. С++ — это мультипарадигмальный язык, поддерживающий различные стили программирования, в том числе объектно-ориентированное программирование. Но, в отличие от многих современных языков программирования, С++ не имеет жесткой привязки к какой-либо парадигме программирования.

3) Он хорошо поддерживается. Поскольку язык С++ уже очень давно используется в игровой индустрии, по нему доступно множество ресурсов. Это и графические API, и возможности 2D и 3D, и игровая физика, и звуковые движки. Программист, работающий с С++, может использовать готовый код, значительно ускоряя разработку новых игр.

На протяжении многих лет С++ был единственным выбором для высокопроизводительных приложений любого типа, включая игры. В результате были сделаны значительные инвестиции в игровые движки и библиотеки, которые также были написаны на С++ и даже сегодня большинство фреймворков для игр написаны на С++.

По мере того как машины становятся более мощными, а большая часть рабочей нагрузки сдвигается на аппаратное обеспечение (графика и даже физика, в некоторых случаях), для выполнения других задач доступно больше системных ресурсов. Таким образом, несмотря на то, что С++ по-прежнему является отраслевым стандартом, оптимизация "вниз к металлу", которую он позволяет, больше не является основной причиной его использования. Основная причина, по которой он используется, заключается в том, что большинство API для разработки игр - это С++ API.

Однако имеет место тенденция, которую стоит отметить, заключающаяся в том, что в настоящее время многие игры используют языки сценариев, такие как Python или Lua, для управления большей частью логики игры. Это дает разработчикам контента гибкость для обеспечения богатого поведения без необходимости кода на С++ или сборки. Сценарий не является высокопроизводительным, но он чрезвычайно удобен, и современные ПК имеют достаточную мощность для обработки сценариев.

![](data:application/pdf;base64,)

![](data:application/pdf;base64,)

![](data:application/pdf;base64,)

![](data:application/pdf;base64,)

![](data:application/pdf;base64,)

![](data:application/pdf;base64,)

![](data:application/pdf;base64,)

![](data:application/pdf;base64,)