Software Requirements Specification

For

CHDBS Engine

1. **Introduction**
2. Purpose

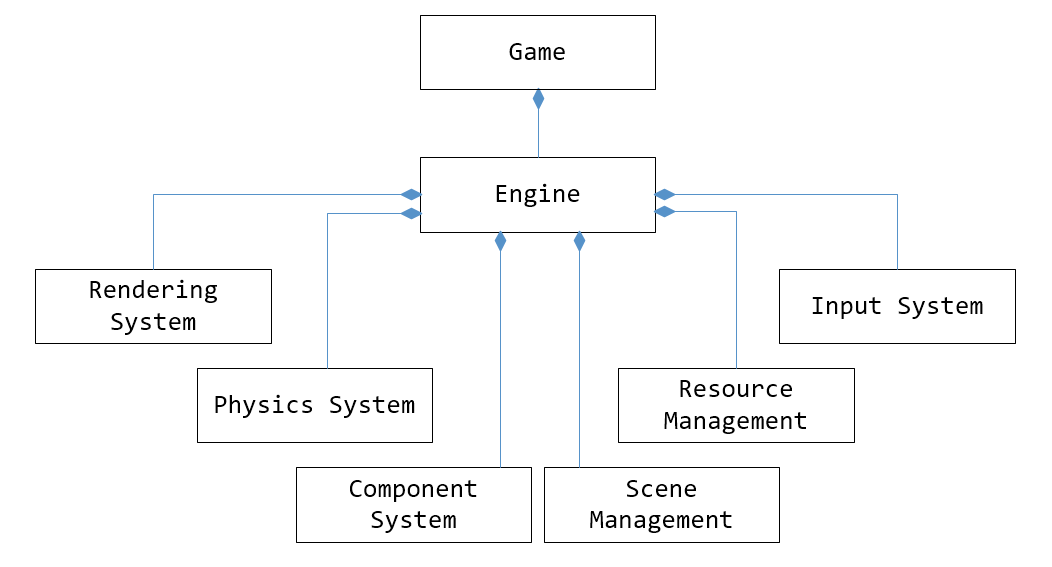
Система предназначена для создания трехмерных видеоигр.

1. Scope

“CHDBS Engine” – это игровой движок, который позволяет разработчикам создавать трёхмерные игры с помощью языка программирования C++.

1. Product overview
   1. *Product perspective*

Цель игрового движка – возможность создания различных видеоигр без существенного изменения основных систем: отрисовка, симуляция физики и т.д.



* 1. *Product functions*

CHDBS Engine предоставляет возможность отрисовки 3D объектов, простейшую симуляцию физики, базовую систему скриптинга.

* 1. *User characteristics*

Пользователями системы являются разработчики видеоигр. Пользователи должны владеть базовыми знаниями C++ и иметь понимание работы системы. Зона ответственности пользователей - конфигурация системы, загрузка ресурсов и создание скриптов.

* 1. *Limitations*

Система не обеспечивает вывод звука, эффективное управление памятью, взаимодействия с сетью. Игровой движок не имеет GUI.

1. Definitions

* Transform – класс трансформаций в трехмерном пространстве. Transform описывает позицию, поворот и масштаб.
* Entity – некоторый класс, содержащий уникальный идентификатор, класс Transform и массив Component.
* Component (компонент) – некоторый класс, прикрепляемый к Entity и определяющий поведение и функции этого Entity.
* Script (скрипт) – код определенного Component.
* Shader (шейдер) – программа, исполняемая с помощью GPU
* Texture (текстура) – изображение, используемое для отрисовки объектов
* Material (материал) – набор текстур и шейдера
* Skybox – специальный объект, используемый для рисования заднего фона
* Cubemap – специальная текстура, обычно используемая для отрисовки отражений. Является набором из 6 текстур
* Particle System (система частица) – специальная техника, позволяющая рисовать большое количество спрайтов или 3D моделей
* Rigidbody (rigid body) – представляет абсолютно твердое тело
* Resource / Asset – ресурс, используемый игровым движком. В CHDBS Engine есть несколько видов ресурсов:
* XML – содержит пары ключ-значение
* Model – содержит трехмерную модель
* Texture – содержит изображение
* Scene – содержит Entity, которые находятся в одном пространств

1. **Reference**

Mike McShaffry, David Graham (2013). *Game Coding Complete.* Boston, MA: Course Technology.

Joey de Vries (2017). *Learn OpenGL.* (n.p.).

Jason Gregory (2009). *Game Engine Architecture.* Boca Raton, FL: Taylor and Francis Group, LLC.

Christer Ericson (2005). *Real-Time Collision Detection.* San Francisco, CA: Elsevier Inc.

1. **Specific Requirements**
   1. External Interfaces
      1. *User Interface*
         1. *Создание Component.*

CHDBS Engine использует Component Based Architecture Pattern, следовательно, чтобы добавить функциональности некоторому Entity, пользователю нужно создать Component:

* создать класс, который должен наследоваться от интерфейса IComponent
* зарегистрировать класс в EntityFactory для возможности загрузки данных из XML файлов
* в заголовочном файле в описании класса добавить “CLASSDECLARATION(<название класса>)” и “CLASSDEFINITION(<название класса-родителя>, <название класса>)”

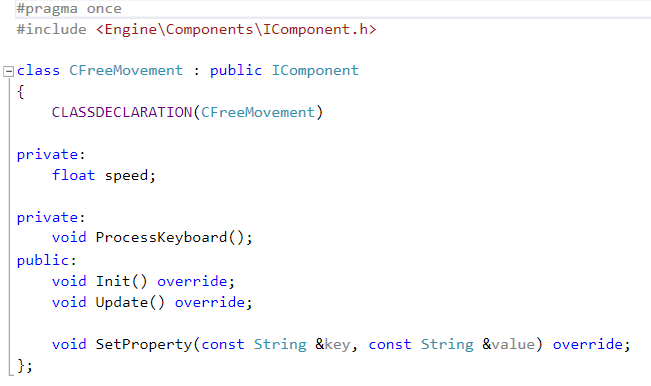


Figure 1 - пример описания Component



Figure 2 - пример описания Component

* + - 1. *Абстрактные функции класса IComponent.*

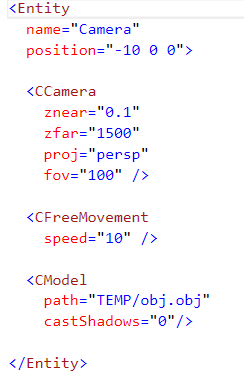


Figure 3 - пример XML файла для создания Entity "Camera"

IComponent содержит 3 абстрактные функции – Init(), Update() и SetProperty(String, String).

Функция SetProperty вызывается при создании Entity в EntityFactory во время парсинга XML файла, в котором хранятся пары ключ-значение. В *Figure 2* приводится пример реализации, при этом если в XML файле (*Figure 3*) есть пара с ключом *“speed”*, то полю *“speed”* в данном Component присваивается значение.

Функция Init вызывается после присваивания всех значений из XML файла.

Функция Update вызывается после каждой смены кадра.

* + - 1. *Загрузка ресурсов.*

Загрузка осуществляется через класс ResourceManager, который возвращает специальный объект ресурса по данному пути к требуемому файлу, а также в нём кэшируются уже загруженные ресурсы.

* + 1. *Hardware Interfaces*

Нужна поддержка клавиатуры и мыши.

* + 1. *Software Interfaces*

Для работы движка требуются:

* поддержка OpenGL версии 3.0 и выше
* операционная система Windows версии 7 и выше
* IDE Visual Studio версии 2015 и выше
  1. Product Functions

Функции игрового движка делятся на:

* отрисовку графики
* симуляцию физики
* обновление компонентов
* управление сценами
* управление ресурсами
* обработку устройств ввода
  + 1. *Отрисовка графики*

1. Создание материала с заданными текстурами и шейдером
2. Отрисовка статичных (неизменяющихся) трехмерных моделей с помощью материалов
3. Отрисовка динамических трехмерных моделей со скелетной анимацией
4. Отрисовка систем частиц
5. Отрисовка skybox
   * 1. *Симуляция физики*
6. Проверка столкновений между:

* сферами
* AABB
* сферой и AABB
* сферой и треугольником
* сферой и набором треугольников
* AABB и треугольником
* сферой и набором треугольников

Между треугольниками столкновения не проверяются.

1. Проверка пересечения луча и отрезка с AABB, сферой, треугольниками
2. Компонент Rigidbody представляет динамический физический объект
3. Компоненты для проверки столкновений:

* AABBCollider – содержит AABB
* SphereCollider – содержит сферу
* MeshCollider – содержит набор треугольников

Эти компоненты являются ICollider

1. Entity с компонентами Rigidbody и ICollider будут взаимодействовать с другими физическими объектами
2. Entity с компонентом Collider, но без Rigidbody представляет статический физический объект
3. Компонент ICollider также содержит поле bool IsTrigger. Если IsTrigger активирован, то физические объекты с этим ICollider не будут сталкиваться, но им будут посланы сообщения.
4. Компонент Rigidbody также содержить поле bool IsKinematic. Если IsKinematic активирован, то силы, столкновения не будут влиять на этот Rigidbody, но остальные физические объекты будут взаимодействовать с этим Rigidbody.
   * 1. *Обновление компонентов*
     2. *Управление сценами*
     3. *Управление ресурсами*
     4. *Обработка устройств ввода*
   1. Communication Requirements

Нет.

* 1. Performance Requirements

Система должна обеспечивать хотя бы 30 кадров в секунду в простых сценах.

* 1. Memory Requirements

Система должна занимать не более 1 гигабайта в оперативной памяти и 1 гигабайта во внешней памяти.

* 1. Site Adaptation Requirements

Нет.

* 1. Design Constraints

Нет.

* 1. Software System Attributes
     1. *Reliability.*

Никогда не должно происходить аварийное завершение системы.

* + 1. *Availability.*

Нет специальных требований по доступности.

* + 1. *Security and Privacy.*

Нет специальных требований по безопасности и конфиденциальности.

* + 1. *Maintainability.*

Только основные функции должны быть полностью задокументированы.

* + 1. *Portability.*

Система должна быть портируемой на любую систему Windows версии 7 и новее.

* + 1. *Safety.*

Нет специальных требований по безопасности.

1. **Verification**
2. **Supporting Information**

Нет специальной дополнительной информации.

1. **Appendices**
   1. Assumptions & Dependencies

Нет.

* 1. Acronyms & Abbreviations
* AABB – axis aligned bounding box
* OBB – object oriented bounding box
* OpenGL – Open Graphics Library