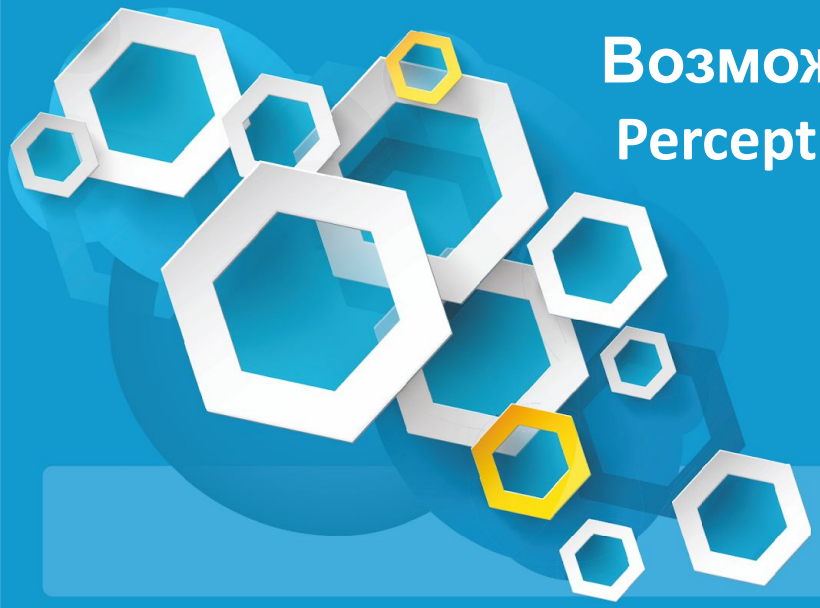
Естественно-интуитивное взаимодействие с компьютером

Лекция 6

Возможности использования Intel Perceptual Computing SDK в игровых приложениях



Содержание лекции

6.1. Введение

6.2. Примеры и возможности использования идей естественно-интуитивного взаимодействия в игровых приложениях

6.3. Игровые движки и фреймворки, основные возможности

6.4. Возможности интеграции Intel Perceptual Computing SDK с игровыми движками

Введение

Современный мир невозможно представить без игровой индустрии

Приложения, использующие для взаимодействия

устную речь на естественном языке,

мимику

и жесты,

позволяют пользователям прочувствовать эффект присутствия и погружения.

Именно такие эффекты востребованы в игровых приложениях.

Введение

Приложения с элементами естественно-интуитивного взаимодействия могут дать пользователям ощущение присутствия и погружения.

Добавить в приложение элементы такого взаимодействия позволяет комплект разработки Intel Perceptual Computing SDK и камера Creative* Senz3D

или

новая разработка Intel® RealSense™ SDK, работающий с камерой Intel® RealSense™ 3D Camera.

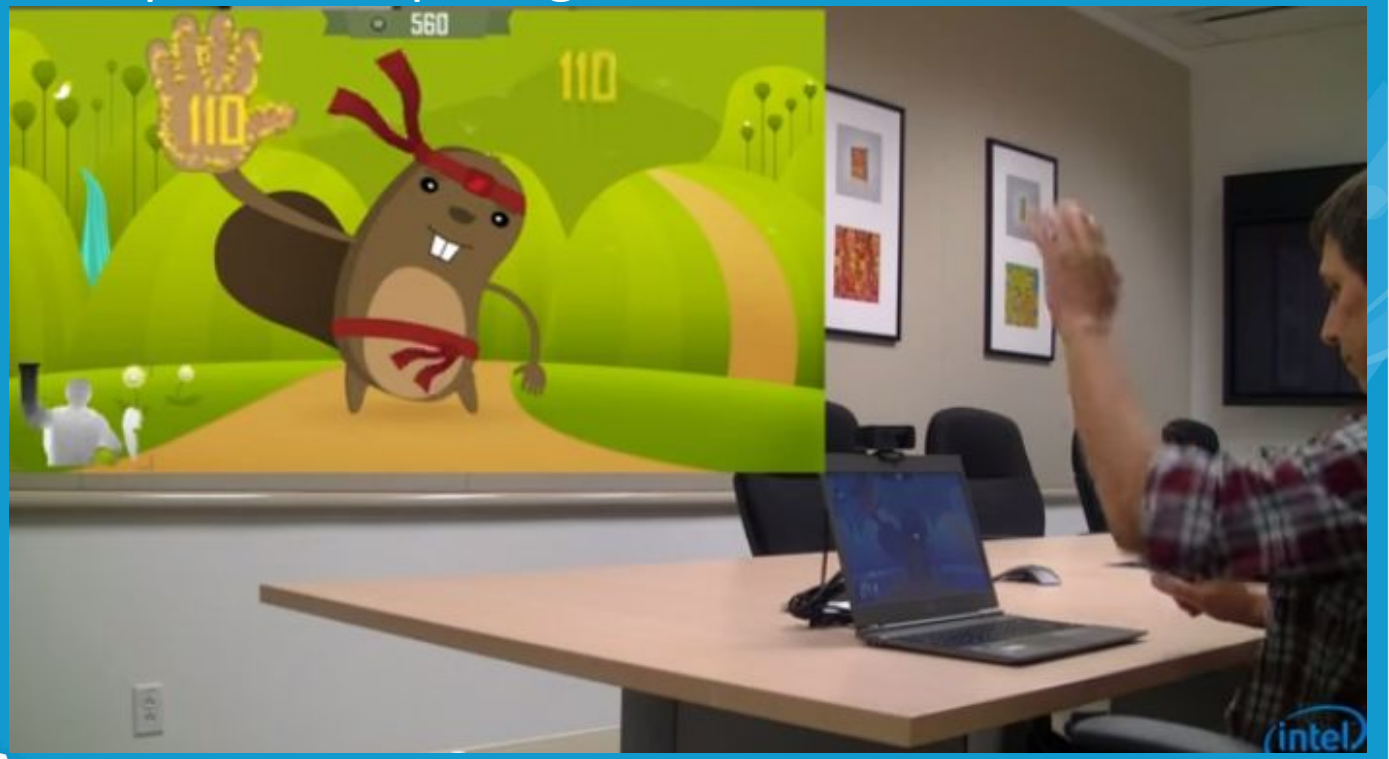
Примеры игровых приложений

Millions of Minions Perceptual Computing Demo

https://www.youtube.com/watch?v=gqUFnKvZ84c

Примеры игровых приложений

Kung Pow Kevin Perceptual Computing Demo



https://www.youtube.com/watch?v=catt5ZHCnl4

Примеры игровых приложений

Gesture Drive Intel Perceptual Computing

https://www.youtube.com/watch?v=4ODA2uWe7yQ

Примеры игровых приложений

Gesture Drive Intel Perceptual Computing

https://www.youtube.com/watch?v=KPE332rX5Yc

Примеры игровых приложений

Gesture Spell Casting Game Demo

https://www.youtube.com/watch?v=V9ut24ua-zs

Примеры игровых приложений

Gesture Spell Casting Game Demo

https://www.youtube.com/watch?v=V9ut24ua-zs

Примеры игровых приложений

"Perceptual Pet" Game

https://www.youtube.com/watch?v=71WHObUzRaQ

Примеры игровых приложений

PuppeTree(Intel Perceptual Computing challenge)

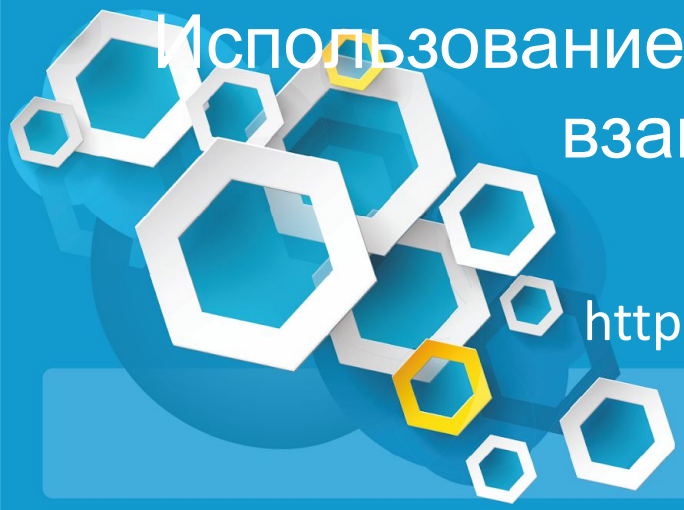
https://www.youtube.com/watch?v=LCWPv67uhxE

Возможности Perceptual Computing в играх

Идеи естественно-интуитивного взаимодействия можно успешно применять при создании игр-головоломок. Например, было бы естественным использовать жесты для сборки картинки из отдельных частей в пазлах или для управления падающими фигурками в тетрисе.

Использование идей естественно-интуитивного взаимодействия является будущим индустрии компьютерных игр.

https://www.youtube.com/watch?v=1edFLuMvyOs



Игровые фреймворки, основные возможности

Фреймворк – каркас для разработки приложений, облегчающий разработку и объединение разных компонентов большого программного проекта.

В отличие от библиотек, которые объединяют набор подпрограмм близкой функциональности, фреймворк содержит в себе большое количество разных по назначению библиотек.

Примеры фреймворков для разработки игр:

openFrameworks

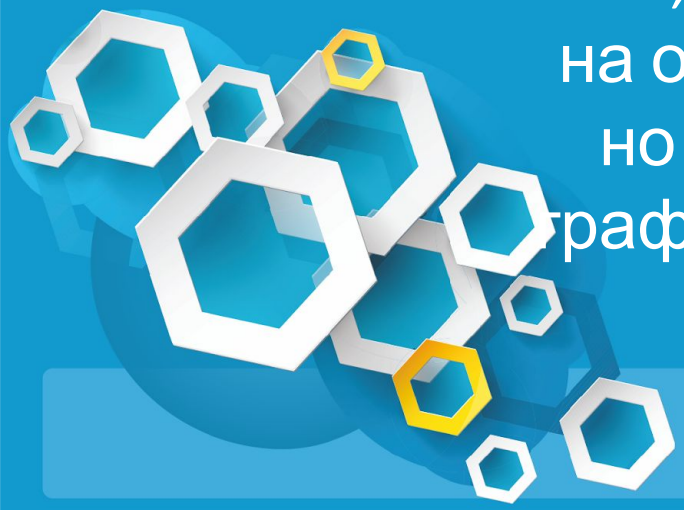
Processing.

Processing

Объединяет в себе язык программирования с открытым исходным кодом и интегрированную среду разработки.

Преследует цель обучения программированию в графической среде и служит основой для электронного альбома рисунков.

Язык, используемый в Processing, построен на основе языка программирования Java, но использует упрощенный синтаксис и графическую модель программирования.



OpenFrameworks

набор инструментальных средств с открытым кодом, разработанный для «творческого кодирования», написан на C++ и может работать под Windows, MacOS X, Linux, iOS и Android.

OpenFrameworks концентрируется на «творчестве» и использует изображения, предоставляет простой интерфейс к мощным библиотекам.

Многие пользователи отмечают сходство

openFrameworks

И

Processing

Игровые движки, основные возможности

Игровой процессор (движок) включает в себя инструменты, созданные для упрощения и ускорения процесса разработки игр, в отличие от фреймворка, который просто предоставляет набор полезных и удобных библиотек, процессор (движок) определяет логику игры.

Примеры

Unity

и

Havok Vision

Unity (или Unity3D)

кроссплатформенный игровой процессор, который имеет встроенную интегрированную среду разработки, создан Unity Technologies. Свыше миллиона разработчиков используют Unity при создании видео игр для различных платформ.

Графический процессор использует Direct3D (Windows, XBOX 360), OpenGL (Mac, Windows, Linux), OpenGL ES (Android, iOS) и собственные API (Wii).

Unity (или Unity3D)

возможности:

- рельефные преобразования;
- зеркальные отражения;
- преобразования смещения;
    - алгоритм SSAO, работающий в режиме реального времени и имитирующий рассеянное непрямое освещение и соответствующее затенение в трёхмерном виртуальном пространстве;
    - динамические тени, используются растровые изображения теней.

Unity (или **Unity3D**)

Unity поддерживает форматы файлов и 3D модели, созданные в 3Ds Max, Maya, Softimage, Blender, modo, ZBrush, Cinema 4D, Chetah 3D, Adobe Photoshop, Adobe Fireworks и Allegorithmic Substance.

Созданные с помощью любого из перечисленных приложений 3D объекты могут быть добавлены в игровой проект и управляться из графического пользовательского интерфейса Unity.

Havok Vision

кроссплатформенный игровой процессор (движок), который предоставляет мощную и гибкую мульти-платформенную технологию, идеально подходящую для разработки игр любого вида и способную обсчитывать и изображать очень сложные сцены, обеспечивая плавную смену кадров.

Технология содержит возможности, позволяющие разработчикам игровых приложений преодолеть технические барьеры и создавать действительно привлекательные игры.

Интеграция Intel Perceptual Computing SDK с игровыми движками

Intel® Perceptual Computing SDK поддерживает следующие игровые движки и фреймворки:

- Unity PRO 3.5.1f2 (или более поздний);
- Havok* Vision* SDK version 2012.2.1 (или более поздний);
- openFrameworks v0071 (или более поздний);
- Processing 1.5.1 (или более поздний).

Интеграция Intel Perceptual Computing SDK с игровыми движками

При разработке игровых приложений с помощью Havok* Vision* SDK или openFrameworks можно использовать все функции и утилиты C/C++ библиотек Perceptual Computing SDK в исходном формате, достаточно соответствующим образом настроить проект.

Настройка проекта
Havok* Vision* SDK или openFrameworks

Для приложений, использующих при компиляции опцию статической загрузки, Runtime Library (библиотеку времени выполнения) Multi-Threaded Debug (/MTd) или Multi-threaded(/MT) необходимо:

импортировать страницу свойств

props/VS2010-12.Integration.MT.props

Настройка проекта
Havok* Vision* SDK или openFrameworks

Для приложений, использующих при компиляции опцию динамической загрузки, Runtime Library Multi-Threaded Debug DLL(/MTd) или Multi-threaded DLL(/MT) необходимо:

импортировать страницу свойств

props/VS2010-12.Integration.MD.props,

Настройка игрового приложения, разрабатываемого с помощью Unity

необходимо выполнить следующие действия:

- создать директорию Plugins в папке Assets проекта Unity;
- из папки

$(PCSDK_DIR)/framework/Unity/hellounity/Assets/Plugins скопировать в созданную директорию Plugins следующие

файлы:

libpxcupipeline.dll

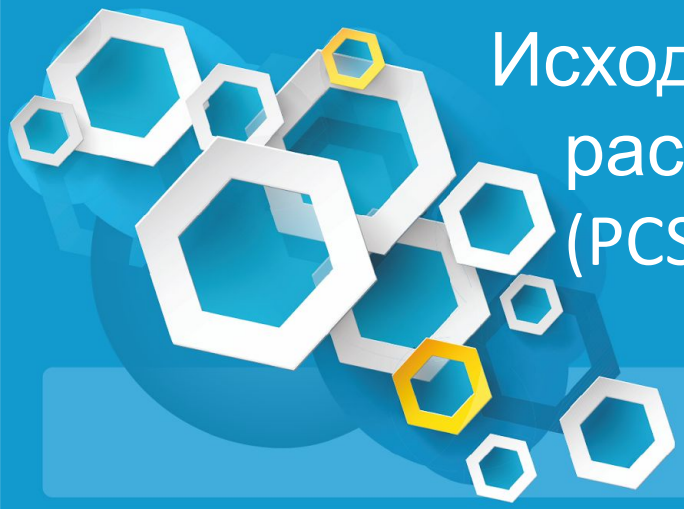
pxcm-structures.cs

pxcupipeline.cs

Настройка игрового приложения, разрабатываемого с помощью Unity

Необходимо иметь ввиду, что **libpxcupipeline.dll** скомпилирован, как 32-битная библиотека,

если разрабатываемое Unity приложение является 64-битным, необходимо перекомпилировать файл **libpxcupipeline.dll**, как 64-битный и его скопировать в директорию Plugins разрабатываемого приложения.



Исходные файлы библиотеки располагаются: (PCSDK_DIR)/framework/common/pxcupipeline.

Настройка игрового приложения, разрабатываемого с помощью Processing

- скопировать директорию libraries с библиотекой PXCUPipeline в папку, на которую указывает Sketchbook location, из каталога: $(PCSDK_DIR)\framework\Processing\libraries;

•для реализации возможности взаимодействия с Intel Perceptual Computing SDK необходимо подключить библиотеку PXCUPipeline:

```
import intel.pcsdk.*;
```

Интеграция Intel Perceptual Computing SDK с игровыми движками

Общие процедуры программирования схожи для поддерживаемых фреймворков и движков, поскольку их поддержка основана на одной и той же базовой библиотеке.

Процедуры программирования

Инициализация.

Используется функция **Init()**, в которой выполняется инициализация конвейера, основанного на классе **UtilPipeline**.

Этот конвейер объединяет в себе обработку цветных изображений, определение и отслеживание положения лица и основных маркеров, отслеживание рук и пальцев, распознавание жестов, распознавание речи и голосовых команд.

Процедуры программирования

Обработка данных.

AcquireFrame() - блокирует фрейм до получения результатов обработки данных.

ReleaseFrame() - снимает блок с фрейма и подготавливает обработку следующего фрейма.

Между вызовами этих двух функций приложение может выполнить серию вызовов функций, которые возвращают результаты распознавания и обработки цветных изображений, мимики, жестов и речи.

Процедуры программирования

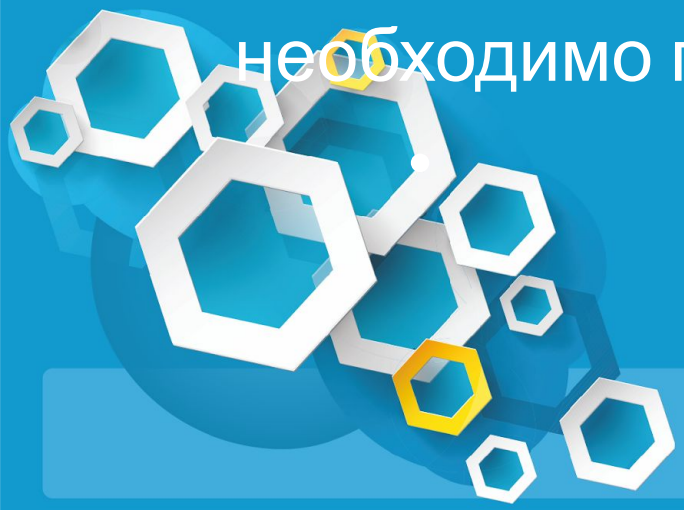
Завершение работы.

Приложение использует функцию **Close()** для закрытия конвейера и освобождения всех ресурсов.

Заключение

Идеи естественно-интуитивного взаимодействия и возможности Intel Perceptual Computing SDK находят самое широкое применение в развитии игровых приложений.

Многие технически сложные вопросы помогают решить игровые движки и фреймворки, но чтобы эффективно с ними работать, необходимо потратить время на их освоение.



Использованные источники

https://software.intel.com/en-us/articles/intel-perceptual-computing-sdk-tutorials

https://software.intel.com/sites/default/files/article/328725/sdktutorial-havok.pdf

http://www.openframeworks.cc/

https://www.processing.org/

http://unity3d.com/

http://www.havok.com/products/vision-engine

http://www.intuit.ru/studies/courses/1104/251/info

http://gamesisart.ru/game_dev_create.html