Intel(R) Perceptual Computing SDK Demo Application

Glass Balls

Table of Contents

1 Legal Disclaimer

Information in this document is provided in connection with Intel products. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Except as provided in Intel's Terms and Conditions of Sale for such products, Intel assumes no liability whatsoever and Intel disclaims any express or implied warranty, relating to sale and/or use of Intel products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right.

Unless otherwise agreed in writing by Intel, the Intel products are not designed nor intended for any application in which the failure of the Intel product could create a situation where personal injury or death may occur.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "Reserved" or "Undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or by visiting Intel's web site http://www.intel.com.

Any software source code reprinted in this document is furnished under a software license and may only be used or copied in accordance with the terms of that license.

Intel, the Intel logo, and Intel Core are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

*Other names and brands may be claimed as the property of others.

1.1 Optimization Notice

Intel’s compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804

2 Introduction

The Intel® Perceptual Computing SDK is a library of pattern detection and recognition algorithm implementations exposed through standardized interfaces. The library aims at lowering barriers to using these algorithms and shifting the application developers' focus from coding the algorithm details to innovating on the usage of these algorithms for next generation human computer experience.

This document describes the `GlassBalls` demo application.

Document Convention

The SDK document uses the Calibri typeface for normal prose.

With the exception of section headings, captions and the table of contents, all code-related items appear in the Courier New typeface (`pxcStatus`).

Hyperlinks appear underlined in blue, such as <u>pxcStatus</u>.

🗒 This is a note that provides additional information to aid your understanding of the procedure or concept.

💡 This is a tip that provides alternate methods or shortcuts.

✓ This is a result statement which indicates what you can expect to see or happen after performing a step.

3 Glass Balls

GlassBalls is a sample application that showcases hand interaction with physical objects (glass balls). The user will be able to use his/her fingertips to interactive with glass balls. The application uses the Unity* game engine for 3D visualization and physics interaction.

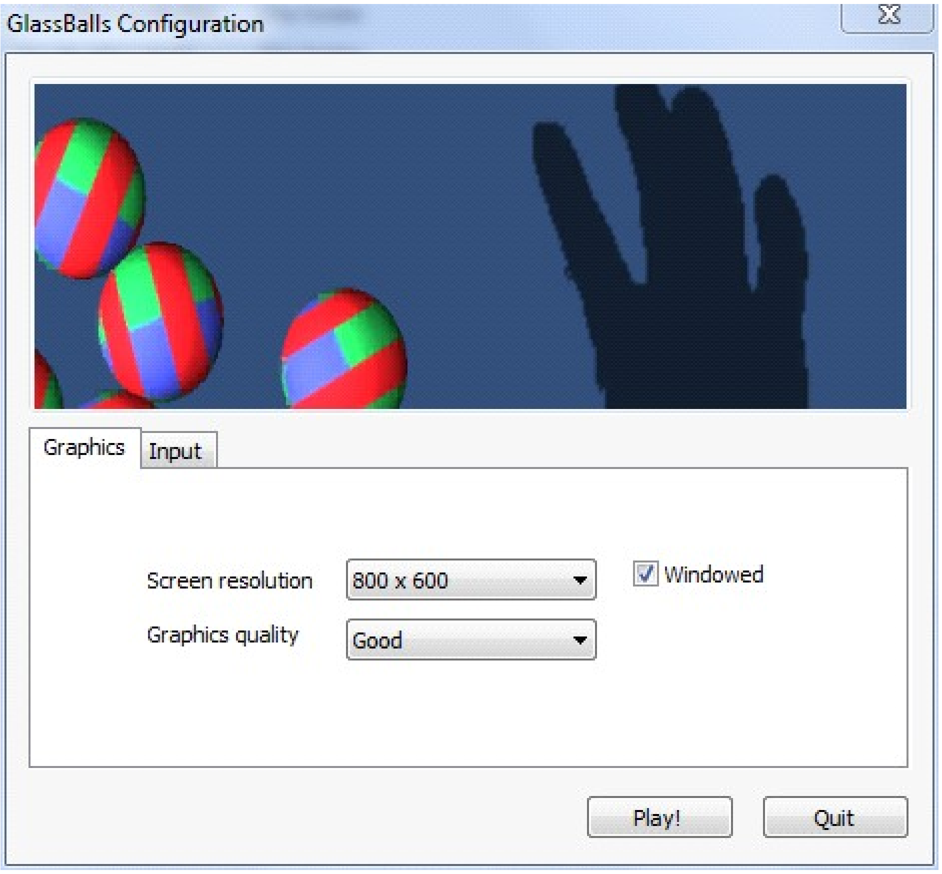
The application directory structure is as follows:

Directory/File	Description
Assets ProjectSettings	The Unity assets and project setting directories that contain buildable source code.
doc	The directory that contains this document, and two farm pictures.
GlassBalls_Data GlassBalls.exe	The prebuilt application executable and data files.

3.1 Operation Instructions

Run the prebuilt executable GlassBalls.exe to start the application. By default, the application uses the screen resolution 800x600. The user can do one of the following to change the screen resolution, as illustrated in Figure 1:

- Start the application by holding down the SHIFT key.
- Start the application in a DOS command prompt window.



Screen resolution: 800 x 600

☑ Windowed

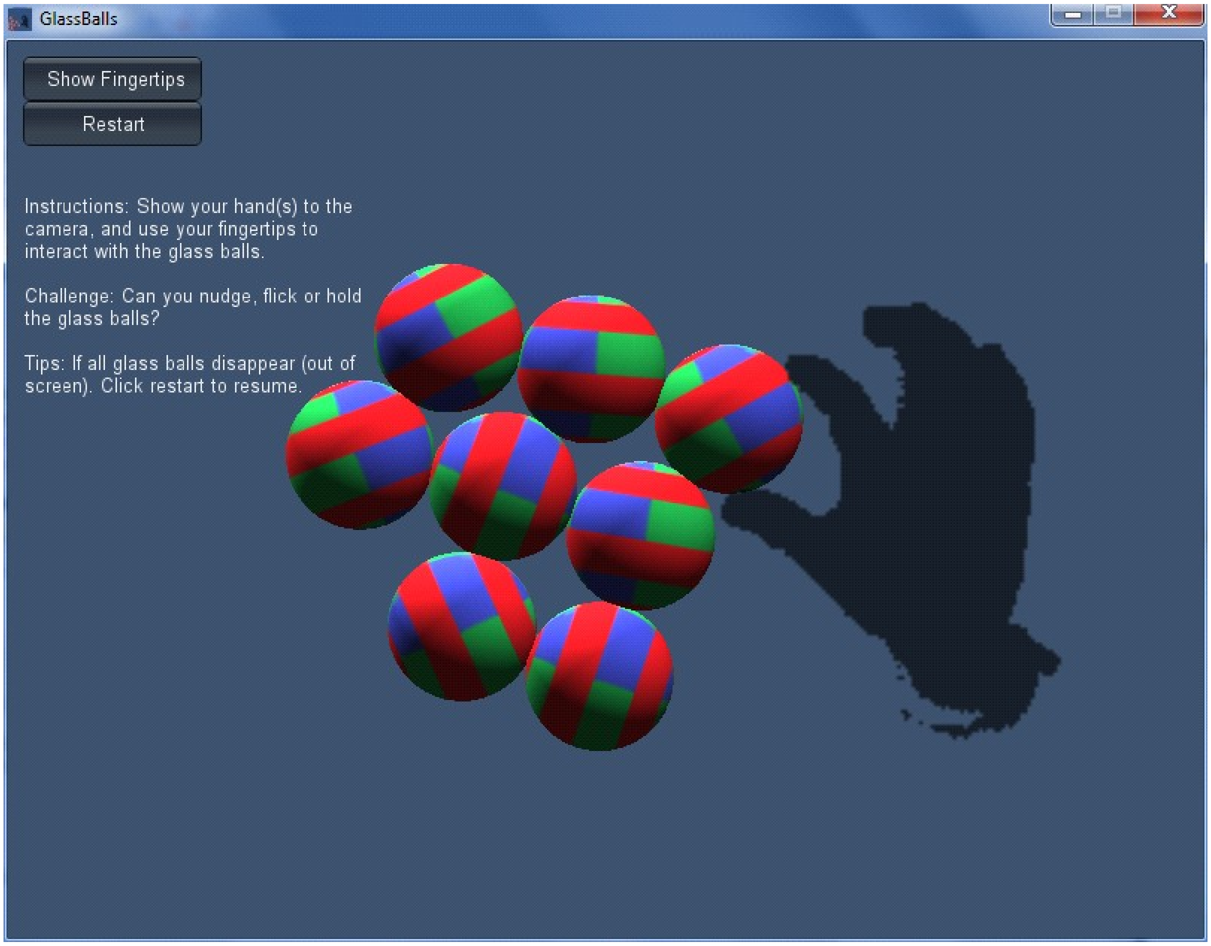
Graphics quality: Good

Figure 1: Screen Resolution Configuration

The application shows a few glass balls at the center of the screen, and some instructions and buttons at the left. When the user shows his/her hand(s) in front of the camera, the application will visualize with the hand shadow(s). The user will be able to use the shadow fingertip(s) to interact with the gall balls, as illustrated in Figure 2.

Click the Show Fingertips button will show the actual interaction points as illustrated in Figure 2. It is recommended to hide the points to make the experience more natural.

The application applies a constant force towards the center of the screen to the balls so that the balls do not fly away. Thus the balls always have a tendency to gather around the center of the screen. In rare cases, mostly due to unreliable fingertip detection, the balls may suddenly gain huge velocity and fly off the screen. Click the Restart button to re-start the application.

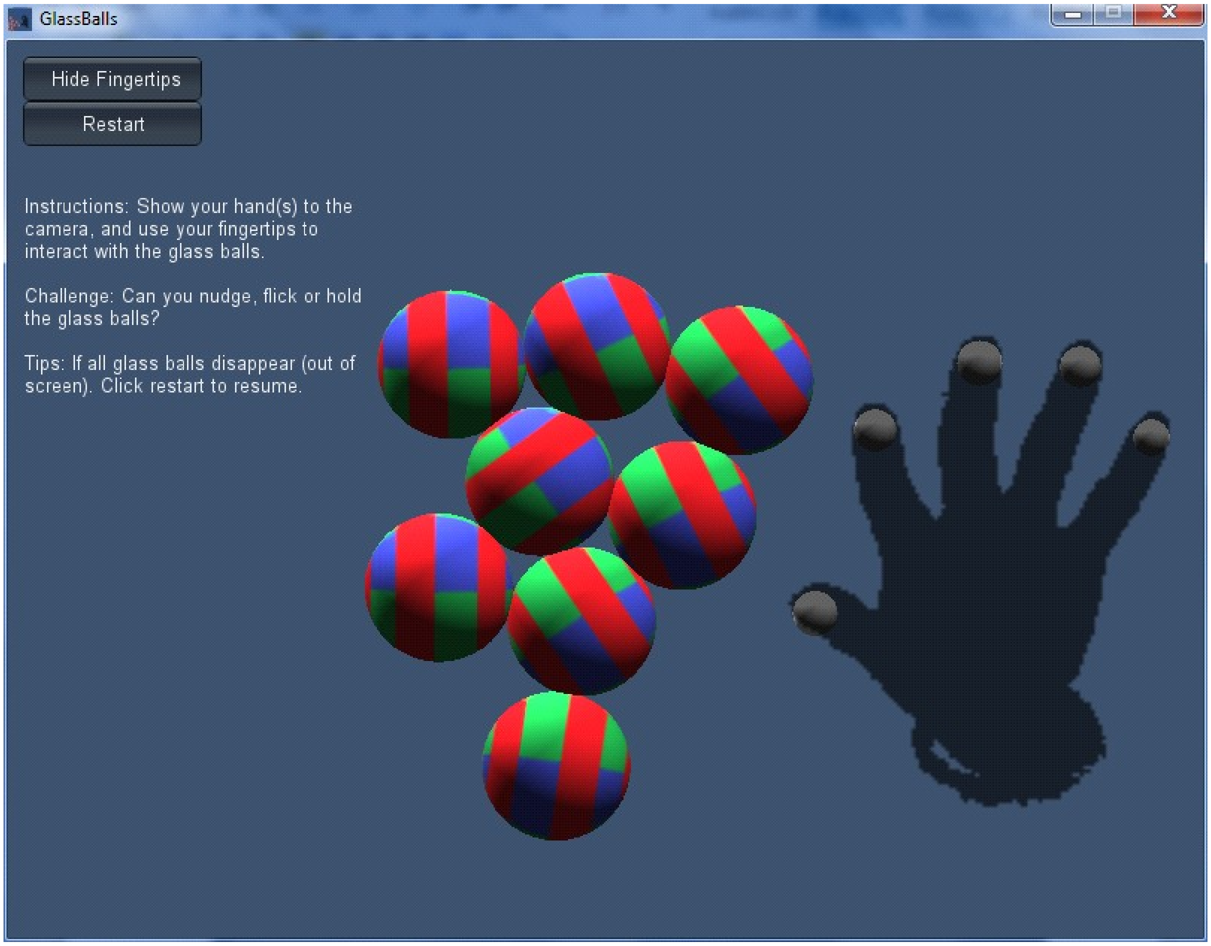


Instructions: Show your hand(s) to the camera, and use your fingertips to interact with the glass balls.

Challenge: Can you nudge, flick or hold the glass balls?

Tips: If all glass balls disappear (out of screen). Click restart to resume.

Figure 2: Hand Interaction with GlassBalls



GlassBalls

Instructions: Show your hand(s) to the camera, and use your fingertips to interact with the glass balls.

Challenge: Can you nudge, flick or hold the glass balls?

Tips: If all glass balls disappear (out of screen). Click restart to resume.

Figure 3: Fingertip Interaction Points

3.2 Build Instructions

The application comes with source code in the `Assets` and `ProjectSettings` directories. The solution file is under `Assets/GlassBalls.unity`. Launch Unity by double clicking on the solution file. Then the developer will be able to make modification or rebuild the application. The application is tested under Unity PRO 3.5.1f2.

3.3 Design Points

This section describes the application design that is related to Intel® Perceptual Computing SDK and is not meant to be a code workthrough of all application designs.

The application creates 8 glass balls with limited Z-axis movement so that they move only in the

XY 2D space. Each glass ball is a rigidbody sphere, whose movement is governed by physical interaction. The script `BallScript.cs` attached to each glass ball applies constant force to the ball to make it by default move to the center of the screen. The script also checks whether the ball flys out of screen. If so, move it back.

The application also creates 10 fingertips (primary and secondary hands in the view). Each fingertip is a kinematic rigidbody sphere, whose size and position are assigned, at fixed interval, from detected finger parameters. See the script `Fingertip.cs` for details.

As the user interacts with the glass balls, the fingertip sphere objects interact with the glass ball sphere objects and cause the glass ball sphere objects to move in different directions.

The `ShadowHand.cs` script runs the SDK finger tracking pipeline and displays the shadow hand(s) onto the screen. At each frame, the script retrieves the detected labelmap, removes any background and displays the shadow hand image. The script also retrieves detected fingertip parameters, to be used by the `Fingertip.cs` script.

The script `Options.cs` handles on-screen instructions and button clicks.