Kinect Superman

Play Superman with Kinect

A project by:
Sicco van Sas
Daniël Karavolos
Maarten van der Velden
Roberto Valenti

For:
UNIVERSITY OF AMSTERDAM
ThirdSight

January 2011

Goal

The goal was to create an immersive virtual environment that tracks the movements of the user, that recognizes specific gestures and reacts to these in a natural feeling way. The user gets feedback through a stereo-vision head-mounted screen and stereo audio. We try to achieve this goal by implementing a 3D game that lets the user take control of ISLA- man, a super hero who looks not completely unlike Superman in appearance and behavior. The game setting gives the user a goal to achieve while having the experience. This might commit the user even more to the virtual environment.

The Project

A long time ambition of the computer science research and industry is to build immersive computer interfaces, in which the user experiences a virtual reality with which he can interact and which behaves in a way that is natural to the user. There have been many attempts to create this kind of experience and to make it accessible to common users.
Recent software developments in movement and gesture recognition from video imagery have been accompanied by the development of hardware that enables recognition in real-time. This has led to the emergence of Microsoft's Kinect system. This is the first user tracker device for the consumer market. The release of the Kinect has interested many hobbyists and researchers, because it provides new opportunities for interaction with computer systems. In the few months since its release quite a number of programs have been developed that enable the Kinect to be used from a PC. These so-called "Kinect hacks" have received much attention.

Why Superman?

We settled for a Superman-like game after reviewing various Kinect games and demo applications. One of the main limitations of the Kinect is that it does not allow the user to move a lot. The user should remain within sight of Kinect's cameras. Therefore it is inconvenient to make a game where the user walks around a virtual environment. Not needing your feet to move around makes sure that the user stays in one place. Other games and demos solve this problem by mounting the user on some kind of moving platform, leaning in a certain direction or swaying the arms to control the legs. We thought it would be interesting to solve this problem by flying. Superman flew into our thoughts quite soon after that. Obviously his other superpowers allow for more diverse interactions with the environment and implementing these superpowers contribute to the challenge of gesture recognition.

The Kinect

The Kinect system consists of a few components. It has a plain RGB camera, two microphones and a tilt motor, but (for this project) the most important part is the distance estimation. This is done with an infrared emitter and am IR camera. The emitter sends a grid of IR dots to the objects in its field of reach, after which the camera will estimate the distance of each point to the Kinect. This is treated as a fourth image layer and sent to the computer attached.

Kinect Drivers

Shortly after the release of the Kinect for the Xbox 360, some open source drivers were released to enable PCs to use the system. The one we used is OpenNI (Open Natural Interface). The advantage of OpenNI compared to other drivers available is that it has integrated body tracking features. It can discern users from each other and it models user joints with a stick figure model.

OpenNI actually consists of a series of drivers and programs that are connected to each other. Reading the input from the Kinect is done by a driver called SensorKinect, which enables OpenNI to use the data. This program does the main processing of the Kinect data, but also uses another program called NITE to do higher order recognition, such as skeleton tracking. OpenNI and NITE were (partly) developed by Primesense before the Kinect was released. Primesense has its own user tracking device and now lets Kinect users use their software too.

Finally, to port the output of NITE to a game engine (in our case Unity3D), a wrapper is needed that reads the NITE libraries. OpenNI has also provided this in a crude demo application.

iWear VR920

The VR glasses we used are the Vuzix iWear VR920 glasses. These glasses have a built in gyroscope and stereovision, both of which enhance the immersive experience.

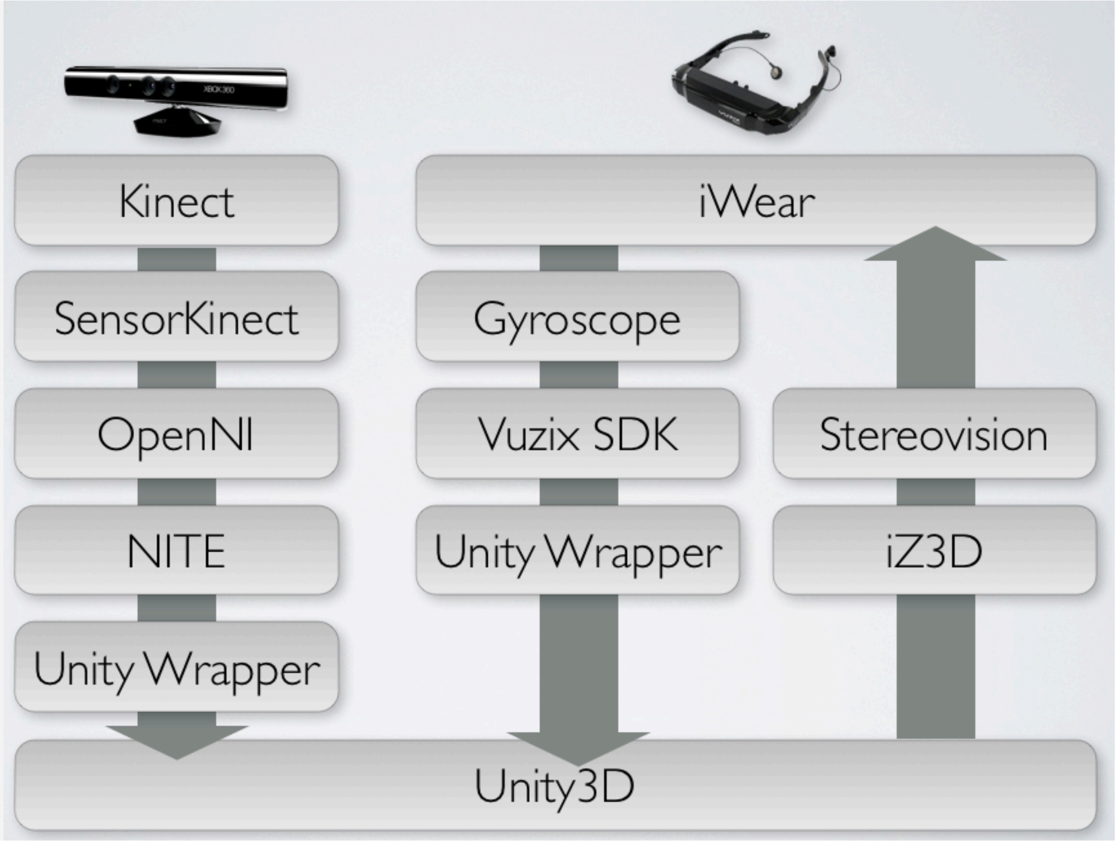
Gyroscope

The gyroscope in the glasses keeps track of the three rotation axes. It tracks yaw (turn your head left to right), pitch (upwards to downwards) and roll (turn your head sideways towards your left or right shoulder). In this way it enables the game to track head movements and map it to rotations of the first person camera view. This is done with the regular drivers of the glasses and with Vuzix's SDK, which supplies a library of functions to get the (calibrated) rotations of the three axes. We built ourselves a wrapper to get these functions into the game engine, because the libraries were in a static C++ library, whereas the game engine runs on scripts that can handle dynamic C++ libraries only.

Stereovision

The Vuzix iWear VR920 contains two independent 640×480 screens, one for each eye. This means that the screens can show different images, which can be used to create a real 3D effect in virtual worlds. The iZ3Ddriver shifts the camera in our game world slightly to the right and to the left and sends these signals to their respective screens, resulting in stereo vision similar to how humans normally perceive the world.

Configuration



Setup of the hardware and drivers.

How it was made

We created an interactive world with a Superman avatar using the Unity3D game engine. Using the existing drivers, we mapped the user's movements onto the ones of Superman. The body movements are tracked by the Kinect, the head movements by iWear's gyroscope.

On top of that we have conditioned special actions (i.e. the superpowers) on certain gestures the user makes. A stretched arm makes Superman fly in the direction it is pointing, two stretched arms make him accelerate in flight. Making a 'salute' gesture (touching your glasses) makes Superman shoot lasers from his eyes or blow an ice breath, depending on the arm. There is also a gesture to interact with the game settings: stretching your arms wide toggles between first-person and third-person perspective.

Possible applications

With this prototype game we tried to show a subset of the possible applications of tracking devices and 3D glasses available on the consumer market. Besides the obvious possibilities it creates for games, like the one that was made, there are more ways in which this kind of applications might be useful. To name a few: It could be used as an enhanced simulator for educational use (learn to drive, learning traffic rules to children). In architecture, similar applications might allow the user to move around and interact with models of a building or neighborhood. On a more abstract level the technique might also be used to enhance data visualization by allowing users to actually see data in 3D, move around it to get more detail or more overview and interact with it to show different aspects.

This project shows an approach that can be taken in developing applications like that.

News and media coverage

Some of the website that have covered our project.

- **computerandvideogames.com**
- **kinecthacks.net**
- **joystiq.com**
- **New Scientist**
- **kinect-hacks.com**
- **i-programmer.info**

More information

Do you have questions?

Mail us at **kinectsuperman@gmail.com**

Visit our webpage at **www.thirdsight.co/research/projects/kinect-superman**