Posture Control Of Unity 3D

Mihir Dhanani
Department of Computer Science and Software Engineering University of Canterbury
Christchurch, New Zealand
Email:msd54@uclive.ac.nz

Richard Green
Department of Computer Science and Software Engineering University of Canterbury
Christchurch, New Zealand
Email:richard.green@canterbury.ac.nz

***Abstract*—In this paper, we explore the capacity of using skeleton information provided by Kinect for human posture recognition to use as a controller this is achieved by using Visual Odometry based on RGB-D Images from the Kinect sensor, a framework will be deployed based on Microsoft Kinect using OpenNI and Unity to animate in real-time. Kinect runs 30fps and we've had 90% accuracy identifying the posture.**

Keywords- Human posture recognition, human skeleton, Kinect, Unity3D, OpenNI, NITE,Zigfu, games, interaction design

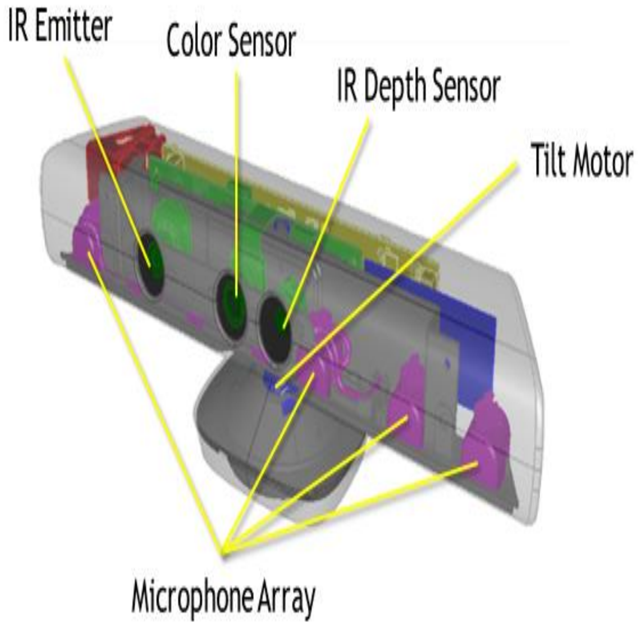
I. INTRODUCTION

With the invention of the Microsoft Kinect sensor, high-resolution depth and visual (RGB) sensing has become available for widespread use. The complementary nature of the depth and visual information provided by the Kinect sensor opens up new opportunities to solve fundamental problems in computer vision. Prior Kinect based Unity 3D research and applications include preprocessing, object tracking and recognition, human activity analysis, gestures, and indoor 3-D mapping. Human Posture Recognition can be viewed as a sub-field of gesture recognition since a posture is a "static gesture". In practice, posture recognition is usually at the crossroad between people detection and gesture recognition. Sometimes we are only interested in the posture at a given time which can be performed by a people detector [1]. In other cases, posture detection can be sometimes considered as a first step for gesture recognition, for instance, by associating postures to states of a Finite State Machine (FSM) [2]. The challenges of posture recognition are seamlessly the same as gesture recognition except that the temporal aspect is not accounted.

In our work, we are interested in detecting human postures for event modelling and recognition. Kinect device allows to capture not only colour information as conventional camera but also depth and motion information. This device has become quickly popular because of its low cost as well as its free SDK. In this paper, we explore the possibility of using Kinect device as an input controller.

II. BACKGROUND

Kinect has a wide range of applicability. Previous research consists of an integrated hardware and software platform developed for rapid prototyping of virtual reality-based games. The exercise protocol has been adopted from an evidence-based shoulder exercise program for individuals with spinal cord injury. The hardware consists of a custom metal rig that holds a standard wheelchair, six Gametraks attached to elastic exercise bands, a Microsoft Kinect, a laptop and a large screen. A total of 21 prototypes using drivers for Kinect and Unity Pro 3 in order to evaluate game ideas based on deconstruction of the exercise protocol. There also exists KiReS which is a telerehabilitation system that offers the users and the physiotherapist's innovative features by using Kinect as interaction device.



Kinect has various advantages: No Data Input Device Required, Voice Recognition, Facial Recognition and Helpful in rehabilitation, etc.

Prime Sense, whose depth sensing reference design Kinect is based on, released their own open source drivers along with motion tracking middleware called NITE. Prime Sense later announced that it had teamed up with Asus to develop a PC-compatible device similar to Kinect, which will be called Wavi Xtion and is scheduled for release in the second quarter of 2011. OpenNI is an open-source software framework that is able to read sensor data from Kinect, among other natural user interface sensors. Controller-free gaming means full body play. Kinect responds to how you move. Kinect uses advanced entertainment technology to respond to the sound of voice. Once we wave our hand to activate the sensor, Kinect will be able to recognize and access the Avatar. Kinect promises a gaming experience that's safe, secure and fun for everyone.

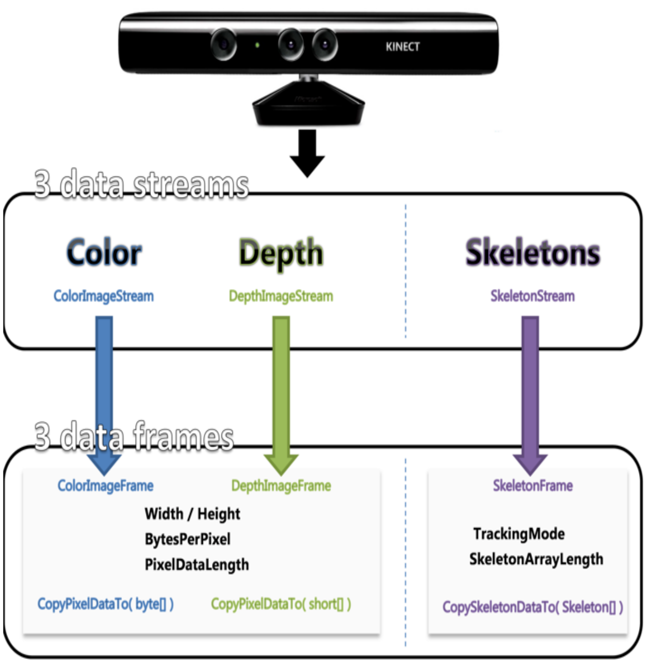
The Microsoft Kinect Development Kit supports both Unity native and Web Player builds for motion-controlled apps. The Unity package included with this Kit comes with several functional sample scenes based on the category of game to be created. The SDK for Unity3D works with OpenNI/NITE and the Microsoft Kinect SDK, and with all 3D sensors. This is a way to make cross-platform, motion-controlled apps with Kinect in HTML5/JavaScript, Unity3D and Flash. Applications made with this SDK are portable across all operating systems, web browsers, computer vision middleware's, and 3D sensors.

The Kinect uses structured light and Machine Learning. i.e. Inferring Body Position, which is a two stage process: first computing a depth map (using structured light), then infer body position (using machine Learning). The depth map is constructed by analyzing a speckle pattern of infrared laser light. This is done with the help of PrimeSense. The technique of analyzing a known pattern is called structured light. The general principle of structured light is to project a known pattern onto a scene and infer depth from the deformation of that pattern. The Kinect uses infrared laser light, with a speckle pattern. The depth map is constructed by analyzing a speckle pattern of infrared laser light. The Kinect uses an infrared projector and sensor, it does not use its RGB camera for depth computation. The Kinect combines structured light with two techniques: depth from focus, and depth from stereo. Depth from focus uses the principle that objects that are more blurry is further away. The Kinect uses a special ("astigmatic") lens with different focal length in x- and y- directions. A projected circle then becomes an ellipse whose orientation depends on depth. The astigmatic lens causes a projected circle to become an ellipse whose orientation depends on depth. Depth from stereo uses parallax. If we look at the scene from another angle, objects that are close gets shifted to the side more than objects that are far away. The Kinect analyzes the shift of the speckle pattern by projecting from one location and observing from another. Body parts are inferred using a randomized decision forest, inferring body position is a two-stage process: First compute a depth map, then infer body position. Use computer graphics to render all sequences for 15 different body types, and vary several other parameters, transforms depth image to body part image. Kinect uses a randomized decision forest. A randomized decision forest is a more sophisticated version of the classic decision tree, in practice, useful = information gain G (which is derived from entropy H):

$$G(\phi) = H(Q) - \sum_{s\in\{l,r\}} \frac{|Q_s(\phi)|}{|Q|} H(Q_s(\phi))$$

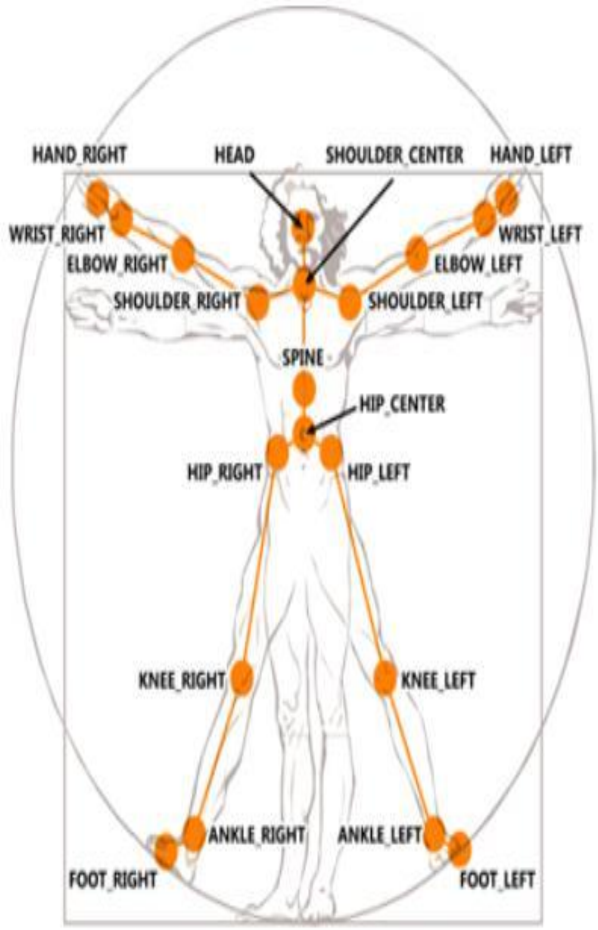
Second transform the body part image into a skeleton. The mean shift algorithm is used to robustly compute modes of probability distributions which is simple, fast, and effective.

The steps involved in Bringing the Kinect SDK in to Unity are [12]: Install the Unity Package: The Unity package (Kinect Wrapper Package for Unity3D) includes all of the scripts required to start running your world.



Setting up the Kinect Prefab: The Kinect Prefab Game Object is required to have Unity talk with / use the Kinect. This empty object needs to exist somewhere in your scene. However, if we are switching scenes, it will persist because of the DontDestroyOnLoad call in the KinectWrapper script, so we don't need to instantiate a new Kinect_Prefab in each scene. To get the best use out of the Kinect, we may need to make some changes with the settings within the Kinect_Prefab object itself.

Controlling the Character: Simply put, to get the character moving, we need to first attach either the KinectModelControllerV2 to control the bones of the model, or KinectPointController to control a series of GameObjects.



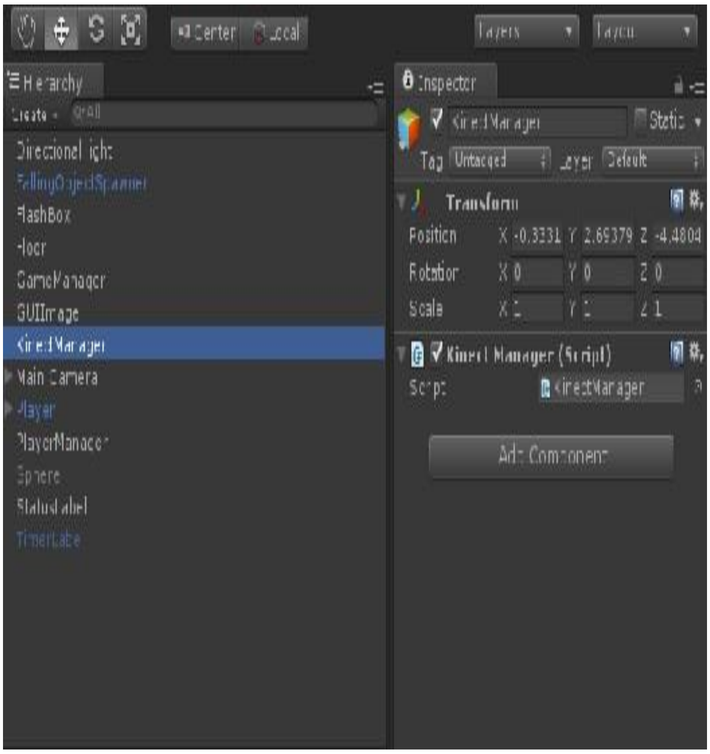
Unity supports deployment to multiple platforms such as mobile devices, web browsers, desktops, and consoles. Unity also allows specification of texture compression and resolution settings for each platform the game supports. The graphics engine uses Direct3D (Windows, Xbox 360), OpenGL, OpenGL ES (Android, iOS), and proprietary APIs (consoles). There is support for bump mapping, reflection mapping, parallax mapping, screen space ambient occlusion (SSAO), dynamic shadows using shadow maps, render-to-texture and full-screen post-processing effects.

Unity supports art assets and file formats from 3ds Max, Maya, Softimage, Blender, modo, ZBrush, Cinema 4D, Cheetah3D, Adobe Photoshop, Adobe Fireworks and Allegorithmic Substance. These assets can be added to the game project, and managed through Unity's graphical user interface. The ShaderLab language is used for shaders, supporting both declarative "programming" of the fixed-function pipeline and shader programs written in GLSL or Cg. A shader can include multiple variants and a declarative fallback specification, allowing Unity to detect the best variant for the current video card, and if none are compatible, fall back to an alternative shader that may sacrifice features for performance

III. METHOD

Create a new Unity project and name it. Once Unity loads the default scene begin importing the assets from the Unity package as required. Once the package is imported plug in the Microsoft Kinect. Check to see if Kinect drivers were installed correctly. If it was successful, you should notice a RGB display screen, a grayscale depth screen, a rainbow colored humanoid model, and various spheres all translating in conjunction with movements in real time. The algorithm used is the Visual Odometry based on RGB-D Images of a Microsoft Kinect sensor which is used to identify the human posture in unity and is displayed alongside in the scene and the decision forest the algorithm on which the Kinect is based on, provides an extremely reliable 3D joint orientation and location map which describes a human's pose in Cartesian space, for one person in the Kinect's view. This allowed us to build gestures and actions support on unity to allow the use of Kinect to with the environment.

The Kinect Prefab is the game object that needs to be referenced in the placeholder in the inspector. This Kinect Prefab will record motion data in real time and send the information to the controller scripts. The controllers scripts will convert this data and transform bones or Game objects that are related to specific parts of the body and mimic the relative placement in a 3d environment. A rigged model contains a digital skeleton bound to a 3D mesh. Expand the custom model fully so that each bone is visible in the hierarchy. Each digital bone that needs to be controlled must be referenced into the appropriate slot in the inspector. The mask will filter and limit what joints need to be animated through the Kinect. Any bones or game objects that are excluded from this list will remain static. Therefore, for the game only using hands and head, limit the range to track this by setting the mask to Upper Body.



The player field specifies which user controls the model. The KinectPointman has scale will transform the size of all game objects that are referenced and used to translate player movements. On the KinectModelControllerV2, to animate the model while the player is controlling it, set "animated" flag and determine how much blending between the animation and the Kinect should occur. [Range from 0 – 1]

IV. RESULTS

The system used to develop and testing purposes is the same, we used a laptop running i5 processor, 8 GB of Ram, Kinect sensor with prime sense version 5.1.2 having the drivers for windows and Unity3D 4.3.1.

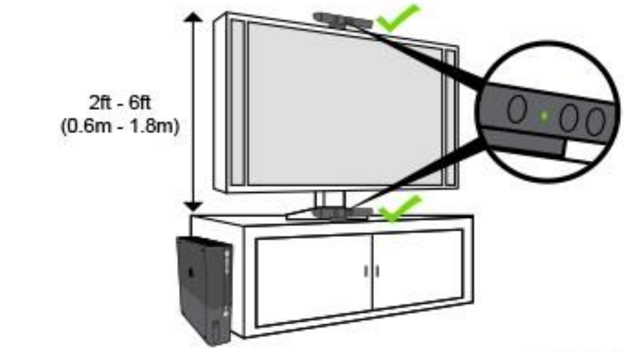
Positioning the Kinect sensor was an important factor for the result along with other environment issues. [21].

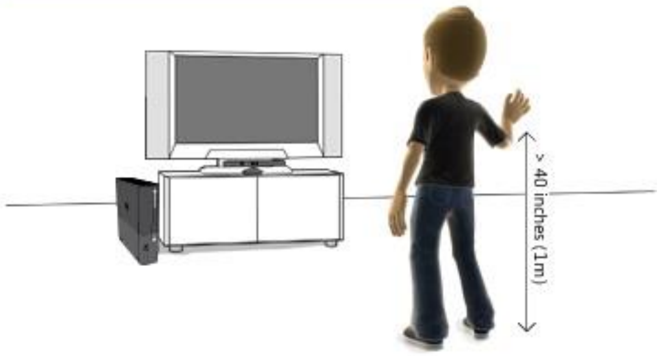
The sensor should be positioned 2 – 6 feet (0.6 – 1.8 m) off the floor, with nothing between the player and the sensor.

The area between the sensor and the players should be clear. **One player:** Stand back 6 feet (1.8 m). Make sure there is enough light so that the face is clearly visible and evenly lit and also to minimize side or back lighting, especially from a window.

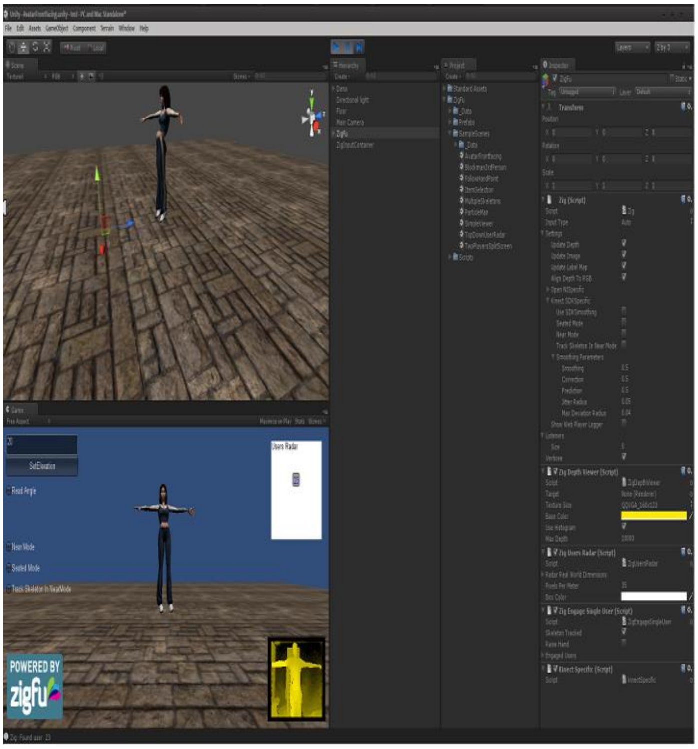
For best tracking results we had to avoid positioning the sensor or player in direct sunlight.

Some lighting conditions can make it hard for Kinect to identify the player or track movements, so we made sure the room has bright, even lighting.





The screenshot below are from the sample scene used to understand the implementation of a working Kinect into unity scene.

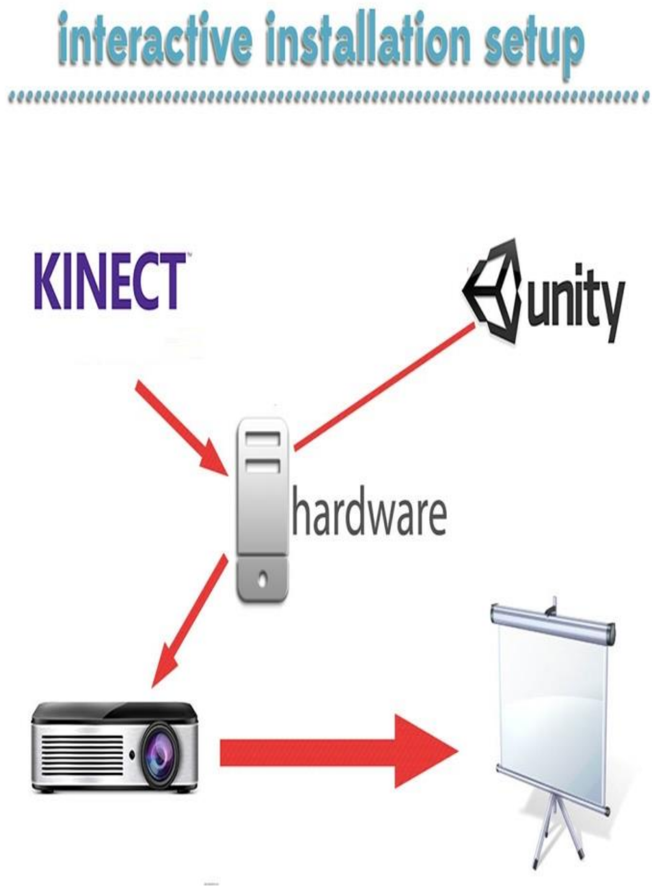


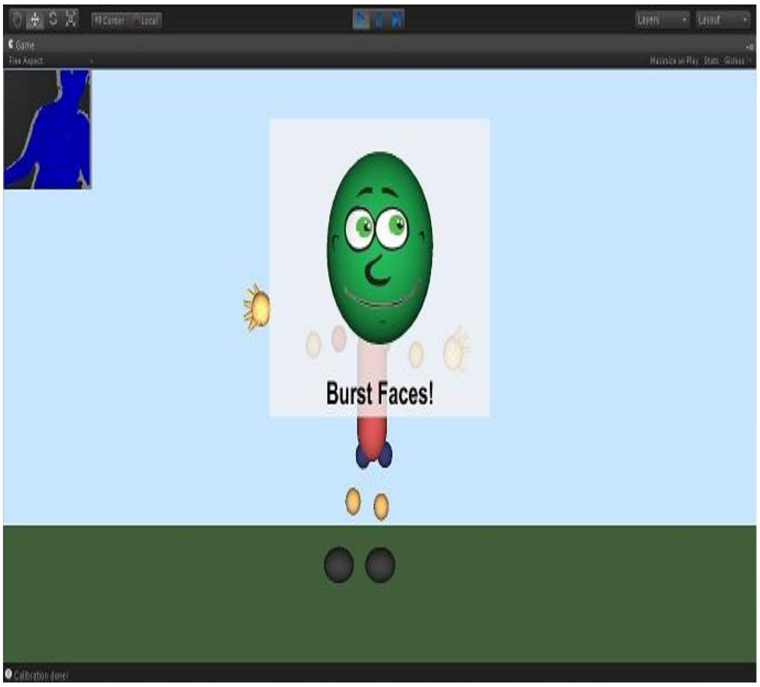
The image above shows the Scene Mode

The image above shows the Game Mode

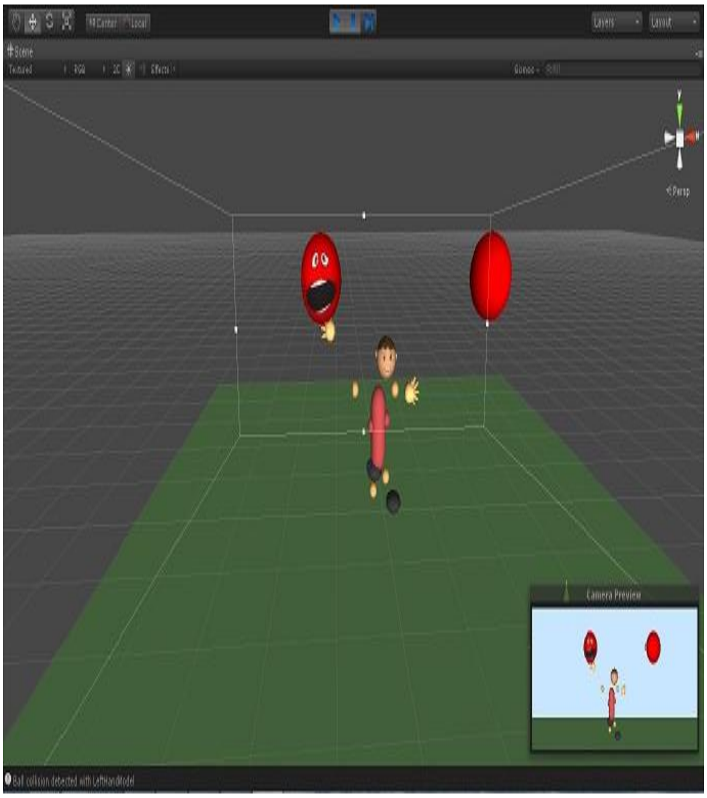
This are results based on the actual project created.

We made use of the based on RGB-D Images from the Kinect sensor and used for mapping it to the posture of the user. Unlike having a radar like in the other methods available. We have had 90% accuracy identifying the posture at 30fps at which Kinect runs.



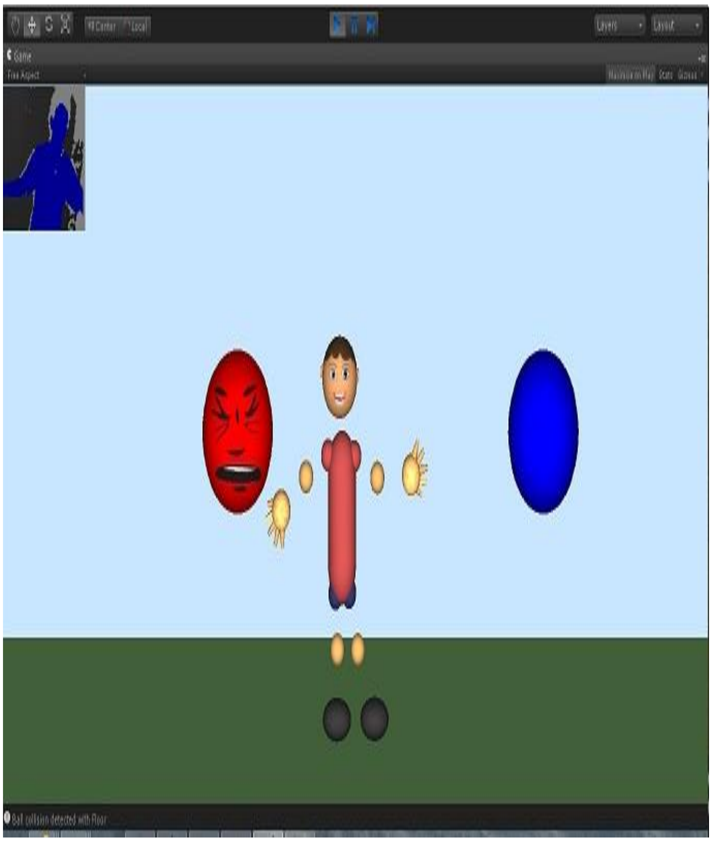


.



V. CONCLUSION

To conclude, the overall goal was to use skeleton information from Kinect for human posture recognition and develop a unity scene implementing it. Prior research into this topic have showed the implementation of posture recognition with unity. With the help of which we had 90% accuracy recognizing posture in synchronization with the system.



Future Research

Use a 3D model and improve animations over the objects, and also check Kinect compatibility in a scene implementing VRPN (Virtual Reality Peripheral Network) whose purpose is to provide a unified interface to input devices, like motion trackers or joystick controllers.

VI. References

[1] Zuniga, M., Incremental Learning of Events in Video using Reliable Information. 2008, Universite de Nice-Sophia Antipolis.

[2] Bernard, B., Human posture recognition for behaviour understanding. 2007, Universite de Nice-Sophia Antipolis.

[3] Le, Thi-Lan; Nguyen, Minh-Quoc; Nguyen, Thi-Thanh-Mai, "Human posture recognition using human skeleton provided by Kinect," Computing, Management and Telecommunications (ComManTel), 2013 International Conference on , vol., no., pp.340,345, 21-24 Jan. 2013

[4] Anton, D.; Goni, A.; Illarramendi, A.; Torres-Unda, J.J.; Seco, J., "KiReS: A Kinect-based telerehabilitation system," e-Health Networking, Applications & Services (Healthcom), 2013 IEEE 15th International Conference on , vol., no., pp.444,448, 9-12 Oct. 2013

[5] Luís Leite and Veronica Orvalho. 2012. Shape your body: control a virtual silhouette using body motion. In CHI '12 Extended Abstracts on Human Factors in Computing Systems (CHI EA '12).

[6] Hsu W. S., Ye T. Planning Character Motions for Shadow Play Animations. In Proc. CASA'05 (2005), 184-190.

[7] Bernard, B., Human posture recognition for behaviour understanding. 2007, Universite de Nice-Sophia Antipolis.

[8] COHEN, I. and H. LI, Inference of Human Postures by Classification of 3D Human Body Shape, in IEEE International Workshop on Analysis and Modeling of Faces and Gestures, ICCV 20 03. 2003.

[9] Mo, H.-C., J.-J. Leou, and C.-S. Lin, Human Behavior Analysis Using Multiple 2D Features and Multicategory Support Vector Machine, in MVA2009 IAPR Conference on Machine Vision Applications 2009: Yokohama, JAPAN.

[10] Gotsis, M.; Lympouridis, V.; Turpin, D.; Tasse, A.; Poulos, I.C.; Tucker, D.; Swider, M.; Thin, A.G.; Jordan-Marsh, M., "Mixed reality game prototypes for upper body exercise and rehabilitation," Virtual Reality Short Papers and Posters (VRW), 2012 IEEE , vol., no., pp.181,182, 4-8 March 2012

[11] Noyes,Mathew.A ,"Implementation of Headtracking and 3D Stereo with Unity and VRPN for Computer Simulations"2013, NASA Johnson Space Center; Houston, TX, United States.

[12] http://wiki.etc.cmu.edu/unity3d/index.php/Microsoft_Kinect_-_Microsoft_SDK#Integrating_with_Unity (Accessed April 2014 and regular for tutorials.)

[13] http://www.openni.org/resources/ (Accessed April 2014)

[14] http://zigfu.com/en/zdk/unity3d / (Accessed April 2014)

[15] M. Gotsis, "Games, Virtual Reality, and The Pursuit of Happiness," IEEE Comput. Graphics Appl., vol. 30, pp. 6-11, September/October 2009, 2009.

[16] B. Lange, Chien-Yen Chang, E. Suma, B. Newman, A. S. Rizzo and M. Bolas. Development and evaluation of low cost game-based balance rehabilitation tool using the microsoft kinect sensor. Presented at Engineering in Medicine and Biology Society, EMBC, 2011 Annual Int. Conf. of the IEEE. Boston, USA, pp. 1831-1834. 2011.

[17] Y. Chang, S. Chen and J. Huang. A kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities. Res. Dev. Disabil. 32(6), pp. 2566-2570. 2011.

[18] Jungong Han; Ling Shao; Dong Xu; Shotton, J., "Enhanced Computer Vision With Microsoft Kinect Sensor: A Review," Cybernetics, IEEE Transactions on , vol.43, no.5, pp.1318,1334, Oct. 2013 doi: 10.1109/TCYB.2013.2265378

[19] Z. Zhang "Microsoft Kinect sensor and its effect", IEEE Multimedia Mag., vol. 19, no. 2, pp.4 -10 2012

[20] I. Tashev "Recent advances in human-machine interfaces for gaming and entertainment", Int. J. Inform. Technol. Security, vol. 3, no. 3, pp.69 -76 2011.

[21] http://support.xbox.com/en-US/xbox-360/kinect/body-tracking-troubleshoot