HAL
open science

Tabletop games using real environment and physical simulation

Jérémy Laviole, Martin Hachet, Christophe Schlick

To cite this version:

Jérémy Laviole, Martin Hachet, Christophe Schlick. Tabletop games using real environment and physical simulation. FDG'11 - Foundations of Digital Games, Jun 2011, Bordeaux, France. hal-00661127

HAL Id: hal-00661127

https://inria.hal.science/hal-00661127v1

Submitted on 18 Jan 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Tabletop games using real environment and physical simulation *

Jeremy Laviole
INRIA - LaBRI - Bordeaux University
laviole@labri.fr

Martin Hachet
INRIA - LaBRI - Bordeaux University
hachet@labri.fr

Christophe Schlick
INRIA - LaBRI - Bordeaux University
schlick@labri.fr

ABSTRACT

The recent widespread of RGBD cameras such as Kinect® device from Microsoft® opens many new interaction metaphors available to the general public. In this poster we introduce a tabletop interaction metaphor using the depth map from the Kinect: each element on the table is included as a set of physical constraints in a virtual environment. We briefly discuss visualization methods and present a gaming genre adapted to this framework : action-construction games.

Categories and Subject Descriptors

H.5.1 [**Information interfaces and Presentation**]: Multimedia Information Systems—*Tabletop Interactions, Physics engines, Depth camera*

Keywords

depth camera, physics engine, virtual reality, video game

1. INTRODUCTION

1.1 Gaming trends and motivations

In the past years, immersive technologies have made a step towards the home with the emergence of innovative interaction devices for video games. For instance, the Nindendo Wii® has brought 3D gestures games as a standard in casual gaming. More recently, the Microsoft Kinect (using PrimeSense® Technology [8]) enables the players to get rid of the joypad, and to play "directly" by fullbody interaction. Among the new kinds of games offered by such interaction devices, games using physical simulations are very popular (eg. Bloom Blox on Nintendo Wii or Angry Birds® on mobile phones), the gameplay is generally about destroying or avoiding destruction of stacked objects. In this poster we propose a method to incorporate the real world as physical elements inside a virtual environment using a depth camera. The purpose is not to destroy but to guide the user into creating real world stacks of objects and movements.

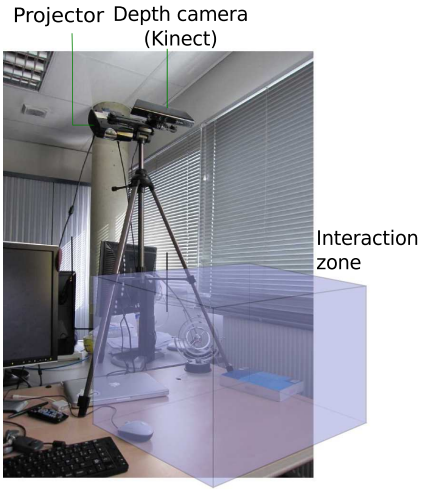

Figure 1: Global view of the system

1.2 Depth sensing games

Wilson created a mixed reality game [12], using the real table as a race area for virtual cars. In that game, the real world is just an elevation map, thus not providing direct human interactions with the virtual world. Later he added gestures using physics simulations[13] and the possibility to hold virtual objects into one's palm[14].

The work described in these papers is based on the use a depth camera to add more features on multitouch interactions. Other approaches of using the real world as a support for virtual scenes such as [2] allows the user to create its own environment overlaying reality. The 3D acquisition is done using structured light and the interactions with a stylus. Using tangible interfaces can improve the precision and the acceptation rate for some tabletop systems as in [4].

Recently many videos [3] [10] [11] present physical interactions using a Kinect, most of them use the body reconstruction. The integration of the depth and color information in the video game Minecraft [7] uses the same kind of input method as we present here. Based on this, we propose a gaming framework for a realtime game type called *action-construction* game. In this approach we focus on interacting in a large 3D space over the table (see figure 1).

*

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
FDG '11, June 29-July 1, Bordeaux, France.
Copyright 2011 ACM X-XXXXX-XX-X/XX/XX ...$10.00.

2. SYSTEM OVERVIEW

2.1 Acquiring depth information

The physical surface of a table is used as the virtual ground and the true interaction zone. To digitalize the table the user has to choose a set of points on the table from the depth view image as seen in figure 2. Once the set of points is selected the 3D plane of the table is fitted using least mean square. The plane fitting is better than selecting three points from plane though it is still a viable option. The view of the Kinect is displayed to let the user understand where he can interact, this view (displayed in white dots in figures 3 and 4) shows only 1/4th[1] of the points for performance and easier understanding.

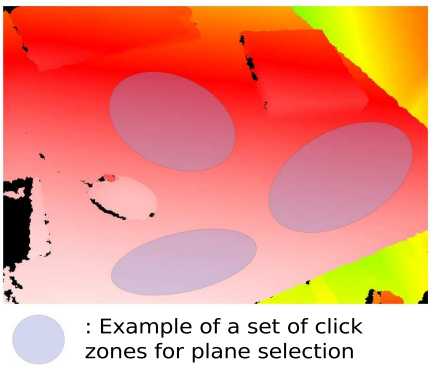

Figure 2: Depth image from the Kinect device using false colors to visualize the 10bit depth.

2.2 Physics engine

The physics engine has a fixed infinite plane (representing the table plane) and uses the Kinect information as cube constraints. As said previously only the information above the table are used, to do this, the distance and direction from each point to the plane are computed. Only the points above the plane and with a distance from 0.2 cm (to avoid to detect the plane itself) to 60 cm (or more if needed) to allow users to perform wide movements. Here again not all points are used, for performance issues, only about 1/12th of the sample elements are transposed into the camera space before being processed by the physics engine.

Each sample element generate a cube constraint that will be used by the physics engine to avoid mutual collisions. The size of each cube linearly depends on the depth of the sample element, and the scaling factor is manually set during an interactive preprocessing step. A straightforward extension would be to automatically adapt the size of the cube according to the distance and the number of sample elements.

2.3 Visualizations

In figure 3, we can see the user's hand in real world and the corresponding virtual interpretation. While white cubes represents the real world elements, the red cobbles are virtual. The interaction is currently limited to moving and handling the red cubes, however it is possible to extend these operations to enable the user to pick objects as in [13], or to manipulate or deform 3D objects.

Unlike usual applications based on the Kinect device, we do not perform user body tracking. So it is not obvious to automatically set the viewpoint. In our current implementation, the viewpoint is manually set during the preprocessing step, and is not changing during the whole game.

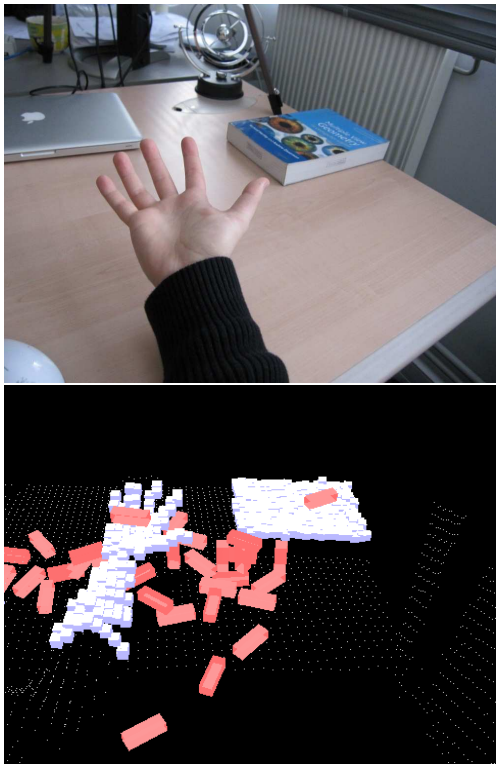

Figure 3: The virtual world ground is placed on the table. Here every object over the table is added in the physics engine. Virtual objects (in red) are on the table, over the book or into the user's palm.

For now, two kind of displays have been tested, a 24" in front of the user and a projector, projecting on the interaction table (see figure 4). In order to interact with the virtual world, the user can put anything on the table. As soon as it is detected by the depth camera[2], it is integrated inside the virtual environment. The view projected on the table is much more comfortable to use than the view on the computer screen. It is mainly due to the fact that the user has a better view of its hand while interacting with the virtual environment. Moreover using projection frees more space on the table and avoids depth camera occlusions.

3. POSSIBLE GAMES

To take a greater advantage of a very dynamic tabletop environment, party/puzzle/action games such as Wario Ware games [5] are well suited for this approach because it asks many different interactions.

It is possible to ask the user to stop, touch, move or avoid objects. In the context of scripted dynamic games,

[1]The resolution used is 640*480, the noise of the Kinect is approximately 0.2 cm if it is 90cm above the table.

[2]Transparent, semi-transparent and reflective objects are obviously problematic as infrared light is used.

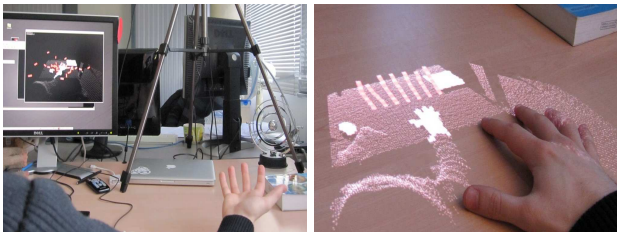

Figure 4: Left: Interaction using the screen for the virtual view. Right: The visualization is projected on the interaction space, but shifted from the most common interacting space to avoid overlapping issues.

the user can be asked to fetch objects from its environment to build real structures on the table and modify it according to the game script, that is the origin of the name "action-construction" (see figure 5). This kind of games offers various possibilites and it even allows the user to cheat: instead of contructing structures, he or she can simulate them with its own hands or with multiple people. Using the color it is also possible to create richer interactions, forcing the user to stack objects matching the desired colors. By constructing, demolishing, moving or handling objects this setup can provide an entertaining experience. It can be used to make "3D Twister® games".

4. IMPLEMENTATION DETAILS

The depth camera is a Microsoft Kinect (640*480 pixels 10bit depth camera), the screen for the tests is a standard 24" LCD screen. The projector used is a Samsung SP-P400BX projecting a 26" image on the table with a 800*600 resolution.

The whole software framework is built with Processing [1], enabling fast 3D programming and interacting. The 3D rendering is partially done using the library toxiclibs [3] and the physics engine used is jBullet [4], a java port of Bullet Physics Library. The kinect library used is *libfreenect* from the open source project OpenKinect[6].

Figure 5: Example of a gaming set. The user puts two boxes (green and blue) to guide incoming creatures (red). He can prevent jumping creatures from getting over the boxes with his arm (yellow).

[3]http://toxiclibs.org/
[4]http://jbullet.advel.cz/

5. CONCLUSION AND FUTURE WORK

In this poster we have presented an innovative interaction meta-phor for creative games. Users are free to use any object from their real environment to achieve winning the game, as soon as it is detectable with the Kinect. Possibilities are vast and the idea presented here is just an example of tabletop game using a depth camera.

Another interesting direction for future work would be to include augmented reality techniques (eg. [9]) in order to augment real objects with information coming from the virtual environment.

6. REFERENCES

[1] Processing. http://processing.org/about/, May 2010.

[2] B. Jones, R. Sodhi, R. Campbell, G. Garnett, and B. Bailey. Build your world and play in it: Interacting with surface particles on complex objects. In *Mixed and Augmented Reality (ISMAR), 2010 9th IEEE International Symposium on*, pages 165 –174, 2010.

[3] C. E. A. list. http://www.youtube.com/watch?v=7-HOc696g6s http://www-list.cea.fr/, Feb. 2011.

[4] A. Lucchi, P. Jermann, G. Zufferey, and P. Dillenbourg. An empirical evaluation of touch and tangible interfaces for tabletop displays. In *Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction*, pages 177–184. ACM, 2010.

[5] Nintendo. Wario ware presentation. http://www.nintendo.com/sites/software_warioware.jsp, May 2010.

[6] OpenKinect. http://openkinect.org, May 2011.

[7] M. Persson. http://www.minecraft.net/, May 2011.

[8] Primsense. Primesense home page. http://www.primesense.com, May 2011.

[9] R. Raskar, K. Low, and G. Welch. Shader lamps: Animating real objects with image-based illumination. Technical report, Chapel Hill, NC, USA, 2000.

[10] Tharsis. http://www.tharsis-software.com/ http://www.youtube.com/watch?v=20J0r76dQC0, Feb. 2011.

[11] N. Viniconis. http://www.orderofevents.com/MineCraft/KinectInfo.htm, May 2011.

[12] A. Wilson. Depth-sensing video cameras for 3d tangible tabletop interaction. 2007.

[13] A. Wilson. Simulating grasping behavior on an imaging interactive surface. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, pages 125–132. ACM, 2009.

[14] A. Wilson and H. Benko. Combining multiple depth cameras and projectors for interactions on, above and between surfaces. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 273–282. ACM, 2010.