VirtualTable: a projection augmented reality game

Dai Corso, Alessandro; Olsen, Mikkel Damgaard; Steenstrup, Kasper Hornbak; Wilm, Jakob; Jensen, Sebastian Hoppe Næsgaard; Paulsen, Rasmus Reinhold; Eiriksson, Eyþör Rúnar; Nielsen, Jannik Boll; Frisvad, Jeppe Revell; Einarsson, Gudmundur

Published in:
Proceedings of SIGGRAPH Asia 2015 Posters

Link to article, DOI:
10.1145/2820926.2820950

Publication date:
2015

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.
VirtualTable: a projection augmented reality game


Department of Applied Mathematics and Computer Science, Technical University of Denmark

VirtualTable is a projection augmented reality installation where users are engaged in an interactive tower defense game. The installation runs continuously and is designed to attract people to a table, which the game is projected onto. Any number of players can join the game for an optional period of time. The goal is to prevent the virtual stylized soot balls, spawning on one side of the table, from reaching the cheese. To stop them, the players can place any kind of virtual stylized soot balls, spawning on one side of the table, from which the game is projected onto. Any number of players can join the game and a projector homography. After the calibration, objects of any significant depth (at least 0.5 cm) can be recognized. The output of the depth camera is used to create a bounding box around the objects. We exclude objects that are connected to the border, to avoid recognizing the players’ hands. In Figure 1 (left) we see that the Kinect camera covers an area bigger than the game area, to not accidentally exclude objects lying on the border.

We transmit the identified bounding boxes to the actual game using a custom made protocol. In the virtual game, wall objects are invisible and affect only the behavior of the soot balls. We project a red glow around the towers to distinguish them and give a visual feedback on their range (see Figure 1 (right)). When we update the set of recognized boxes, we compare it with the existing set. Matching boxes have their position updated, interpolating it with their old position to avoid flickering. The remaining boxes are either added or removed to the game accordingly. Objects are distinguished only by shape: elongated objects are walls, square-like objects towers.

The behavior of the soot balls is simulated using Unity Engine’s built-in navigation system on navigation meshes. A tower, with a given frequency, shoots bullets to the soot balls within its range, removing them from the game.

The game explores the concept of augmented reality games, combining the tangible sensation of the pieces from board games and the immediate visual feedback from modern computer games.

Our approach

VirtualTable uses a computer unit attached to both a Kinect camera and a projector. We first process the input from the Kinect depth camera, then we pass it to the actual game to display the output.

The objects are recognized using the depth camera of the Kinect. We automatically calibrate our software once before the game is actually started, to estimate both a ground depth and the Kinect-camera covers an area bigger than the game area, to not accidentally exclude objects lying on the border.

We transmit the identified bounding boxes to the actual game using a custom made protocol. In the virtual game, wall objects are invisible and affect only the behavior of the soot balls. We project a red glow around the towers to distinguish them and give a visual feedback on their range (see Figure 1 (right)). When we update the set of recognized boxes, we compare it with the existing set. Matching boxes have their position updated, interpolating it with their old position to avoid flickering. The remaining boxes are either added or removed to the game accordingly. Objects are distinguished only by shape: elongated objects are walls, square-like objects towers.

The behavior of the soot balls is simulated using Unity Engine’s built-in navigation system on navigation meshes. A tower, with a given frequency, shoots bullets to the soot balls within its range, removing them from the game.

The game explores the concept of augmented reality games, combining the tangible sensation of the pieces from board games and the immediate visual feedback from modern computer games.

References


