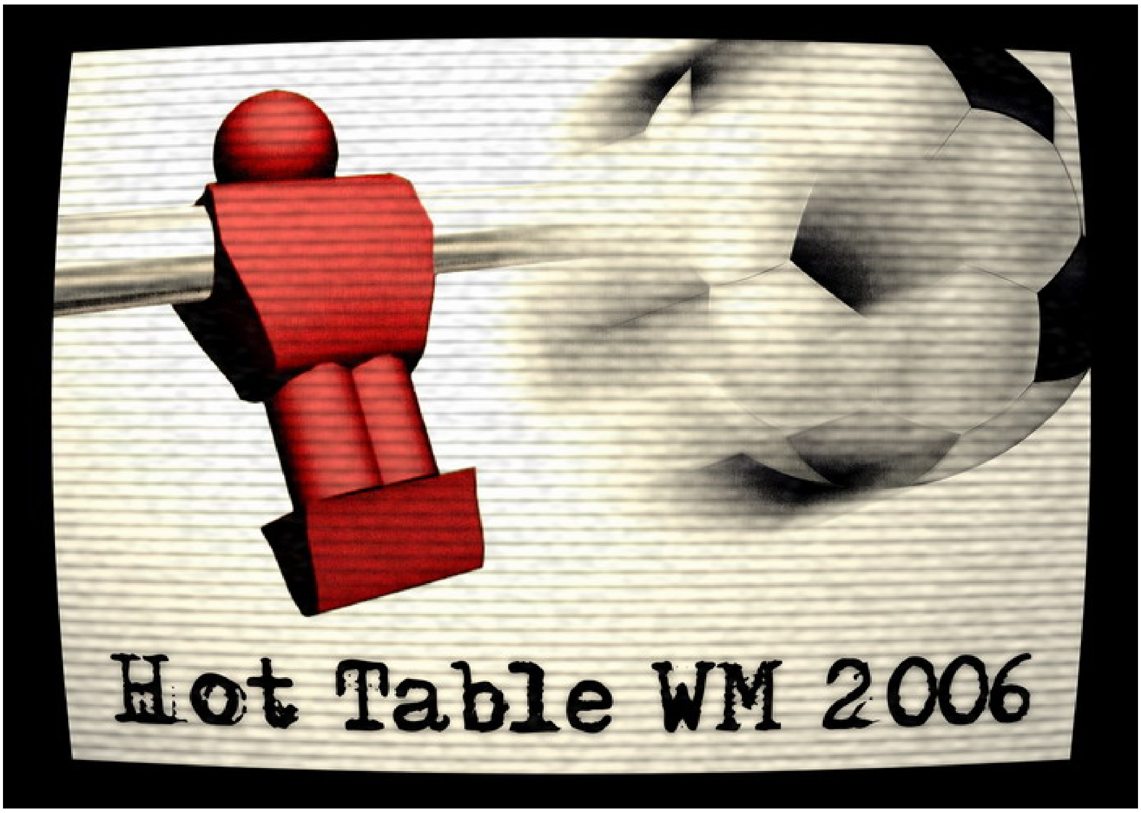
LU Computergraphik 2 SS 2.0, 186.124
LU Computergraphik 3 SS 2.0, 186.725

Abschnitt 3

Hot Table – WM 2006©

„Mitten drin statt nur dabei!“

Die Wuzlmeisterschaft© 2006



Trieb Nicholas
0009112 | 532 | nico@dietriebs.com
Gringinger Eduard
0126915 | 532 | e0126915@stud4.tuwien.ac.at

June 2005

Professor: Univ.Ass. Dipl.-Ing. Dipl.-Ing. Dr.techn. Michael Wimmer
Institut für Computergraphik und Algorithmen

Index

1. Plot

It’s 20:15, At the center of Schloss Schönbrunn...
The atmosphere is great and the location is fantastic. Hottable, is a special foosball game, bringing in more and more players from around the world. It’s an exhibition about fame and glory...

Hot Table – WM 2006
Play for the 'WuzelMeisterschaft'
www.hottable.at.tf

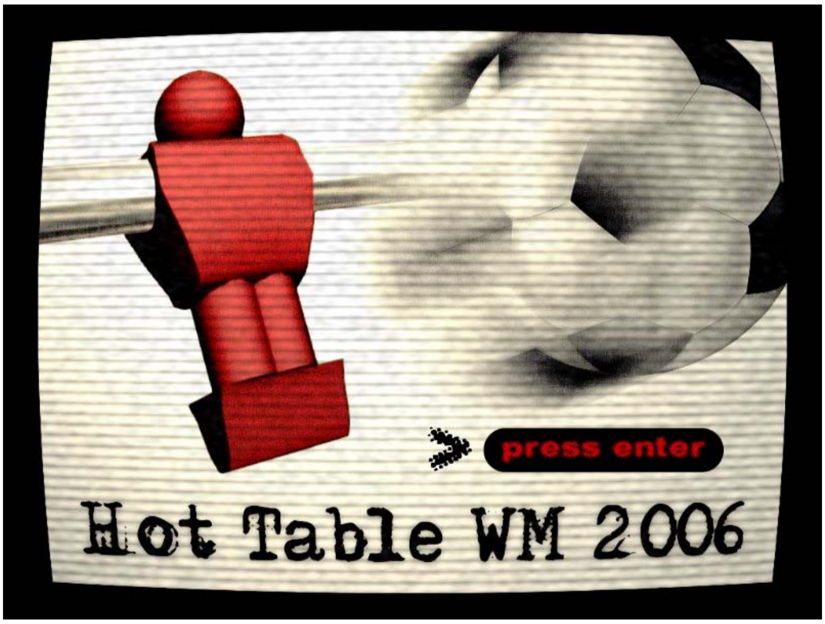
2. Description of the Game

„Hot Table – Die WuzlMeisterschaft 2006“ is an action-sports-game. It starts in the middle of Schloss Schönbrunn. After a camera flight the player will start with camera option 1, which can be chosen by the keys 1, 2 or 3. Or for free view press the left mouse button and move around. The scoreboard on the top will give you an overview about the result. For the controls you can press F1 for help or look at chapter 7. We also implemented some arrows which lets you know which players are being controlled (especially important for those who wanted to play in the manual mode, which is more difficult to play).

3. Rules / Goal of the game

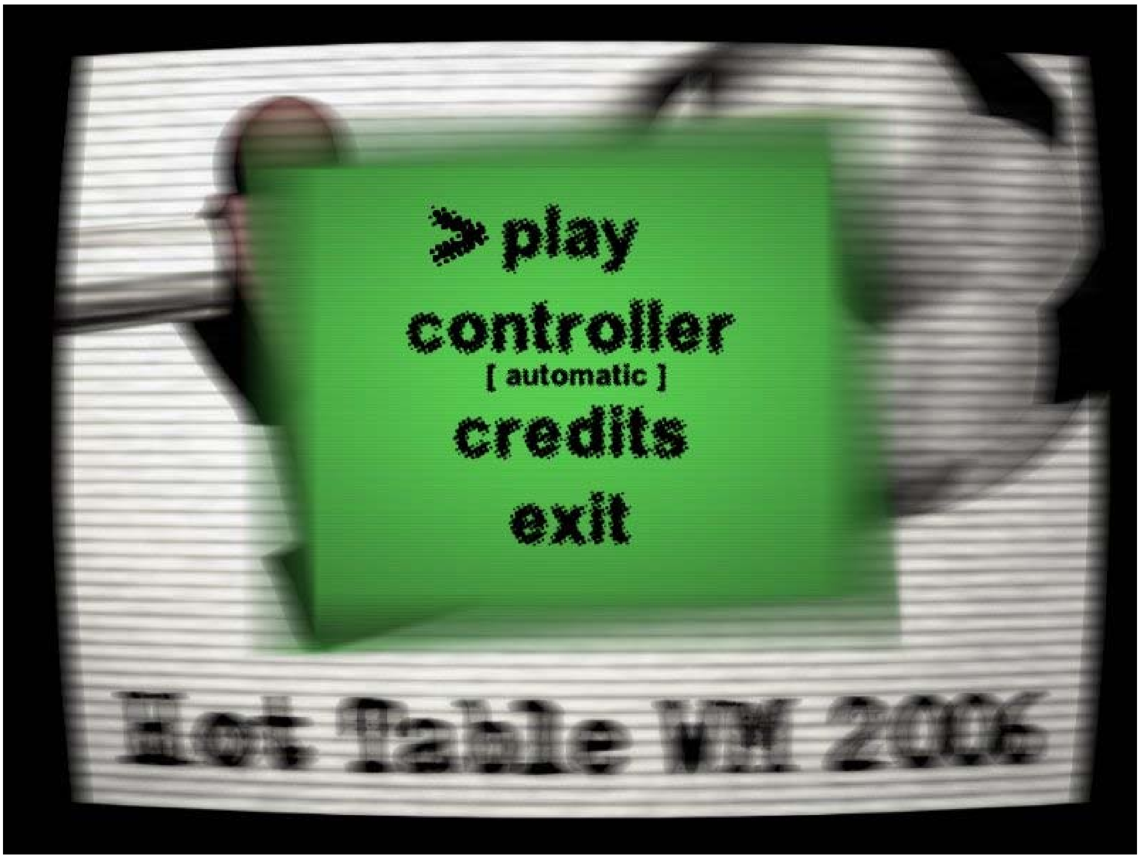
The first Player to shoot 5 goals before his opponent will win the game. This will be indicated by a window, which contains the player that has won the match. You’ll have to press ENTER to get back to the menu. There you can start the next game.

4. Starting the game



In order to start the game, all you need to do is go to the 'bin' directory and double click 'HotTable.exe'. To start the game, all you need to do is to press ENTER and then go through the menu.

5. Menu



We created our own pictures for the Menu and used the same glBlendFunc to display the font and the arrows. We also used sounds for the atmosphere. In the Menu you can click through the options with the arrow keys, and the enter button. The Menu consists of four options: "PLAY" "CONTROLLER" "CREDITS" and "EXIT"

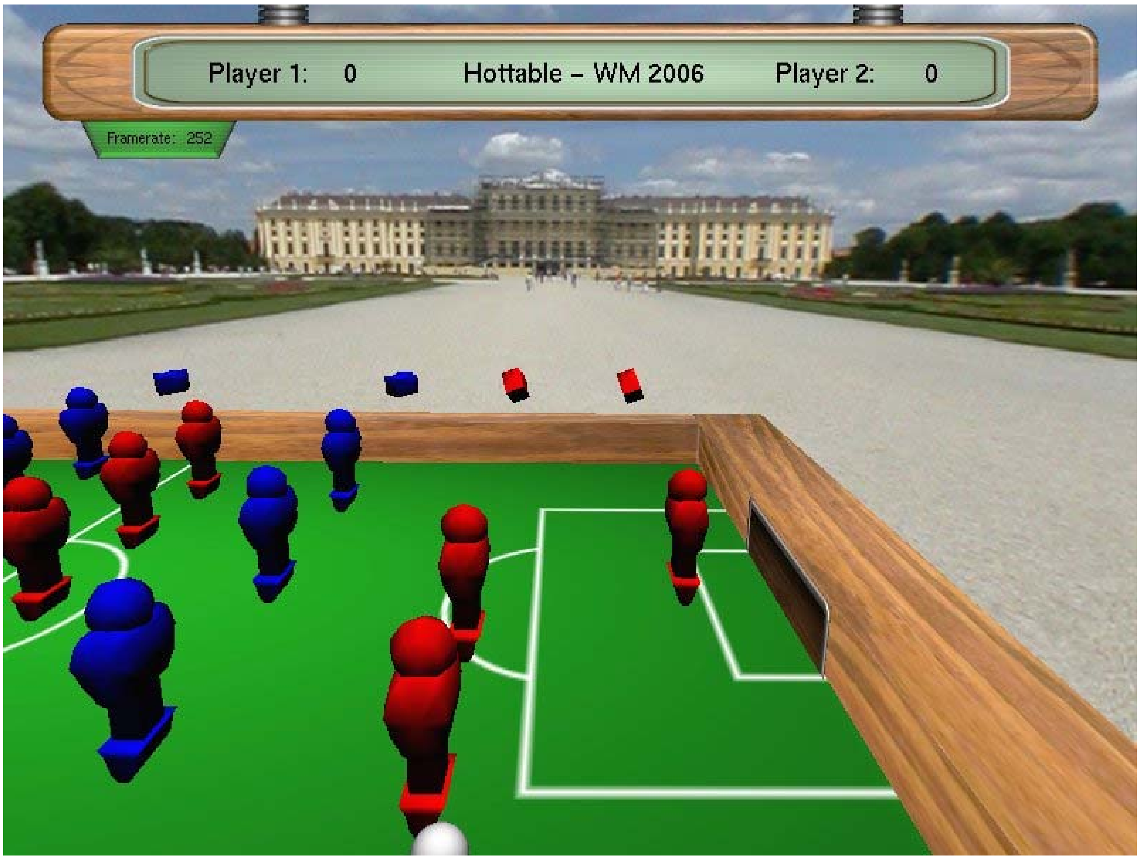
"PLAY" starts the game.

"CONTROLLER" allows you to change between "automatic" and "manual" controlling of the players. "Automatic" controls automatically switches between the players depending on the position of the ball. It jumps between the goalie and the defenders to the midfielders and attackers. The "manual" controls allows you to choose to switch using the "q" or "7" button for player1 and player2 respectively.

"CREDITS" opens the credits picture, where you can get back to the main menu with the enter button.

"EXIT" speaks for itself...

6. Game play



The Game starts off with a camera flying in from above down towards the field. As a background we wanted a stadium skybox, but were unable to get the permission to take the picture in any of the stadiums. Therefore we integrated a beautiful view of Schönbrunn. You can skip the fly through by pressing ENTER. With F1 you can look at the help display for the controls...

7. Controls

Since there is no AI player implemented, we created a two player game. The game needs two pair of hands which are played simultaneously on the keyboard.

ENTER/SPACEBAR: Start the game and make the ball roll if it gets stuck.

FOR PLAYER 1:

A: Stop ball/shoot/pass
D: Stop ball/shoot/pass
W: Move up
S: Move down
Q: Switch players

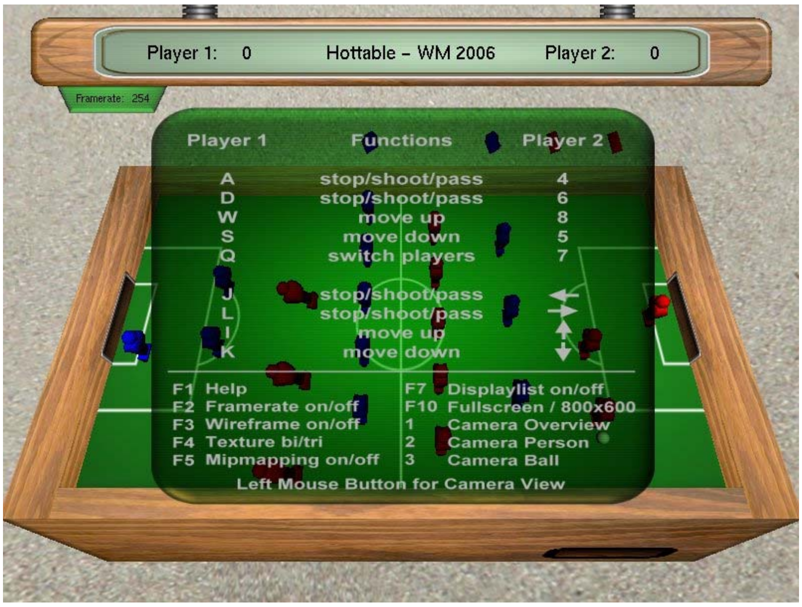
J: Stop ball/shoot/pass
L: Stop ball/shoot/pass
I: Move up
K: Move down

FOR PLAYER 2:

Arrow key left: Stop ball/shoot/pass
Arrow key right: Stop ball/shoot/pass
Arrow key up: Move up
Arrow key down: Move down

On the num pad:
4: Stop ball/shoot/pass
6: Stop ball/shoot/pass
8: Move up
5: Move down
7: Switch player

The game starts with the controls of the midfielders and strikers. If the controls are set to Manual you can switch between the players by pressing either 'E' or '7'. The button pretty much jumps you between the back two and the front two. The shoot/stop buttons, allow you to slow down the ball or even stop it. If the ball is moving towards a player, and you player is presses away from the moving ball, it can slow down the ball and even stop it. By letting go of the button you shoot the ball. You can also pass the ball with less speed than the shot, by simply pressing forward towards the ball. This allows you to pass between the players with a little more control.



Player 1	Functions	Player 2
A	stop/shoot/pass	4
D	stop/shoot/pass	6
W	move up	8
S	move down	5
Q	switch players	7
J	stop/shoot/pass	←
L	stop/shoot/pass	→
I	move up	↑
K	move down	↓

F1	Help	F7	Displaylist on/off
F2	Framerate on/off	F10	Fullscreen / 800x600
F3	Wireframe on/off	1	Camera Overview
F4	Texture bi/tri	2	Camera Person
F5	Mipmapping on/off	3	Camera Ball

Left Mouse Button for Camera View

Further controls include...

ESC:	Menu/Exit	
F1:	Help	//Will give an overview about the buttons.
F2:	Framerate on/off	//Display framerate or not
F3:	Wireframe/rendering	//Allows you to change to the wire frame mode.
F4:	Texture billinear/trillinear	//Changes between bilinear and trilinear texturing
F5:	MipMapping on/off	//Turns mipmapping on and off
F7:	Displaylist on/off	//speed up or slow down (nvidiaspeedstar;)
F10:	Fullscreen/Window mode	//Allows you to change between fullscreen and window modes

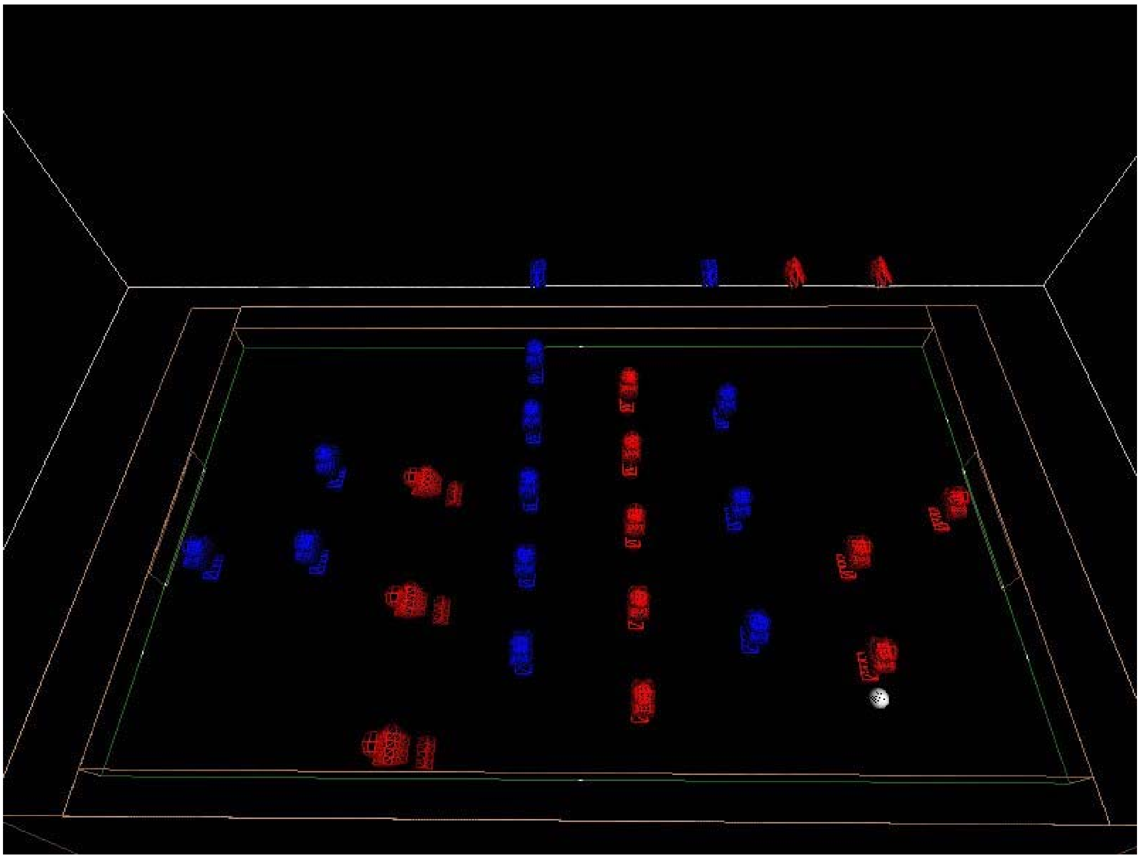
NUMPAD

1:	Camera Overview
2:	Camera Person
3:	Camera Ball

MOUSE
To select your own view, press the left mouse button and move...

8. Modeling Tools

We used 3d Studio max to create the complex models. We also used Cool 3d to optimize the meshes. In order to lower the processing time, we also hand changed parts of the models that were too detailed.



9. Texture

We load our Windows Bitmap separate into each class. Only BMP files could be loaded with this class. As Tutorial we used http://nehe.gamedev.net especially lesson 45. First it opens the filename in read only binary mode and reads out the bitmap file header. After reading it checks if it is a bmp or not (with an id check). Setting the file pointer to the beginning of the bitmap data it allocates the data and so the image data could be read. Then it swaps the R and B values to get RGB because the bitmap color is in BGR. After closing the file we build the texture. 'glBindTexture(GL_TEXTURE_2D, ID);' binds the texture to a texture object. Then it filters if the texture is bigger or smaller as it should be. For the skybox we used Mipmapping to load the cube.

10. Sound (FMod)

We implemented sound with the FMOD-library because it allowed us to use different file-formats (such as wav or mp3) and its easy to use (more details on http://www.fmod.org/). For the Fmod-soundsystem it is necessary to link the fmodvc.lib and include the following header-files 'fmod.h', 'fmod_errors.h' and 'wincompat.h' and don't forget the right 'fmod.dll' file (version 3.7.4). For the sound we created our own class file in order to keep our main class slim. InitFMOD() initializes the sound board and FSOUND_Sample_Load loads some sound files. More then 80 soundfiles (all recorded by or own) will give you a real atmosphere.

We also added commentary to the game itself, using the random function to play different files, and not always the same ones. For example you can hear the world famous quote from Edi Finger after you score goals "TOR!! TOR!!...". We've different commentary-sounds for defense, own goal, introduction, shot, goal, winner and even background information to the game itself. We also implement sounds for the Menu (especially the blur song 'song2') and for the camera animation at the start of the game.

11. Skybox

For the pictures (panoramas), we have to thank the people from the IGW for lending us the fisheye camera, which allowed us to create our own cube-pictures for the skybox. Therefore we used the demo of Panoweaver 4 to stich/defish and bring it to a cube-form. For the implementation we used http://home.planet.nl/~monstrous/skybox.html for help.

12. Environment

glBlendFunc defines the operation when blending is enabled. We have to glEnable and glDisable with argument GL_BLEND to enable and disable blending. Transparency is best implemented using blend function (GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA) with primitives sorted from farthest to nearest. Blend function (GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA) is also useful for rendering antialiased points and lines in arbitrary order. So we build our environment, which contains the scoreboard, the help-window, the information-windows (for mipmapping on/off, texture bi/tri and displaylist on/off) and the frame-window.

13. Collision Detection and Response

Our collision detection was hand written by our own engineering team (Nicholas Trieb, and Eduard Gringinger). We were able to update the collision detection and response to a new and improved version. In general, our collision system is built with boundary boxes around the players, and the response is dependent on where it hits the player, and what movement the player is acting out.

14. Objects

Table (texture)

Height: 89 cm (86,5 cm middle of players tick)
Length: 140 cm
Width: 76cm (125 cm, with sticks)
Game field: 120 x 70,5 cm
Goal width: 20,5cm
Goal height: 6,5cm

Player (3ds model)

There are 22 players which can be displayed with or without a displaylist, but be carefully because of your fps (check with F2)

Ball (sphere)

Ball color is white, with a diameter of 33mm and a weight of 17 Gram.

Scoreboard (texture, alpha-blending used)

Here you can see the result of your gaming.

Arrows (3ds model)

The arrows help you detect which players you controlling at the time.