Momentum

Group members: Bernhard Buss, Claudia Kuster, Jürgen Fornaro

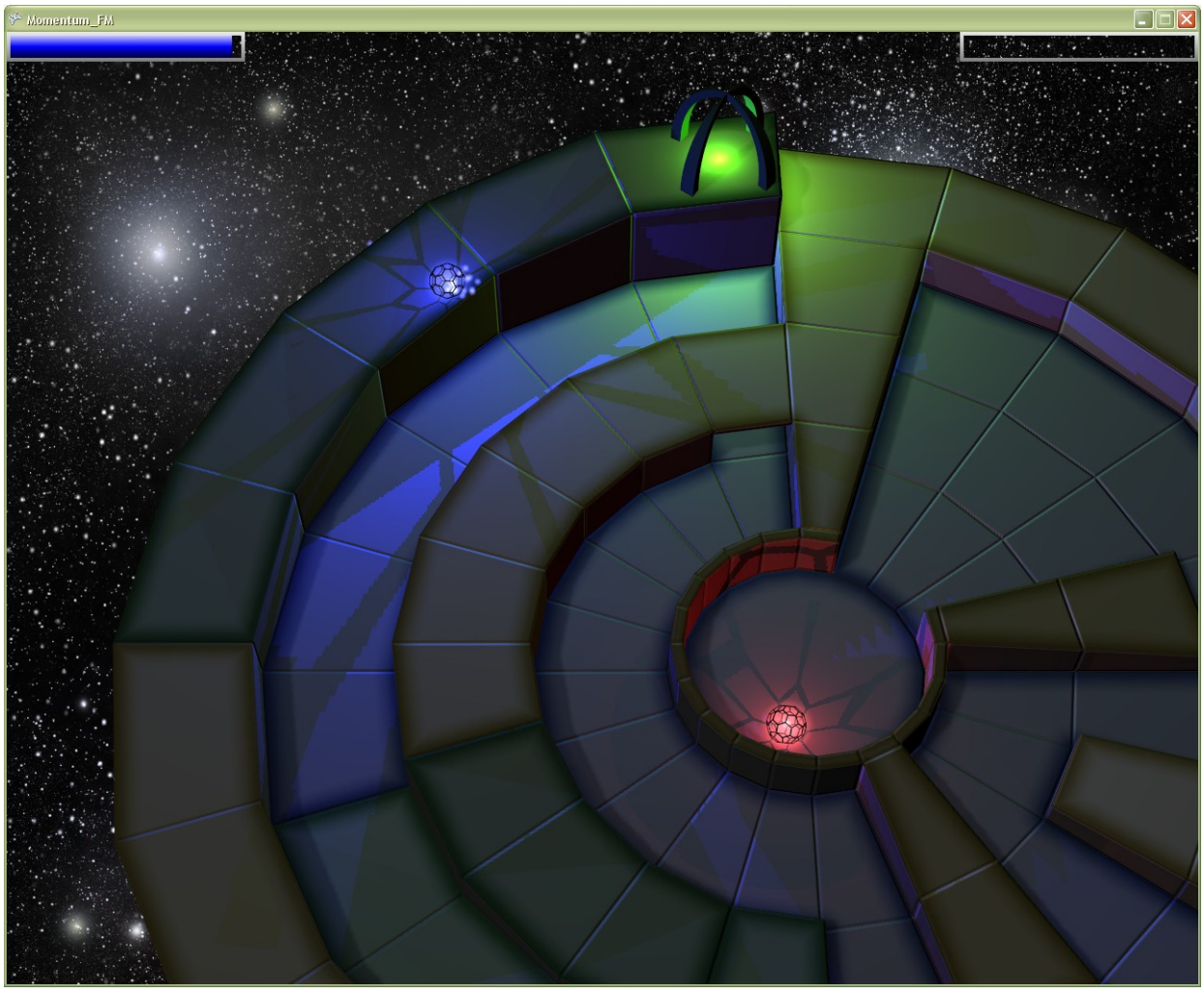

Illustration 1: Momentum, the last steps to the finish...

This document describes developments since our in-class demonstration of the Low Target stage. But first a summary of the game play is presented.

1. Summary of Game Play

Momentum is inspired by the original board game where you have to balance a wooden board in order to move a metal ball from one end of a labyrinth to another, avoiding the holes along the path.

The game provides two different game modes:

Classic game mode:

Only single player mode.

Simulates the original game play on the wooden board. The player controls a metal ball by balancing the board with the game pad. If the ball falls off the board, the game is lost. See illustration 2.

Illustration 2: Classic mode, wooden board and metal ball. The board is rotated by the player.

Momentum game mode:

Single player or multi-player modes (up to 4 players, single screen, see illustration 1). The player directly controls his ball (a spherical cage containing a quantum particle) with the game pad, accelerating it in the desired direction (the board is fix). The

particle gains momentum if the player is able to navigate the ball smoothly through the labyrinth on the board, staying off the obstacles and the walls. The player can release the momentum of his quantum particle to perform special moves such as speed-ups, jumps or pushing an opponent off the board. The player that reaches the other end of the labyrinth, the so-called warp core matrix portal, wins.

An important feature of our game is universal (space-like) physics and rendering engines, enabling the construction of diverse game boards by simply importing a geometric model that was built using external software (e.g. Blender or Maya, in our case the software used for 3D content creation).

2. Progress of Implementation

Since our in-class demonstration of the Low Target stage we have managed to implement all features of the Desired target stage. We have already done some preliminary play-testing during the past few weeks. This showed for example the need for starter game levels for new players which we complemented with info screens, where the player is introduced to the main ideas of the game play step by step. We are now well prepared to do further play-testing.

In addition to that we managed to get our game running on the XBOX. The problems with the Custom Model Processor and the creation of the collision bounding hierarchy are resolved, but we are now facing performance issues regarding the collision detection with the particles, because of the slower CPU of the XBOX (see section 3 „Next Targets of Development“ on how we plan to resolve those issues). However, graphics on the XBOX are very smooth, and we do not have problems with shader or shadow effects.

Next follows an informal list of new features we implemented for the alpha release of our game.

Menus / Info Screens:

The menu is kept simple and is actually not a traditionally styled menu with entries to choose from. Instead, the player has to navigate through the screens by pressing controller buttons which correspond to well outlined actions (see illustration 3 and 4). This leads to a more intuitive and faster navigation, especially for non-gamers and young children. The info screens shown before a level starts are meant to give new players an introductory help to the game and on how to use the controller.

Illustration 4: The main menu showing the two game modes.

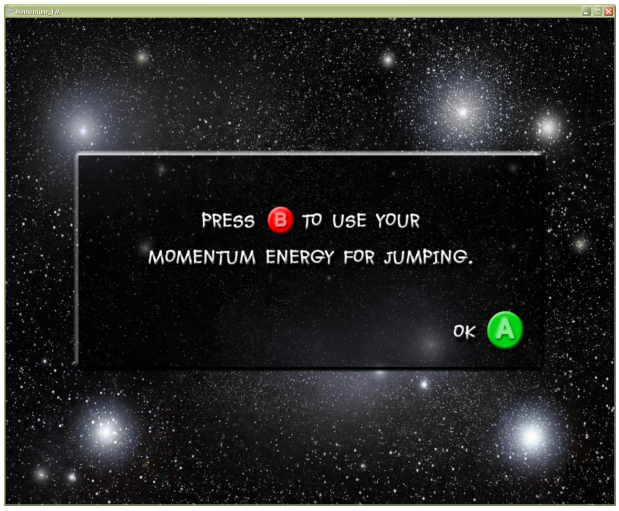

Illustration 3: An Info Screen showing helpful information for beginners.

Game Boards:

All game boards have specifically been enhanced with new textures and normal maps to give each one an individual and more plastic look. Also, the textures help convey information to the player on the direction towards the finish, and to better visualize ramps and walls. The normal maps make the boards look even more 3D, and help to increase the appeal also to observers of the game. See illustration 5 for an example of texturing.

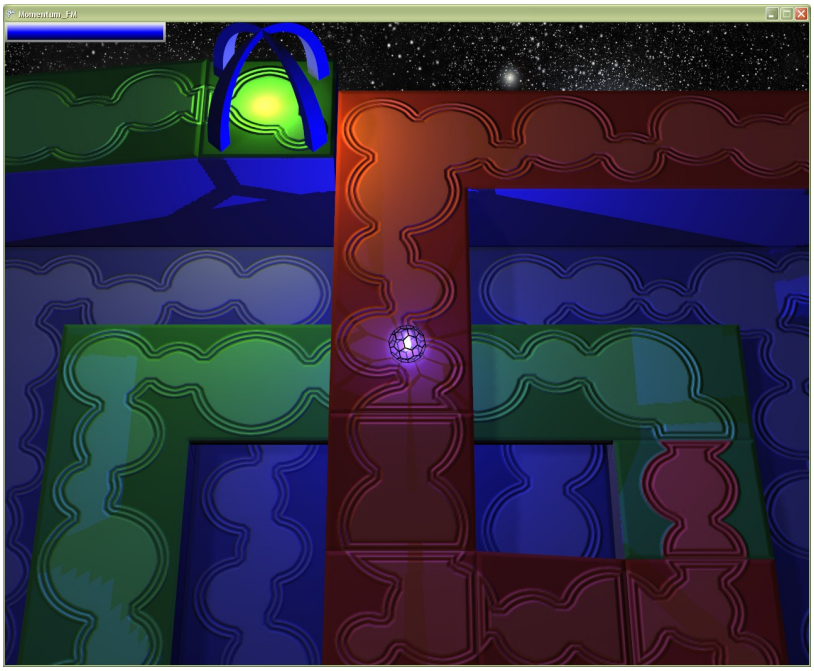

Illustration 5: Game board with individual texture and normal map, showing the path to the finish.

Particle Simulation:

Each ball has a particle system which creates a certain number of particles depending on the amount of momentum energy the player has accumulated. The particles orbit around the center of the ball because of a virtual gravity in the center (see illustration 6). If the ball undergoes a higher acceleration than the gravity provides, e.g. when hitting a wall, the particles fly out of the ball cage, collide with the level, bounce off, and eventually return towards the center of the ball. Particles that move away too far are left behind. They will rest on the floor and will fade out after a short time. Here an idea to yet implement is, that other players can gather these particles to gain momentum.

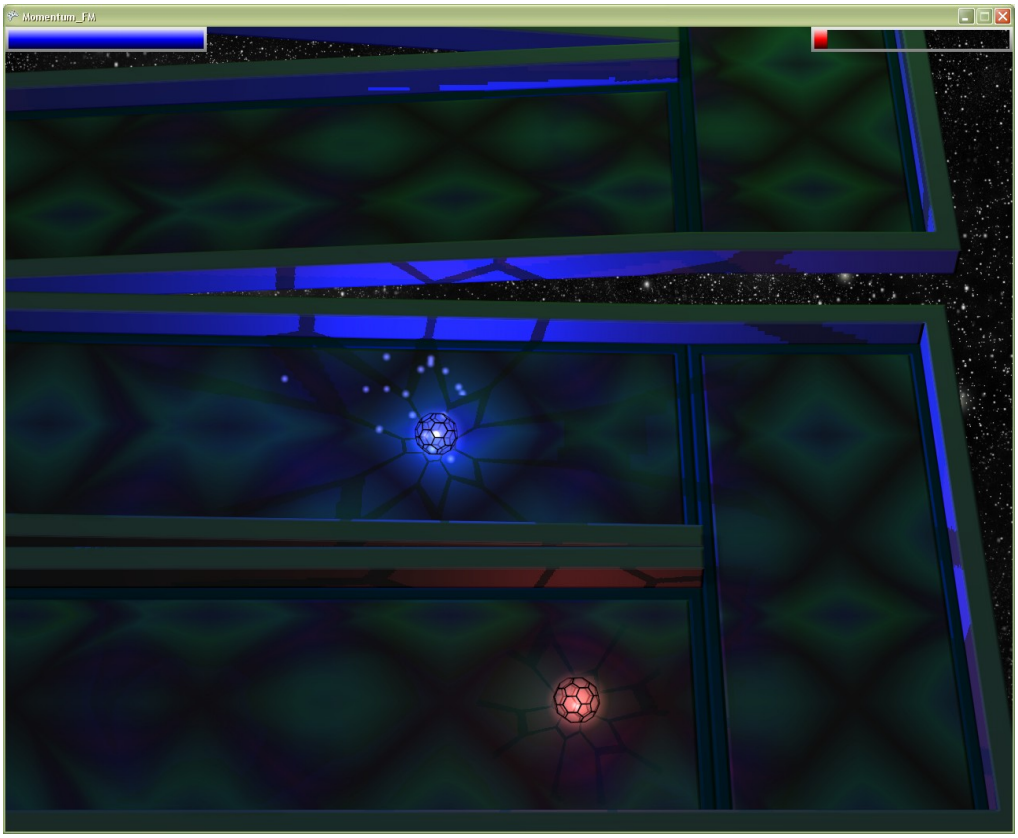

Illustration 6: The blue player has gathered full momentum energy, while the red player has nearly none. The blue orbiting particles and the brighter light further visualize this.

Lighting / Shadowing:

We have added our own shader code to add shadowing from the global light (momentum and classic modes) and from the point lights of the balls which better visualize the concept of momentum energy (only momentum mode).

For the global light we implemented shadowing by the shadow mapping technique, replacing the previous stencil-buffer implementation. Shadows from the global light are anti-aliased and smoothed (see illustration 7).

For visualizing momentum energy inside the game balls we put a point light at the center of each game ball. The light gets brighter with increasing momentum energy and casts attractive shadows on the scene resulting from the wireframe cage hull of the ball. Shadowing for these ball lights is implemented by a shadow mapping box inside each ball (see illustration 7). For performance reasons the shadows cast from the ball lights are not anti-aliased.

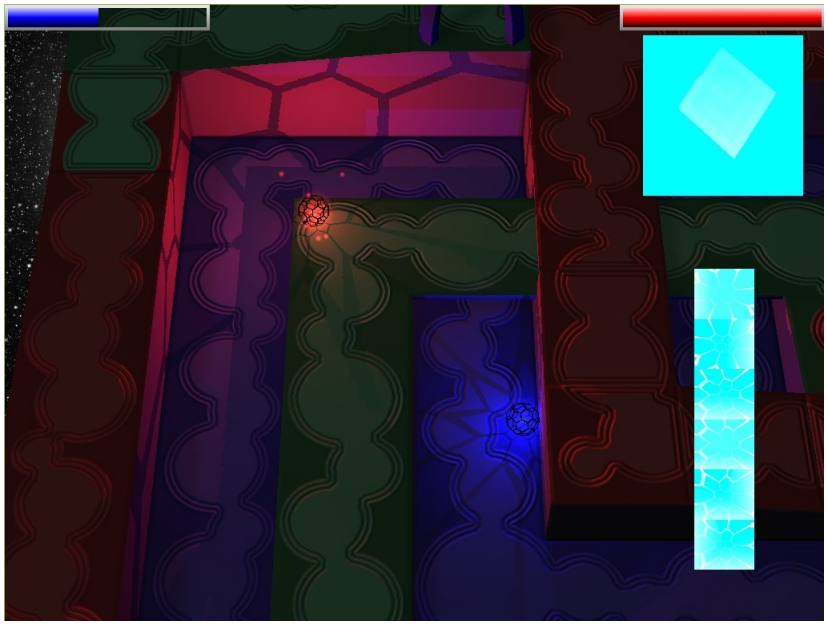

Illustration 7: Shadow map of global light on upper right, 6 sides of shadow map box of blue ball on lower right. Red ball with higher momentum energy casts brighter light and shows orbiting particles visualizing energy.

Audio Engine:

We used a trial version of MAGIX Music Maker (www.magix.com) to create three different background music tracks for momentum mode, one track for classic mode, as well as background music for the menu and ranking screens. Furthermore there are sounds for different events in the game such as balls bumping into walls or into each other, and a player finishing the level.

The three background musics for the momentum mode are manually fitted to the starting process of each level, such that the usual countdown of „3-2-1-GO“ is replaced by the pace of the intro and the individual „bang“ effect at the third second of the music tracks. Together with the zooming in of the camera to the balls starting positions and the enabling of the balls point lights, this gives a clear impression that the game is now starting, calling the player for action! Naturally, as it is with music, this is not perceived actively, but we

believe the music significantly enhances the player's game impression.
One thing to worry about and to further improve is to avoid repetitiveness of music and sound effects.

3. Next Targets of Development

Play-testing will provide us with feedback of new unskilled players. Their opinion will mostly influence our next steps of development.

Deployment on Xbox: We managed to deploy our game on the XBox but have to tackle some performance issues, namely with the collision detection for all particles. In order to improve the situation, we are looking into following possibilities:

- parallelizing the computations of the particle collisions using the third core of the XBOX CPU,
- distributing the computations of the particle collisions over several update steps,
- calculating the collision over several time steps by extrapolating the particle movement,
- using specific particles that do not require collision detection (resting on the board, staying inside the ball).

We are confident that already the first two solutions together will resolve this performance issue and allow to use 100 particles per ball on the XBOX resulting in a total of 400 collision detections, which is currently working on a PC with a Intel Core 2 Duo 2.4+Ghz CPU.

Further detail and finetuning the interaction of the players resulting from varying amounts of momentum energy. For example one player might steal momentum energy from another player by collecting its particles, which the latter lost when hitting an obstacle on the game board or during movement on the board, when the particles get too far away of their origin.

Implement features of High Target stage as far as possible in time. E.g. to better balance the levels and their paths to the finish, we could add collectable items which will add points to the player's score.