Snow Board Game

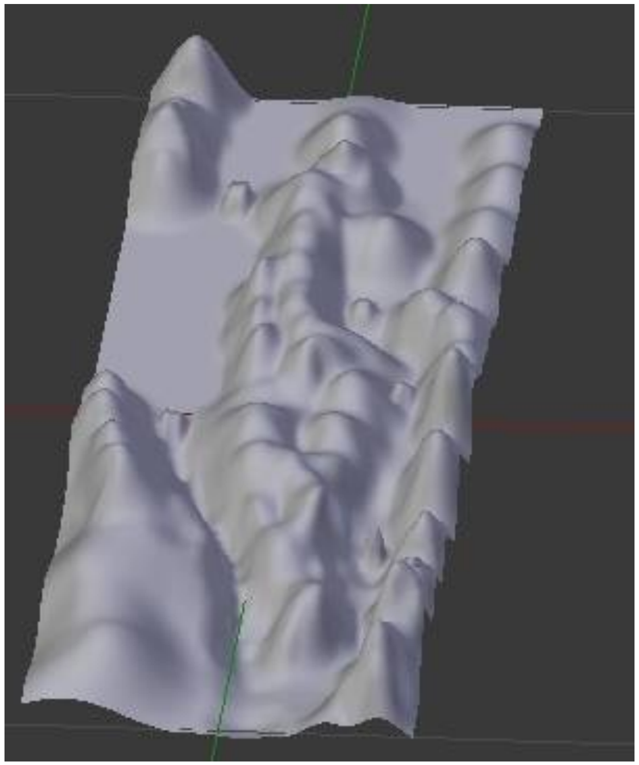
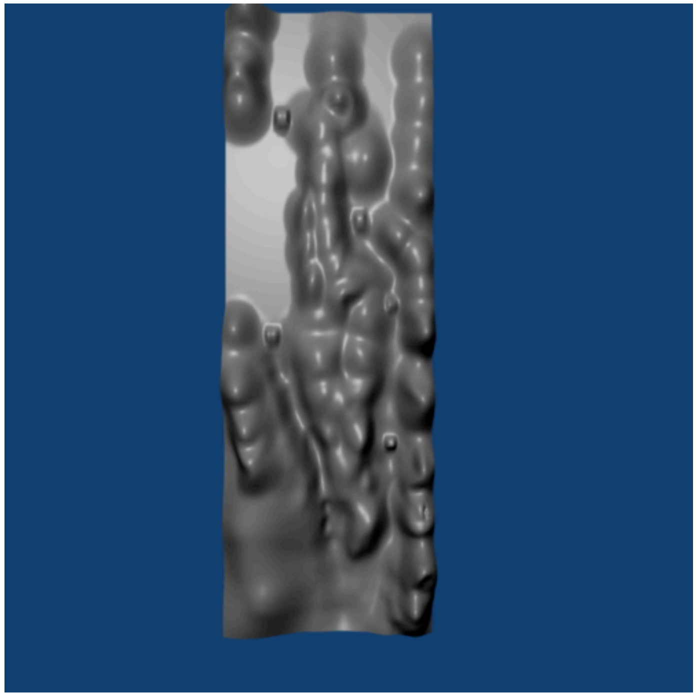
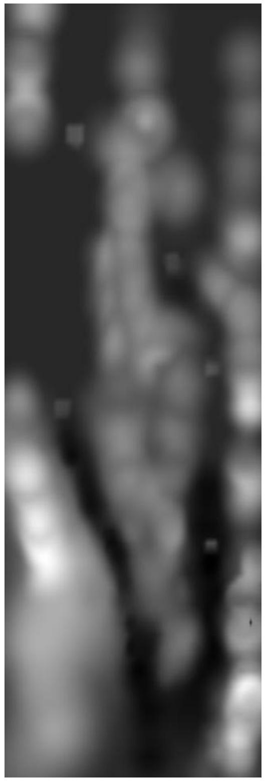
A presentation on the development of a VR Game by Group B

Visual Computing Institute

RWTH AACHEN UNIVERSITY

Content Creation – Map

- Most important object of the game is the map
- Was first modeled using Blender then exported into a height map
- Height map is an image representation of the map
- Has many advantages concering game logic and game play

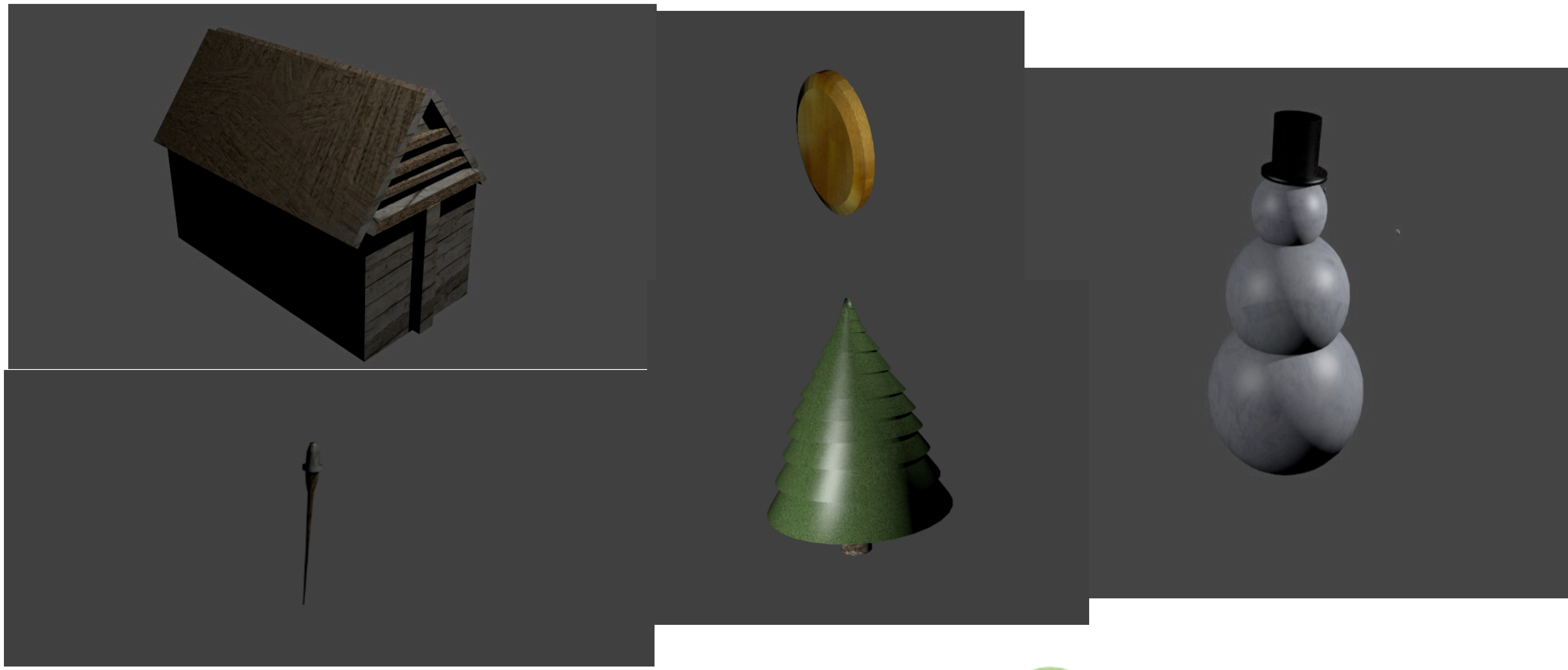




Visual Computing Institute

RWTH AACHEN UNIVERSITY

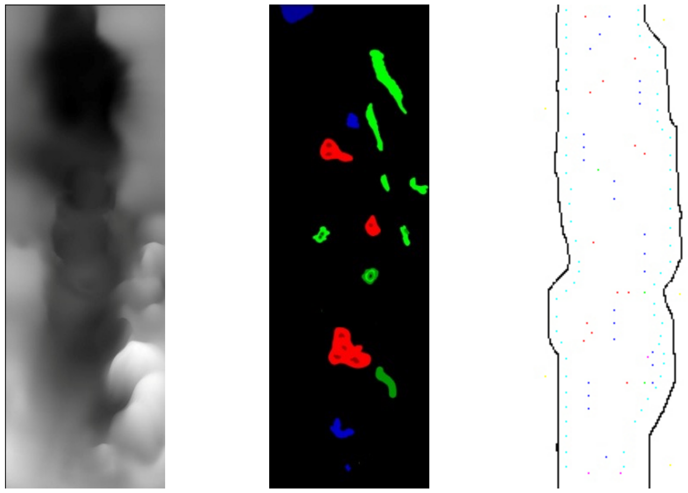
Content Creation - Objects

- Created Objects: Power-Ups, coins and obstacles
- All created using Blender
- Texture painting „by hand“ per brush tool
- Textures exported separately in image file



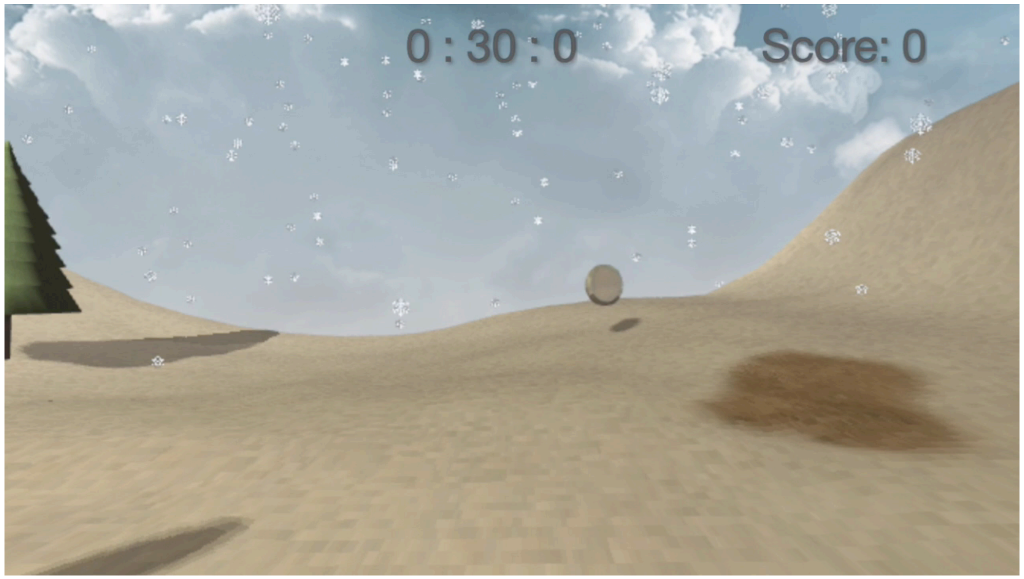
Map Generation

- Extract heights form height map onto vertex grid
- Indexed vertex buffer for rendering
- Blend map for textures
  - Passed to the shader for texturing
  - Saving information for area detection
- Entity map for objects
  - Setting of objects on the map
  - Saving information for hit detection



Map switching

- Two maps loaded at a time
- Loading of new map in a separate thread
- Endless map chaining possible
- Two switching modes
  - Cyclic map switching
  - Random map switching

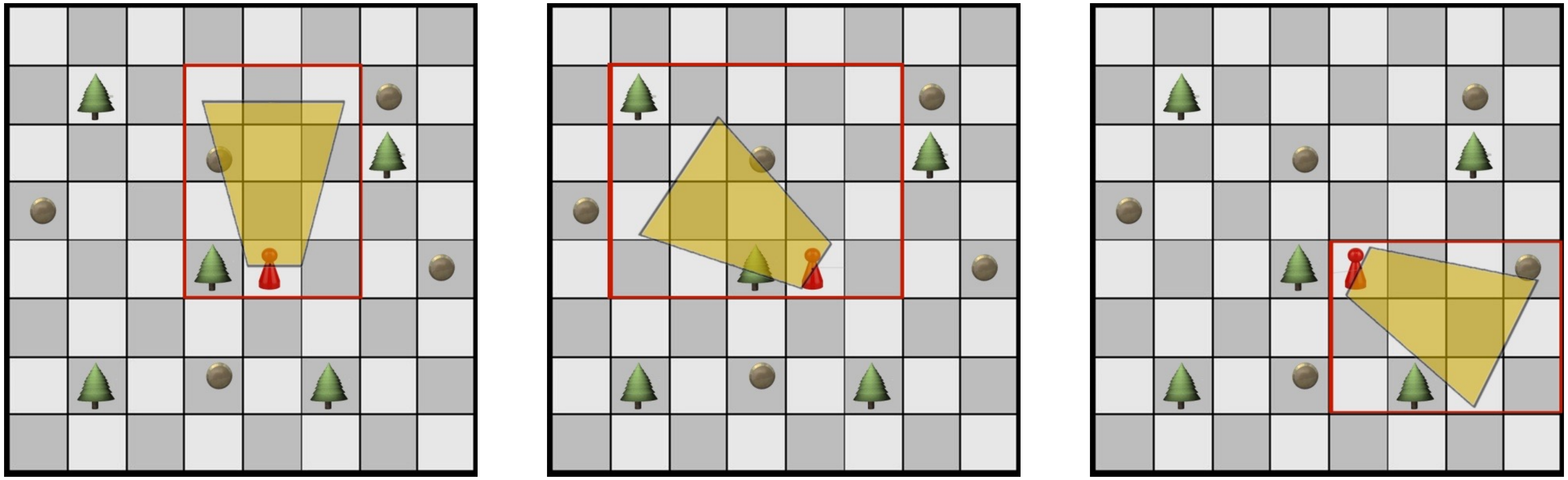


Movement / Area Detection

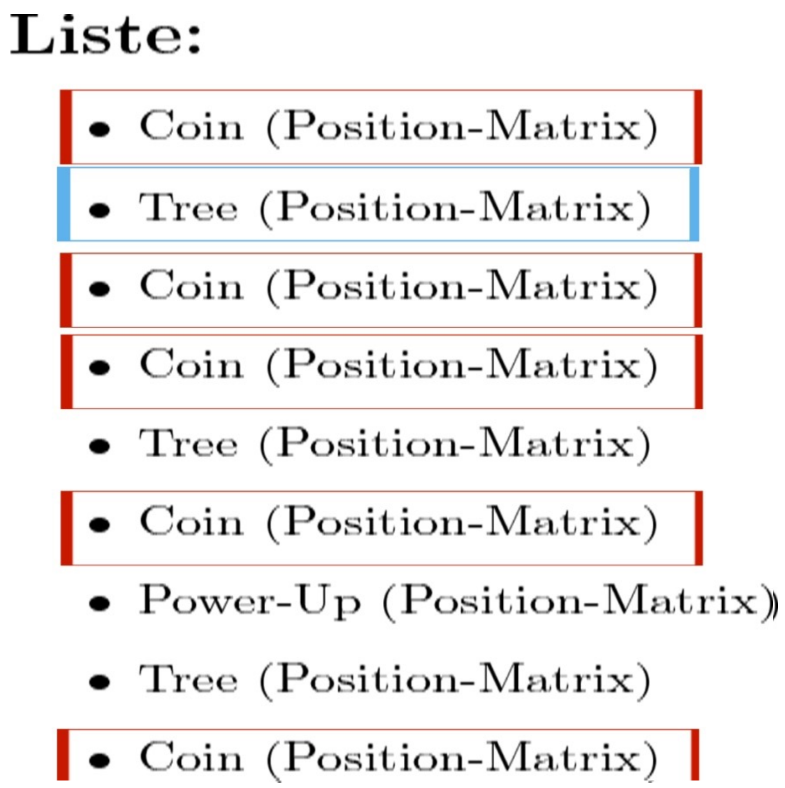
- Use of a movement vector
- Steering by tilting around the z aches
- Different acceleration depending on underground
- calculation of player height
  - Calculating of barycentric coordinates of current position
- Gravity effect when driving down a cliff

Render Menegement

- Entity-Map -> Table
- Double-For-Loop for searching through the table
- Frustrum-Culling
- Adding all to a List



Model Rendering



Liste:

- Coin (Position-Matrix)
- Tree (Position-Matrix)
- Coin (Position-Matrix)
- Coin (Position-Matrix)
- Tree (Position-Matrix)
- Coin (Position-Matrix)
- Power-Up (Position-Matrix)
- Tree (Position-Matrix)
- Coin (Position-Matrix)

- Load Vertecies
- Load Textures
- Light Information
- Matrices
- ...
- Render all model, of the same type

- Next model which is not already being rendered

- Render just what we see (frustum culling)
- Loadings to shaders minimized

Light effects

- Phong Lighthing :
  - Diffuse Color
  - Ambient Color
  - Specular Color
- Phong-Blinn Approximation
  - Calculating specular light with halfvector
- Multiple Light Sources
  - Point-Light
    - Calculatet with the distance from the lightsource

- Individual values for ambient, diffuse, specular color
- Sun changes color during gameplay

Particle effects

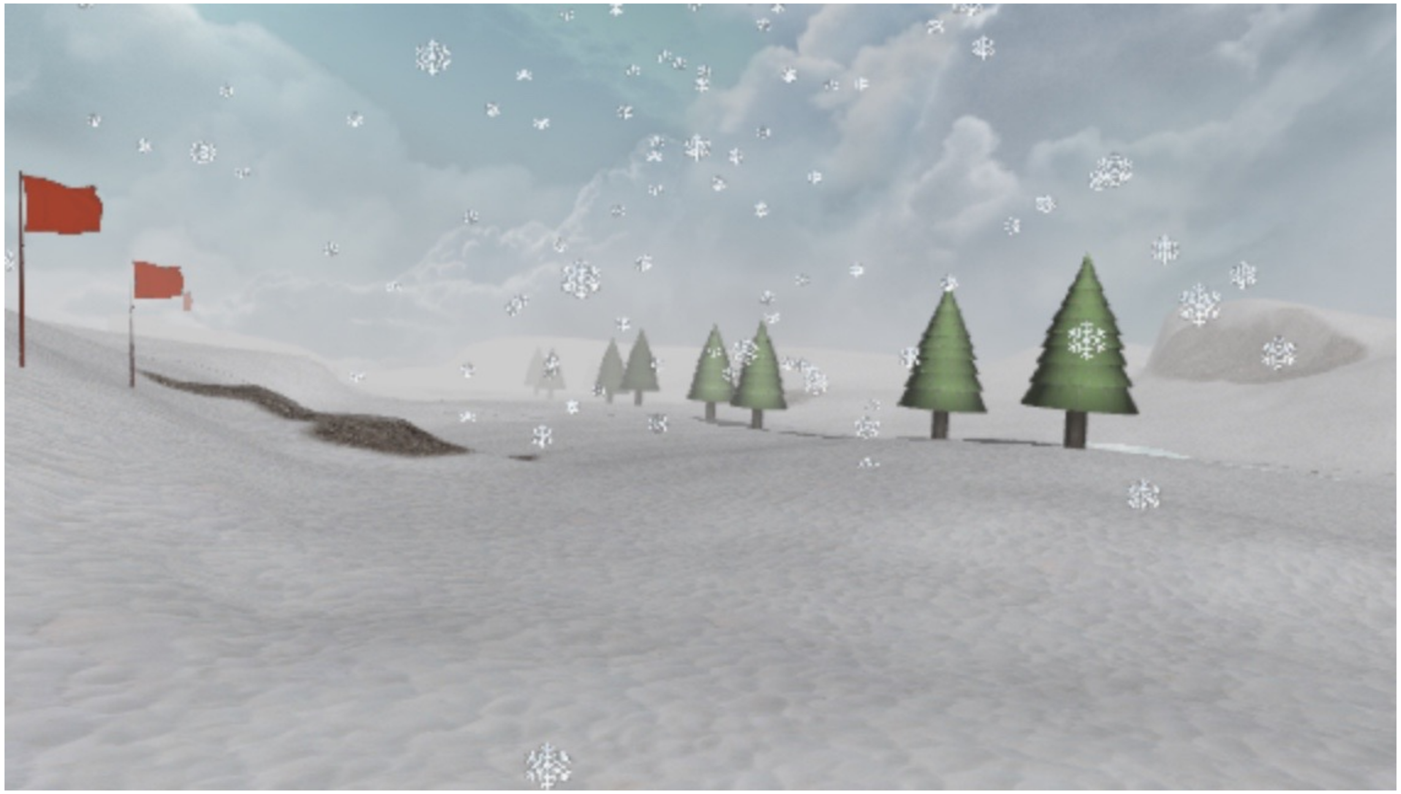
- Rendering snow particles arround the player position
- Fire particle for any flare model

Bluring Effect

- Bluring motion by fast movement

Fog Effect

- Fare models disappear in fog



Shadow Map

- Shadow mapping
- Poisson sampling
- multisampling

Cubemapping

- Displaying the sky with a 3d-texture
- Sky rotation
- Adding reflection to the world and the model