COSC490 SPRING 2011
FINAL GROUP PROJECT

Michael Tice
Aaron Boothe
Daniel Riggs
3-Level Obstacle Course/Maze

- Each member will design a separate level.
- First person view (third person impractical: Explained later)
- The end of one level will link to the next via anchor node.
Character modeling with Biped in 3Ds Max. (problems in VRML)

Moving obstacles - (moving boxes that would knock the character off path)

Various cliffs / Narrow Passages - (problems in VRML)

Obvious Sensor nodes - (make clickable objects ‘pop out’ / look obvious)
Sensors

DEF ProxSensor001 Transform {
  translation 305.9 162.5 -76.15
  children [
    DEF Prox1 ProximitySensor {
      enabled TRUE
      center 0 70.22 0
      size 800 140.4 500
    }
  ]
}

DEF ProxSensor002 Transform {
  translation 118.9 0.005 187.3
  children [
    DEF Prox2 ProximitySensor {
      enabled TRUE
      center 0 18.23 0
      size 2225 36.47 1532
    }
  ]
}
More about Sensors

ROUTE PRXS4.orientation_changed TO T5.set_rotation
ROUTE PRXS4.position_changed TO T5.set_translation

ROUTE TouchSensor002-SENSOR.touchTime TO SCORE.Trigger
ROUTE TouchSensor003-SENSOR.touchTime TO SCORE.Trigger
ROUTE TouchSensor004-SENSOR.touchTime TO SCORE.Trigger
ROUTE TouchSensorforBox1-SENSOR.touchTime TO SCORE.Trigger
ROUTE TouchSensorforPlane5-SENSOR.touchTime TO SCORE.Trigger
ROUTE TouchSensorforPlane8-SENSOR.touchTime TO SCORE.Trigger
ROUTE TouchSensor005-SENSOR.touchTime TO SCORE.Trigger
ROUTE TouchSensor006-SENSOR.touchTime TO SCORE.Trigger
ROUTE SCORE.Score TO TXT2.set_string

ROUTE TS2.cycleTime TO CHECKCAMERA1.Trigger # triggers the camera1 script
ROUTE CHECKCAMERA1.ResetCamera TO Camera001.set_bind # makes camera1 the active camera

ROUTE CONE.collideTime TO DEATHTOUCH.CollisionTime # send the collide time
ROUTE DEATHTOUCH.ResetCamera TO Camera003.set_bind # send the script output
ROUTE Prox1.isActive TO Camera001.set_bind
ROUTE Prox2.isActive TO Camera003.set_bind
ROUTE TouchSensorforBox1-SENSOR.touchTime TO Box1-TIMER.startTime
ROUTE TouchSensorforPlane5-SENSOR.touchTime TO Plane012-TIMER.startTime
ROUTE TouchSensorforPlane8-SENSOR.touchTime TO spikes-TIMER.startTime
ROUTE TouchSensorforPlane5-SENSOR.touchTime TO Plane013-TIMER.startTime
ROUTE TouchSensorforPlane8-SENSOR.touchTime TO Box009-TIMER.startTime
ROUTE TouchSensor002-SENSOR.touchTime TO Plane011-TIMER.startTime
Lights

- Omni-directional lights.
- Emissive lights from objects
Goals/Objectives

- Collision Detection with objects
- Falling/ restarting level (dying)
- Manipulating light sources
- Character Movement (jumping, crouching, etc.)
  (DNC)
- Crowds with simple behavior that need to be avoided.
- Teleporters. Objects that transport the player to a new location when interacted with. (DNC)
Why VR?

- For an interactive game/maze VR is the perfect format to give the best experience for the user.
- More visual stimulation as opposed to text-based games.
- Feels more ”rewarding”
Collision Detection Physics not perfect & difficult to implement

Physics: Moving platform with character on it doesn’t function properly

Clipping issue with the way that the objects are rendered when there is no light source.

Timing/Scoring had to be created with objects/ script to orient to the player (unnecessary processing, could affect performance in larger program)
Problems in VRML (cont.)

- Player death - Unable to implement in Max, only possible with scripting/ camera manipulation in VRML
- No shadows in VRML!
Remaining problems

- Character implementation (skin/ animation/ movement)
- Improved character control/ movement
- Physics/ interaction
Hardware/Software used

- Intel Core 2 Quad 2.40 GHz 8GB
- Windows 7 Enterprise
- 3D Studio Max
- VrmlPad
Sources

- [http://www.cs.bowiestate.edu/~sharad/vrlab/research.html](http://www.cs.bowiestate.edu/~sharad/vrlab/research.html) (previous projects/research)
- Google Images (for textures)