Escape from Planet Mars: Simulation of a Mars Colony and Evacuation

Designed by:
Kourtney Ramseur
Tunde Ballack
Kaifi Jamil
Babatunde Adekoya

(Report by Kaifi Jamil)

COSC 729: Virtual Reality I with Dr. Sharma
May 5, 2015

Introduction

Mars has been the subject of exploration by NASA and other space agencies during the past decade and a perennial interest for futurists, writers, and scientists. More recently there is even speculation about what efforts would be needed to establish a long-term colony of humans on Mars for research purposes. We were inspired by these interests to create the virtual environment for a small Mars colony with futuristic (but plausible) buildings on a Martian landscape with several avatars to represent colonists. For interactivity we decided to set up an evacuation scenario in which the avatars panic and the main user must walk through rooms and hallways of buildings to collect a few key items. Once the user collects the items he has completed the evacuation mission and "wins" the scenario.

Goals and Objectives

Firstly, we felt that such a project idea would be fun to design because some of us are science-fiction enthusiasts. Creating a virtual colony on Mars allows us to use the Vizard, Sketchup and 3DS Max skills we acquired earlier in the semester in a new way: representing a man-made structure that is not even on Earth. Secondly, we decided to use our knowledge of university campus evacuation in this new extraterrestrial setting. The evacuation is, however, a simple game that is not meant to simulate all the realistic concerns that would arise during a real threat on the colony. We do not have the expertise or resources to determine that in the limited time we had to complete the project. The user who plays through the scenario should be able to have fun with the experience and appreciate both the internal setting of the rooms and halls as well as the external landscape of a red desert.

Modeling & Programming

As can be observed from playing through the simulation and viewing the screenshots, the virtual environment is a collection of buildings and symbols on a Martian landscape. It is meant to be a reasonable replica of what a Mars colony could look like but it may not contain the exact number of apartment rooms, labs, indoor gardens, and other facilities that would be necessary to sustain an actual population in future reality. Our first step in modeling the environment was to search for buildings and the external landscape. We found most of these through the 3D Warehouse in Sketchup, a popular 3D modeling software. We found the smaller items and furniture from SketchUp and 3DS Max as well and positioned them inside the

buildings using SketchUp and the Inspector in the Vizard IDE. We added textures for some of the buildings while leaving others' walls as plain, believing that a Mars Colony would not try to appear too aesthetically pleasing in the first place. We found the textures we did use through SketchUp and in some cases simply from website searches through Google. We used Sketchup as well as Vizard to scale the buildings to a size that would be appropriate for the Colony and relative to human colonists, but it was difficult to do this accurately.

We used the Vizard IDE to enhance the models and environment with sound files, avatar animations, interactivity of picking up items, sensors, and rewarding the user when he or she successfully finishes the evacuation.

- Audio files, we included sounds for a mandatory alarm, reward sound for when an item is collected, background music borrowed from the Japanese video game *Final Fantasy XIII* (the battle theme), an avatar screaming "run for your lives" when clicked, an injury sound for another avatar, and the *Star Wars* theme if the user finishes the game in time.
- Sensors: We set up proximity sensors around the key items so that when the user walks close to the item the item will disappear and the item counter will increase by one.
    - A timer sensor keeps track of how many seconds are left to complete the evacuation. When the timer reaches five seconds remaining the color of the numbers change and flash.
    - There are touch sensors on two avatars. One of them raises his hands and shouts when clicked (panic!). The other sensor touch sensor causes a male avatar to fall down and the female next to him cheers.
- Animations: The avatars in the scene perform actions when the e, f, and r buttons are pressed by the user.
- Avatars: The avatars used are the inbuilt ones from the Vizard IDE.
- Evacuation threat: We found objects to represent aliens that were modeled on "Skitters" from the TV series *Fallen Skies*.
- Sky and environmental map: For most of the simulation the sky outside buildings will appear dark like outer space, with a few moons and the Earth visible to the viewer. For the end scene we used a blue sky because we did not know how to display a "outer space" sky for a cube texture.

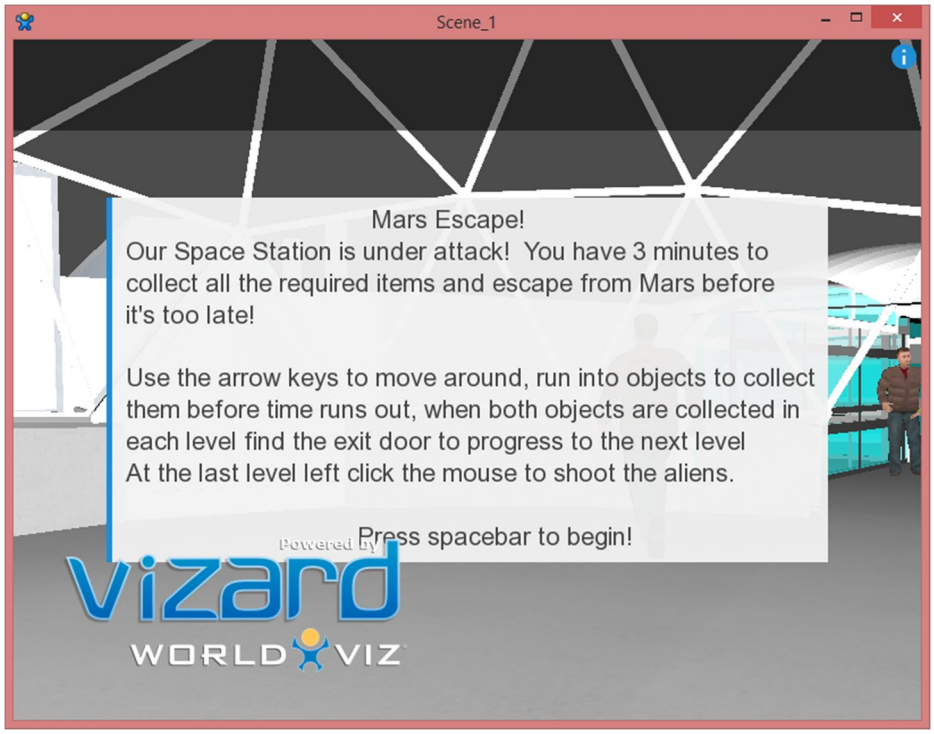
Problems & Future Improvement

Some of the difficulties we faced with the programming aspect of this project included connecting one scene with another when the user enters a new room, maintaining the correct score and timer when transitioning scenes, scaling the building and items, finding appropriate items for the user to collect, and implementing the ability to shoot the aliens invading the colony on the Mars landscape. For the alien challenge we were unable to implement a way for the user to shoot the aliens. The lasers do not recognize when they have hit an alien, the alien does not disappear from the screen, and the user's health bar does not lose points if one is hit by an alien.

Our suggestions for improvement would be to make the alien battle more realistic and workable so that lasers cause each alien to disappear and they also affect the user if they collide with the user. Another improvement would be to find and animate custom avatars wearing space suits when they are outside the colony buildings. It would be helpful if there was a way to scale the buildings more accurately than to rely on the programmer's visual guesswork. The simulation would be more realistic if one could find more buildings to include and the right type of buildings, such as a bio-dome or indoor garden.

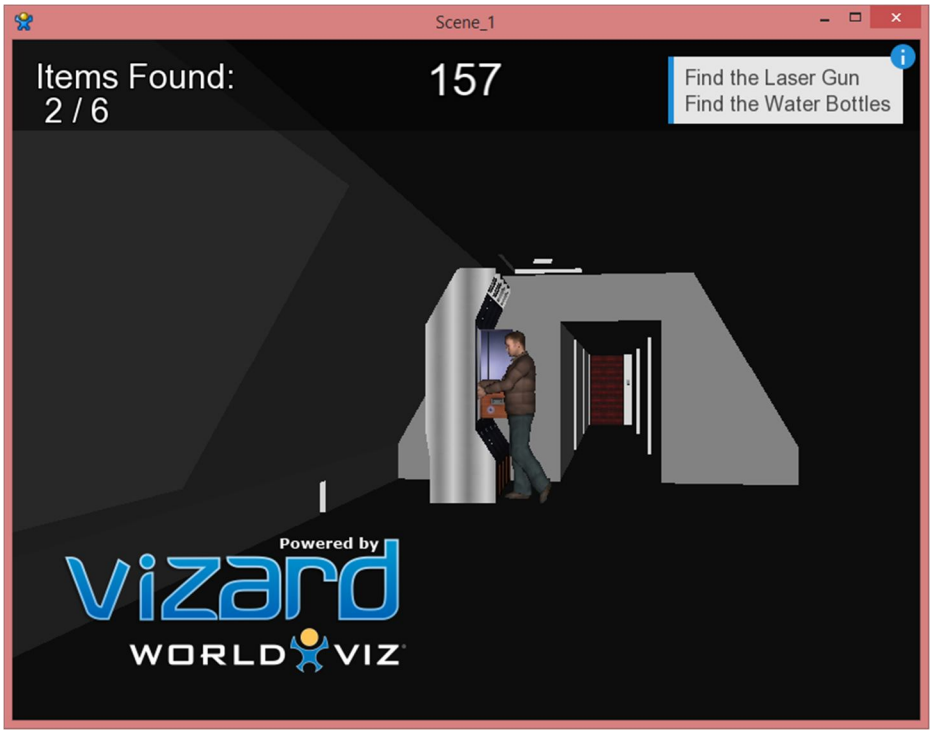
Although this virtual environment was fairly simple and completed on a modest scale for a class project, the user could feel more immersed in the environment if interfacing with it through a Head-Mounted Display. We believe that NASA or other space agencies should attempt to create virtual colonies in order to realistically plan settlements, precautions, and human needs.

Screenshots & Code Snippets


Scene_1
Mars Escape!
Our Space Station is under attack! You have 3 minutes to collect all the required items and escape from Mars before it's too late!
Use the arrow keys to move around, run into objects to collect them before time runs out, when both objects are collected in each level find the exit door to progress to the next level
At the last level left click the mouse to shoot the aliens.
Press spacebar to begin!
Powered by
vizard
WORLD VIZ


Scene_1
Items Found:
1 / 6
141
Powered by
vizard
WORLD VIZ


Scene_1
Items Found:
2 / 6
120
Exit
Powered by
vizard
WORLD VIZ

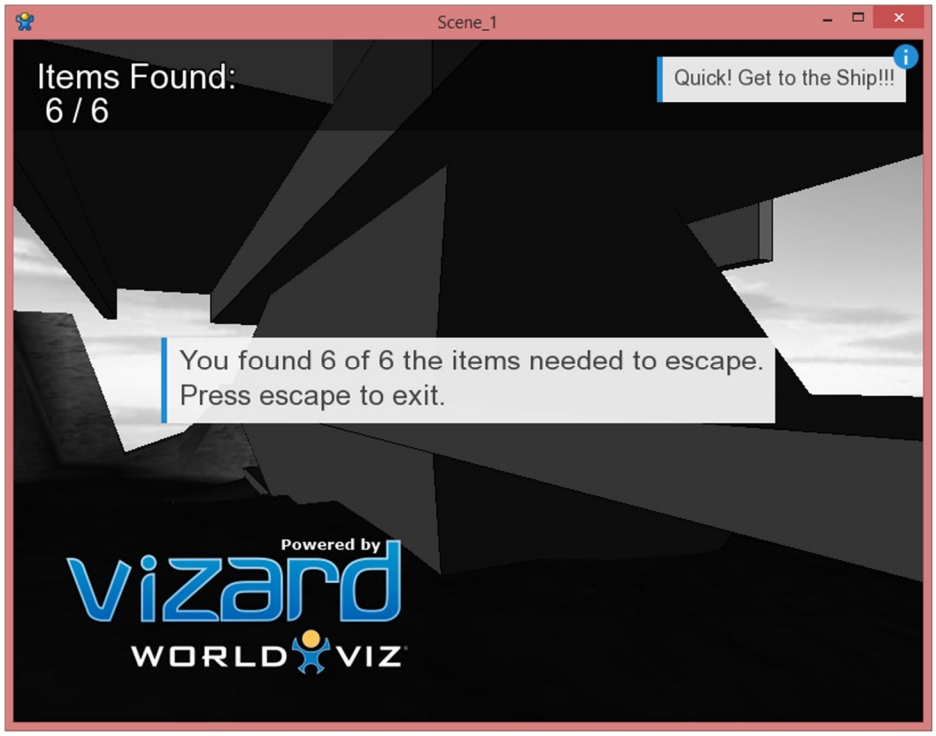

Scene_1
Items Found:
2 / 6
157
Find the Laser Gun
Find the Water Bottles
Powered by
vizard
WORLD VIZ


Scene_1
Items Found:
3 / 6
128
Find the Laser Gun
Find the Water Bottles
Powered by
vizard
WORLD VIZ


Scene_1
Items Found:
4 / 6
73
Find the Bag
Find the Key Card
Powered by
vizard
WORLD VIZ


Scene_1
Items Found:
6 / 6
26
Quick! Get to the Ship!!!
Powered by
vizard
WORLD VIZ


Scene_4
Items Found:
6 / 6
Quick! Get to the Ship!!!
Powered by
vizard
WORLD VIZ



Code Snippet 1:

```
################ Movement Control Code ########################

#Script to control viewpoint with keyboard arrow keys
MOVE_SPEED =5000
TURN_SPEED = 60

def updatecar():
    #move view forward and backward
    if viz.key.isDown(viz.KEY_DOWN):
            view.move([0,0,MOVE_SPEED*viz.elapsed()],viz.BODY_ORI)
    if viz.key.isDown(viz.KEY_UP):
            view.move([0,0,-MOVE_SPEED*viz.elapsed()],viz.BODY_ORI)
    if viz.key.isDown(viz.KEY_PAGE_UP):
            view.move([0,MOVE_SPEED*viz.elapsed(),0],viz.BODY_ORI)
    if viz.key.isDown(viz.KEY_PAGE_DOWN):
            view.move([0,-MOVE_SPEED*viz.elapsed(),0],viz.BODY_ORI)

    #rotate body of view left and right
    if viz.key.isDown(viz.KEY_RIGHT):
            view.setEuler([TURN_SPEED*viz.elapsed(),0,0],viz.BODY_ORI,viz.REL
_PARENT)
    elif viz.key.isDown(viz.KEY_LEFT):
            view.setEuler([-
TURN_SPEED*viz.elapsed(),0,0],viz.BODY_ORI,viz.REL_PARENT)
vizact.ontimer(0,updatecar)
```

Code Snippet 2:

```
############# Avatars & Animation added to Environment #############
avatar1 = viz.addAvatar('vcc_male.cfg')
avatar1.setScale(700,700,700)
avatar1.setPosition(-7000, 1490, -15000)
avatar1.state(1)

def Avatar1():
        #vizact.onkeydown('r',runDelay)
        walk1 = vizact.walkTo([-8000, 1550, -17500], walkSpeed= 750, verb =
'run')
        walk2  = vizact.walkTo([-9000, 1550, -17000], walkSpeed= 750, verb =
'run')
        walk3  = vizact.walkTo([-9000, 1550, -17000], walkSpeed= 750, verb =
'run')
        walk4  = vizact.walkTo([-11000, 1550, -18000], walkSpeed= 750, verb =
'run')
        walk5  = vizact.walkTo([-13000, 1550, -21000], walkSpeed= 750, verb =
'run')
        walk6  = vizact.walkTo([-7000, 1490, -15000], walkSpeed= 750, verb =
'run')
        avatar1.addAction(walk1)
        yield viztask.waitTime(5)
        avatar1.addAction(walk2)
        yield viztask.waitTime(5)
        avatar1.addAction(walk3)
        yield viztask.waitTime(5)
        avatar1.addAction(walk4)
        yield viztask.waitTime(5)
        avatar1.addAction(walk5)
        yield viztask.waitTime(5)
        avatar1.addAction(walk6)
viztask.schedule(Avatar1())
```