BOWIE STATE UNIVERSITY
SPRING 2015

COSC 729 : VIRTUAL REALITY AND ITS APPLICATIONS

PROJECT REPORT:
GAMING : ROBOT CAPTURE

PROFESSOR:
Dr. SHARAD SHARMA

STUDENTS:

Issiaka Kamagate
Jamil Ramsey

OUTLINE

INTRODUCTION

1. GOAL AND OBJECTIVES
2. MODELING
3. PROBLEMS ENCOUNTERED
4. FUTURE IMPROVEMENTS
5. RESUME

INTRODUCTION

Video games have become an incredibly popular and pervasive form of entertainment. Video game use has increased steadily overtime (Rideout, Foehr, & Roberts, 2010) and today 9 out of 10 American children and teens play video games (Gentile, 2009; Gentile & Walsh, 2002). On average, youth play video games for two hours a day (Rideout, Foehr, & Roberts, 2010). However, a significant percentage of males report playing four hours a day or more (e.g., Bailey, West, & Anderson, 2010). This fact that such a large number of children and adolescents frequently play video games makes video game very important today.

1. GOAL AND OBJECTIVES

The objective of this game is to capture 5 escaped robots on a college campus. The game will simulate a crowd's reaction to danger while we try and capture the robots.

2. MODELING

- Environment: The game is taking place in a virtual college campus. The virtual campus is a unique blend of old and new. Boasting numerous cultivar of trees, shrubs, and flowers, the campus is dotted with structures of the early 1900s, as well as buildings exemplifying the most contemporary modes of architecture. There is also a soccer field.

Picture 1: screenshot of virtual campus

Picture 2: screenshot of virtual campus

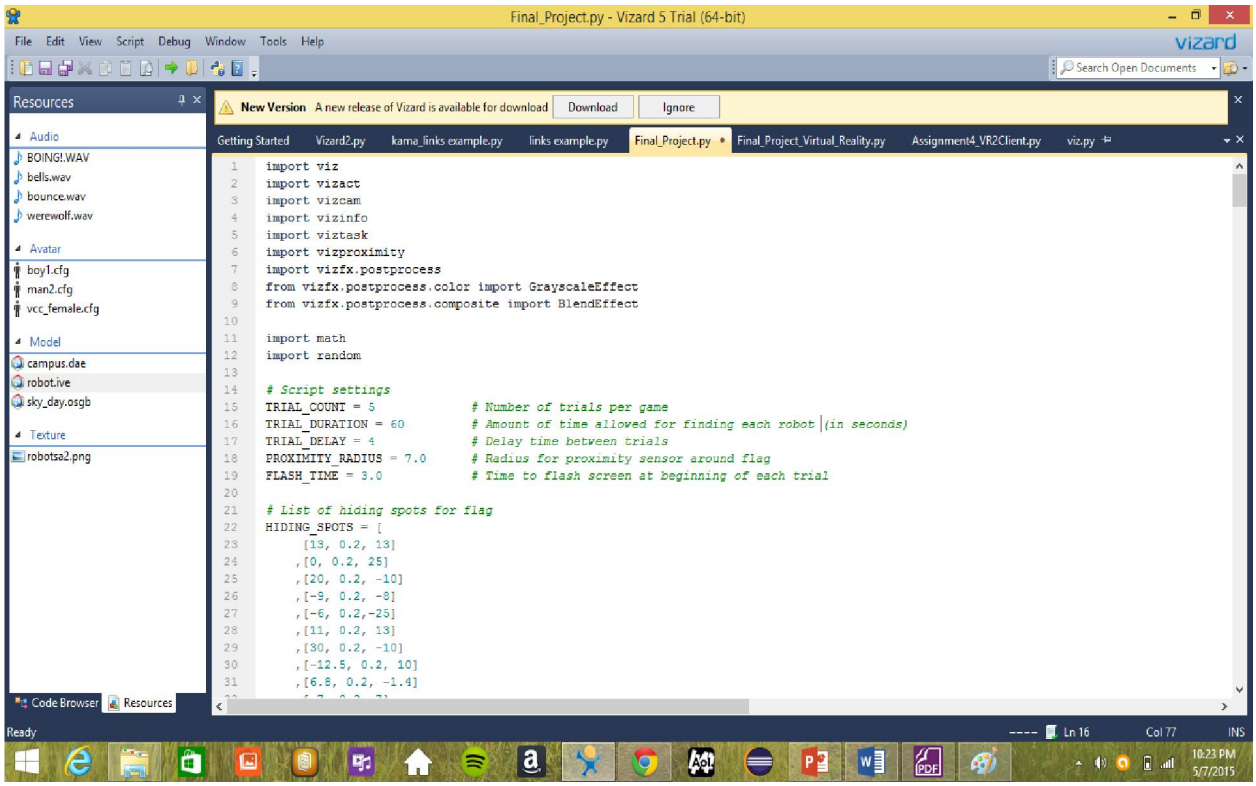
- Light: We used a global illumination.
- Sky: We used a dynamic sky (sky_day.osgb) including in vizard.
- Timers: The game also integrated a timer of 60 second to find the robot.
- Audio: There are several sound in the game. First, there is a sound of a bell which allows the player to locate the robot. Next, there is a sound for catching or not the robot and a sound to alert the player when he has less than five second.

- Programming tools:

  Design: We used SketchUp to design the environment.

  Programming: Vizard software for building, rendering, and deploying the applications. Vizard uses Python for scripting.

Picture 3: vizard screenshot

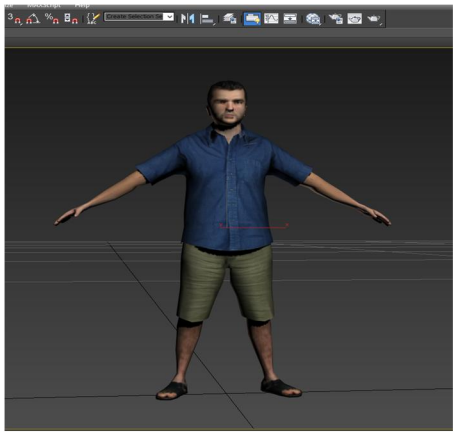


- Avatars: We included some inbuilt avatars and agents from vizard with their own animations. We used 3DS Max and bip motion capture files to create and run animations for inbuilt avatars.
  Avatars have multiple animations like talking, seating, walking and dancing.

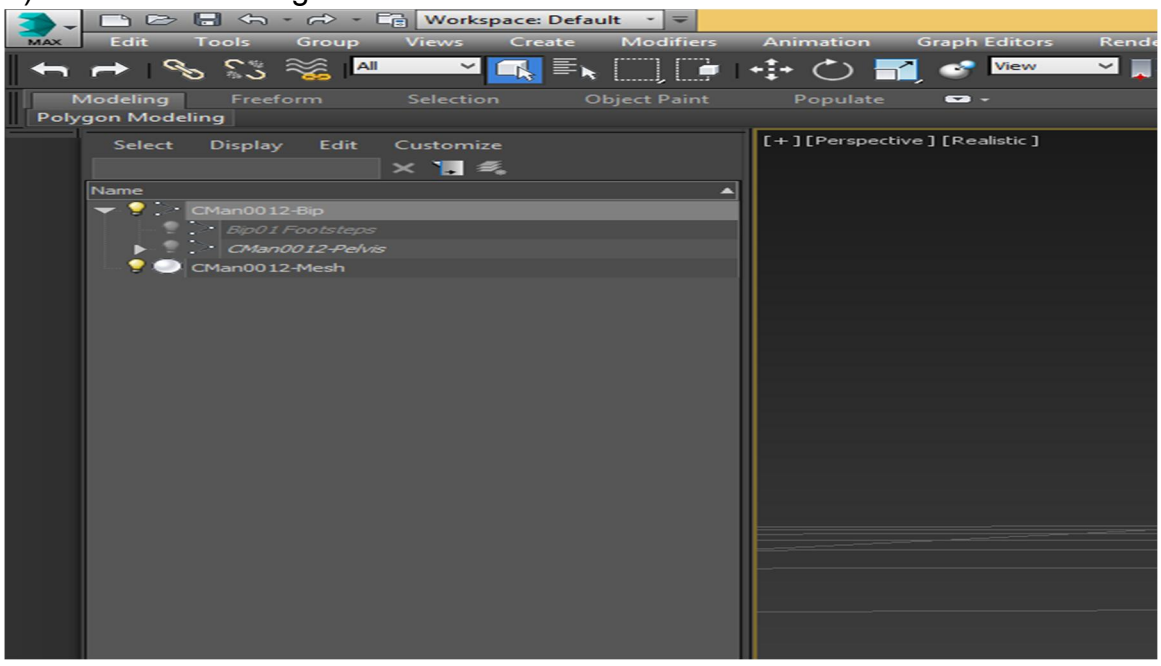
8 custom avatars were created using 3D Studio Max. Each avatar has 4 animations. A stand, walk, sit, and run animation.

Custom Avatar creation procedure:

1) We started with a fully textured and rigged model in MAX, The cal3d export plugin should already be installed. For the cal3d export to work correctly you MUST export the Skeleton first (followed by the mesh(s), material(s), and animation(s) in any order you choose)

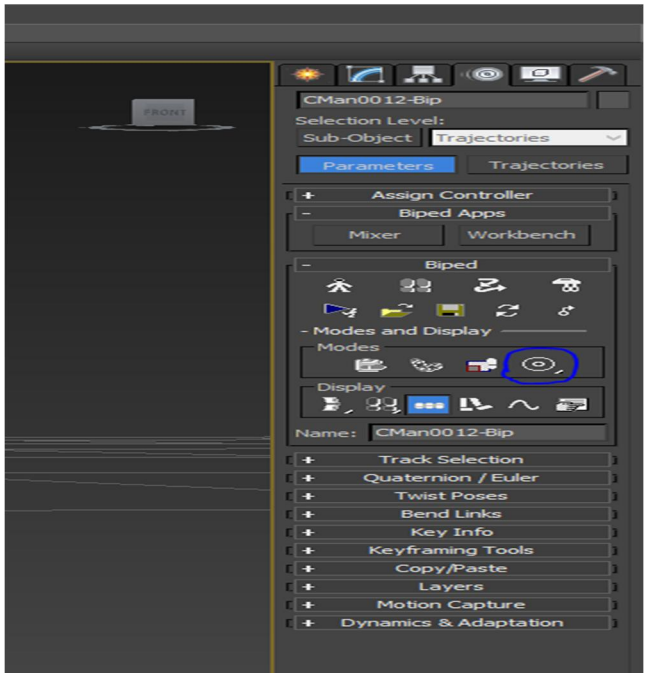


2) Select the bone rig

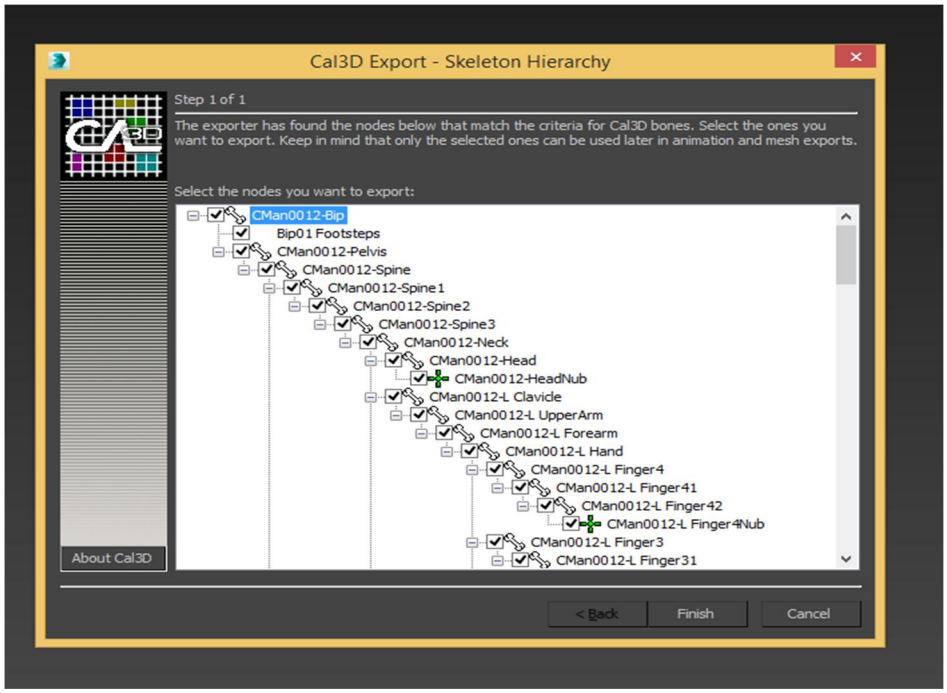


3) Select the motion menu and click the load file button.

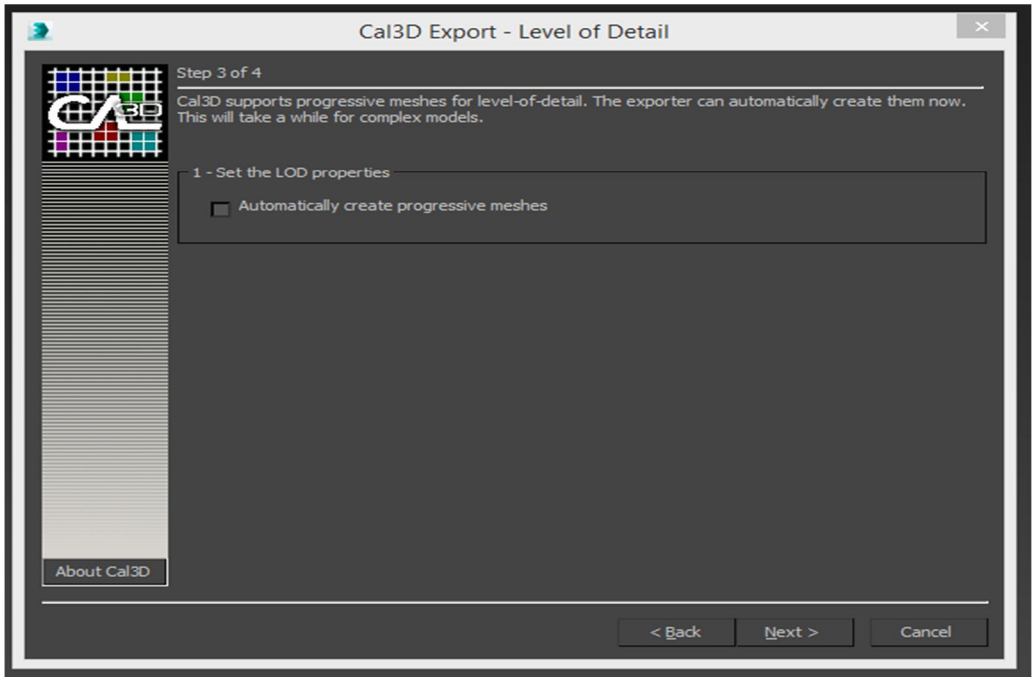
4) Select “In place mode”



5) Select Export > Cal3d Skeleton. All bones should be selected. Click finish.

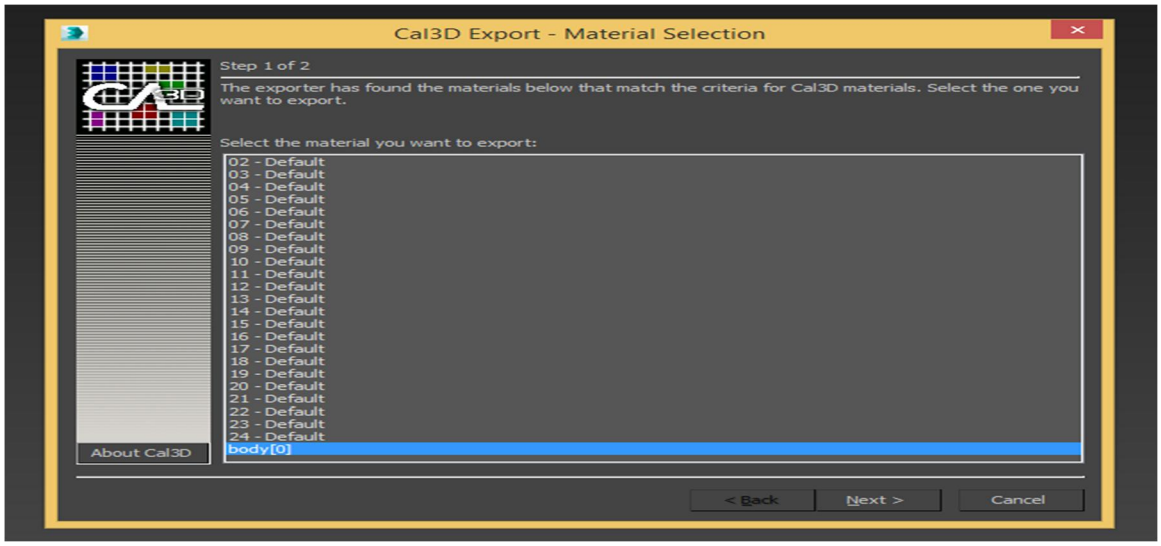


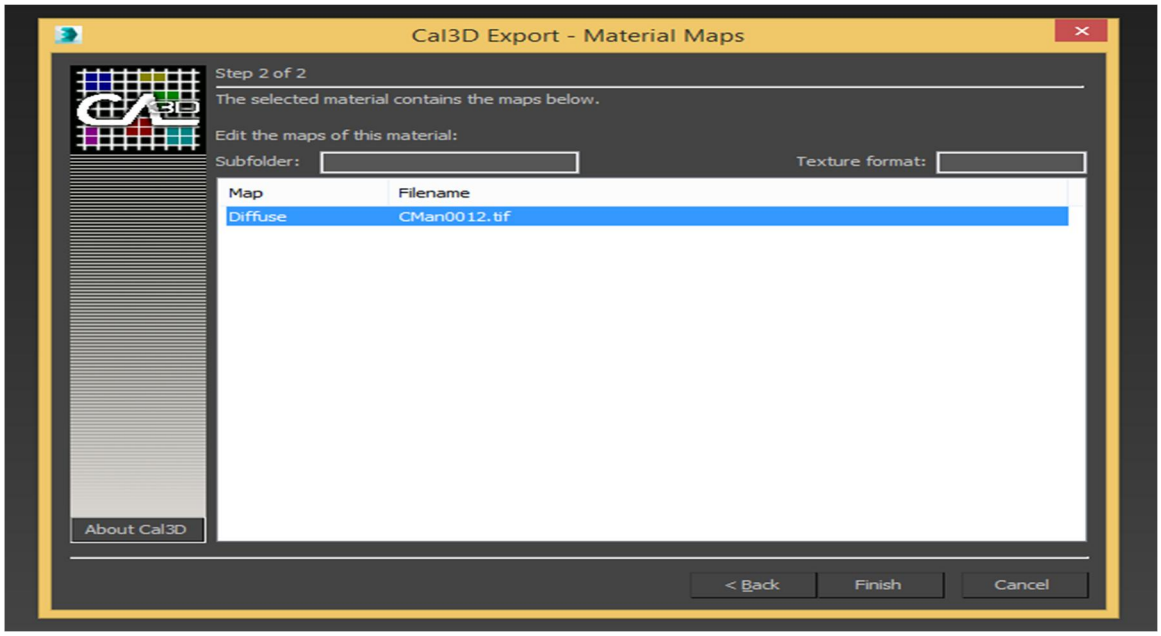
6) Select the mesh in the object explorer and then Export > **Cal3d Mesh**. Select the defaults and finish.



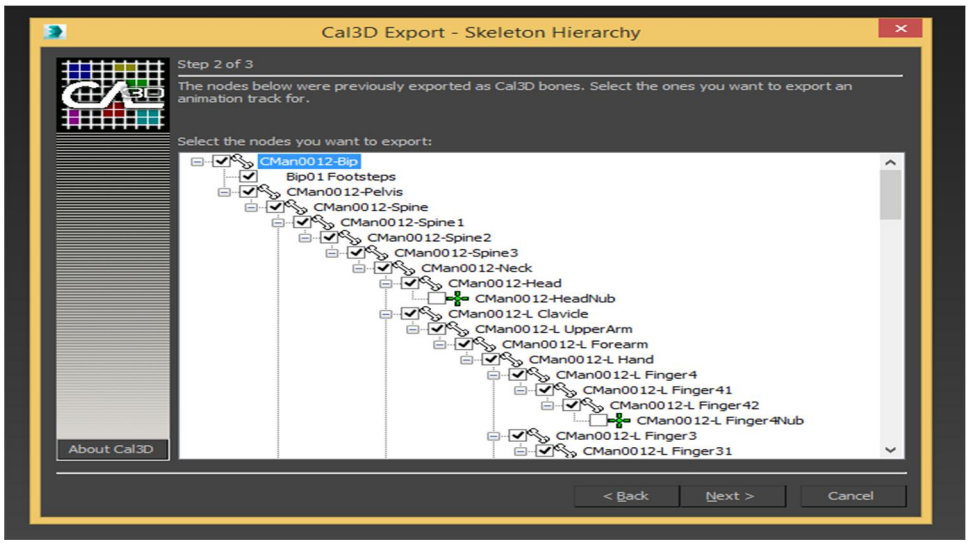
7) Select the material editor. Right click the textured node and rename. Material must follow a specific format. They must be ordered starting with 0. My model only had a single full body texture so I renamed it body[0]. If you textures are broken up you may have to do leg[0], arms[1], body[2]….

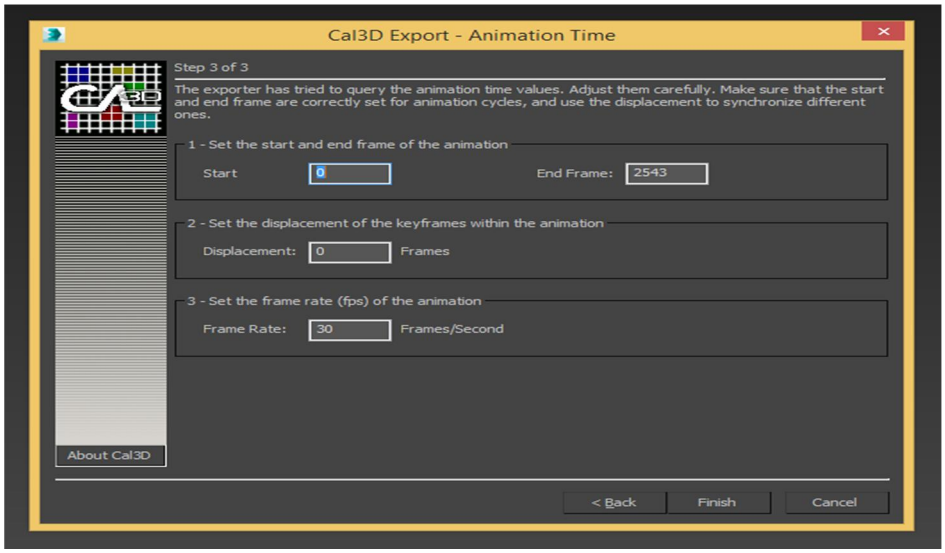
After your textures are named properly select the mesh in the object browser the **Export > Cal3d Material** and select the material to export. The texture file will be listed click next. At this point you need to copy the listed texture file to the same folder as the exported cal3d files.





8) Next select Export > Cal3d Animation. By default it'll select the full animation range. If you want to export a partial animation enter the start frame and end frame.





9) Create config file using the format below.

```
##################################################

#

# Cal3d cfg File

#

##################################################

path=cwom0012/
```

scale=0.025

################## Skeleton #################

skeleton=cwom0012.csf

################## Meshes #################

mesh=cwom0012.cmf

################## Animations #################

animation=stand.caf

animation=sit.caf

animation=walk.caf

animation=run.caf

################## Materials #################

material=cwom0012.crf

3. PROBLEMS ENCOUNTERED
   The campus computers did not have the exporter for Cal3d. The instructions for exporting avatars were not clear and took a lot of trail and error to complete.

4. FUTURE IMPROVEMENTS
   - Adding client server functionality
   - Incorporating hardware support (Oculus and Data Glove).

5. RESUME
   Catch The Robot is a capture the flag style game. The objective is to capture 5 missing robots. A proximity sensor is attached to the robot and when you get close the robot is considered captured. The game uses the WASD control scheme. You have a 60 second timer to capture each robot. The robots make a sound that gives you a clue to where it's hidden. Also the player can simulate an evacuation (by pressing 'm' button) to isolate the robot and make easier its capture. It triggers the run animation for the avatar. Simulating crowd behavior when confronted with a dangerous situation.