빈

BIT DOME

WELCOME TO THE BIT DOME. The Bit Dome is an interactive light simulation developed by Zane Cochran for Berry College's Physical Computing final project. It allows a person to immerse themself in a 360 degree environment of light and music. Its core function is based upon the Korean principle of bringing harmony to one's life through a balance of activities including music, meditation, light therapy simulations and interactive games. WE HOPE YOU ENJOY YOUR EXPERIENCE.

BASICS

DOME CALIBRATION

The Bit Dome uses a XBox Kinect to create a continuous 3D image of the Dome's interior. As such, a simple calibration is required to allow the user interact with the Dome by pointing at a variety of calibration lights. To begin, stand on the square until the Dome instructs you to move to the triangle. You will then be asked to point at five calibration lights that will later be used as menu options for choosing a program.

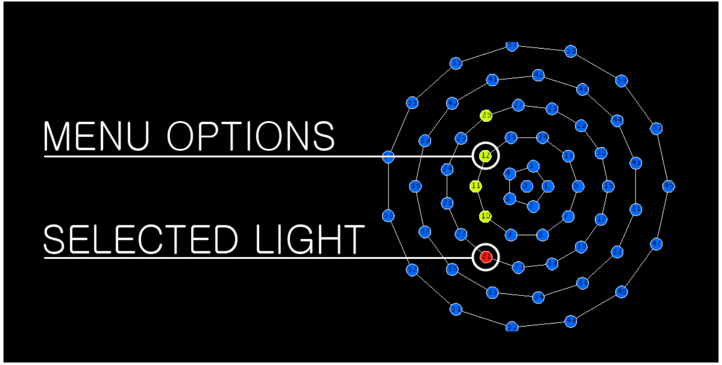
MENU NAVIGATION

After calibration, you can point to the five calibration lights to choose a program. The main menu has four types of programs: Music, Meditation, Simulations and Games. The last option is Exit, which will end the dome experience.

To select a program type, point at its corresponding light and continue point for three seconds. When the dome announces "Loading," stand up straight and wait for the next menu to load. Once loaded, you can choose a program to run. To return to the Main Menu, select the last light.

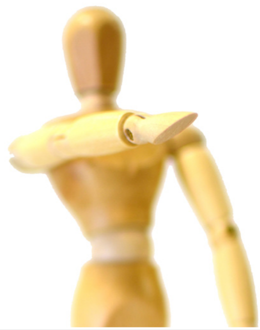
To exit a program at any time, return to the triangle, face the sensor, and place both hands on your head until the dome announces, "The program will now exit."

DOME MENU



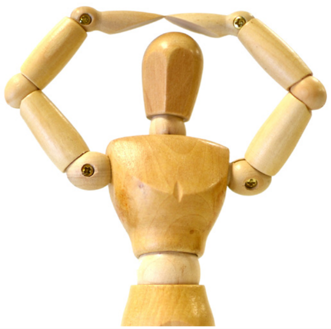
SELECT AN OPTION

CHOOSE A MENU ITEM



POINT WITH RIGHT HAND

EXIT A PROGRAM



BOTH HANDS ON HEAD

BIT DOME

MUSIC

The Bit Dome features four distinct types of music to allow you to express yourself through dancing or relaxation. Each type of music is accompanied by a custom light visualization that is driven algorithmically so that no two experiences are alike. Additionally, the music is provided via Internet streaming service so there is virtually no end to the amount of time you can spend in the dome dancing or relaxing.

As a safety precaution, people who are prone to seizures should avoid these programs as they often include strobing lights. Also, please be considerate as you move around the dome not to make contact with the sides of the structure. Most importantly, enjoy the sound and the lights as you express yourself in the dome.

TRANCE

Electronic dance from around the world causes the dome to progressively light up from top to bottom. The number of lights that illuminate are determined by the intensity of the music and its color intensifies every time it is triggered. Suggested Activity: Dance.

DUBSTEP

Overwhelming bass influences a random pulse of intense lights in the dome. Lights are triggered by an increase in bass which determines a random array of between one and 61 lights to display. Suggested Activity: Dance.

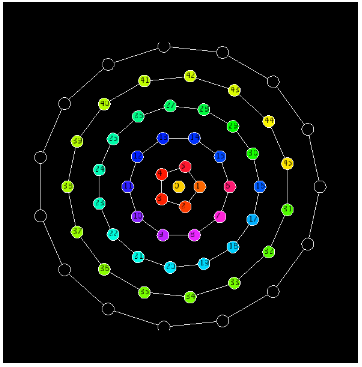
TECHNO

Classic electronica creates a spectrum of color that dances to the beat of the music. Each light gradually steps through the spectrum while strong beats force more rows of lights to illuminate. Suggested Activity: Dance.

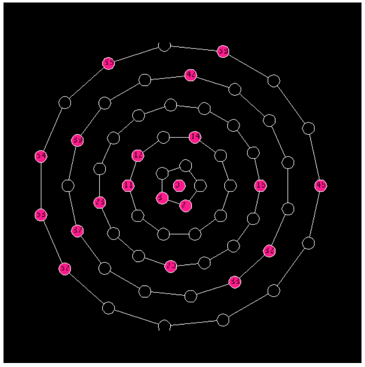
CHILL

Soothing house music invites users to relax and enjoy the subtle transitions of vibrant colors. A Perlin noise filter creates a more organic transition between lights resulting in more subtle changes in tone. Suggested Activity: Sit, Lay, Relax.

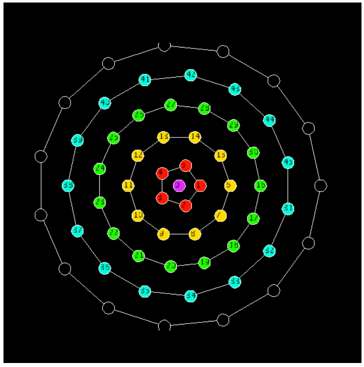
TRANCE



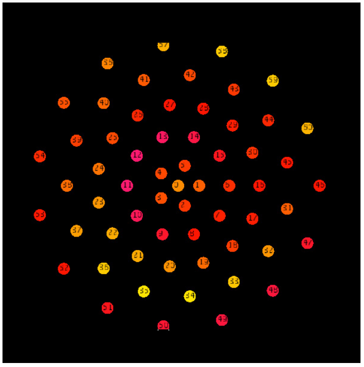
DUBSTEP



TECHNO



CHILL



MEDITATION

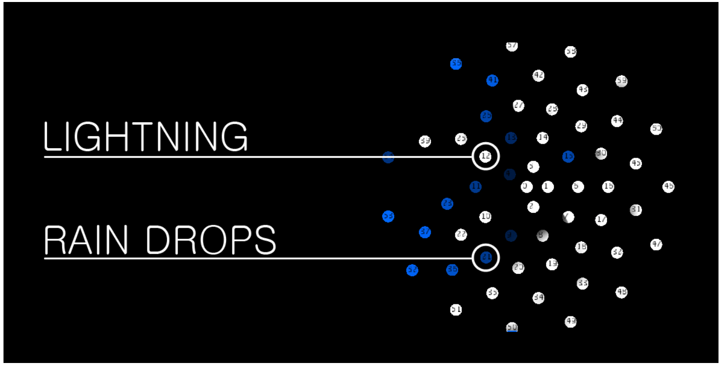
The Bit Dome is the perfect meditative space. It allows you to shut out the world in a completely isolated area away from distractions, noise and light. Every aspect of the environment can be programmed and controlled to suit your needs. For your convenience, the Dome comes with four programs that allow you to relax and concentrate. They include Rainstorm, Sunrise / Sunset, Concentration and Power Hands.

During these programs, please feel free to explore the dome. You may stand, sit or lay in any meditative pose that wish. The important key is to find a position that is comfortable so that you can enjoy the experience and focus on your mind and body. Many of these programs will run for hours if you need an opportunity to escape from the busy world for an extended period of time. There is no rush in the Dome, so please make yourself at home.

RAINSTORM

Relax as subtle drips of rain give way to boisterous claps of thunder in this exciting program. Simulated rain is dynamically generated by the program's algorithm controlling both the number of drops of rain and their speed at which they run down the dome. During particularly lively parts of the storm, flashes of lightning intensify the experience. Suggested Activity: Sit, Lay, Relax.

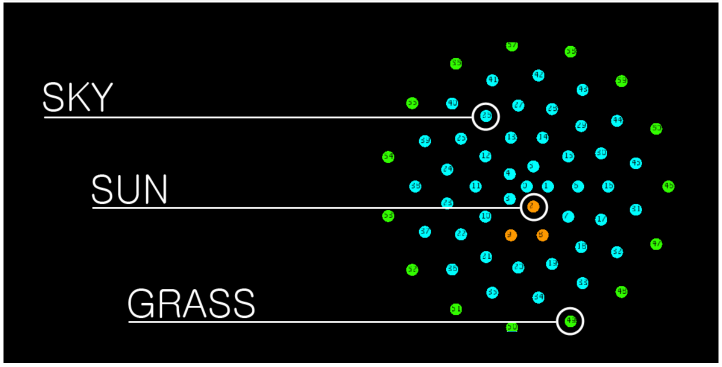
RAINSTORM



SUNRISE / SUNSET

Because the dome is completely immersive, it only seems appropriate to simulate the passing of 24 hour cycle. The program begins at 3 a.m. with crickets chirping as stars begin to appear in a dark blue sky. In time, a sun begins to rise in the east as the sky gradually fades to hues of orange and yellow. Finally the sun makes its way across the blue sky, only to set in the west as birds chirp and play. A vibrant sunset yields to a field of twinkling stars as the cycle begins again. Suggested Activity: Sit, Lay, Relax.

SUNRISE / SUNSET

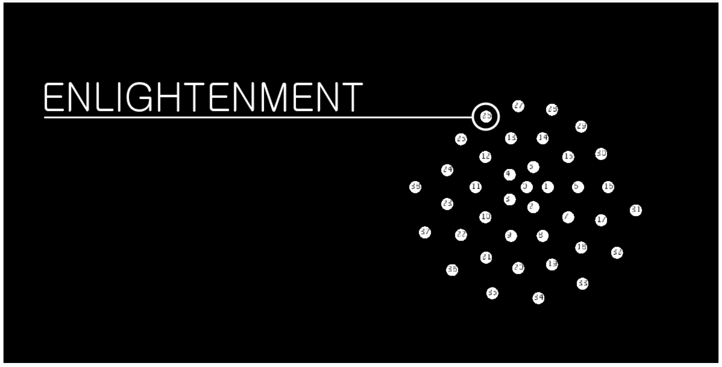


BIT DOME

CONCENTRATION

Test your mind's ability to control your body in this classic challenge. Choose a meditative pose and hold as still as you can as each light in the Dome slowly lights up and rewards you for your effort. However, the slightest movement will result in the loss of your progress. When all 61 lights in the Dome are illuminated, you've reached enlightenment and have mastered this exercise.

CONCENTRATION



POWER HANDS

The power of the Dome is in your hands! Simply point at a light with your right hand to begin charging it with power. Once it has reached its peak capacity, swipe in any direction to see it circle around the Dome until it disintegrates. This particular program uses a wide variety of functions to create an interactive experience. A thorough calibration among 35+ points enables users the ability to interact with individual lights in the dome. Using an extensive list of neighboring lights and some physics, the experience of flinging lights around the Dome can be created.

CHARGE A LIGHT

POINT WITH RIGHT HAND

SEND IT FLYING

SWIPE AWAY

LIGHT SIMULATIONS

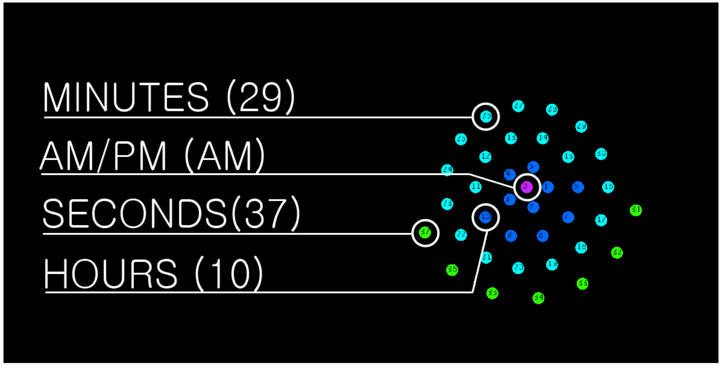
With 61 individual RGB LEDs as your command, the Bit Dome is capable of displaying over one billion possible scenes. For your enjoyment, you can explore the four included light simulations: Dome Clock, Around the World, Conway's Game of Life and Light Painting. These simulations demonstrate the how the Dome can become a powerful way of communicating even complex information in a simple and creative way.

In a world of high resolution screens, the Bit Dome offers a refreshingly minimalist approach to displaying scenes, shifting the effort of interpretation to the users imagination. So sit back and let your mind absorb the stimulating colors and patterns that you can only experience within the walls of the Bit Dome!

DOME CLOCK

Because the Dome completely isolates you from any outside distractions, it's easy to lose track of the minutes and hours that quickly pass. The Dome Clock allows you to quickly check the current time. Because the Dome has 61 LEDs, generating and displaying the current time becomes elementary. Simply read the sequence of overlaying lights to determine the current number of hours, minutes and seconds. Even reading time becomes enjoyable as you watch time slowly creep down the sides of the dome in a veritable display of color!

DOME CLOCK



AROUND THE WORLD

One unique aspect of the Dome is its ability to quickly interface and communicate with its host computer. Around the World demonstrates its versatility and speed as it quickly cycles through a swirling display of brilliant color.

You can adjust the cycle speed by simply raising your right or left hand. Try experimenting with the limits of the software by increasing the speed to an imperceptible level or even bringing the animation to a complete standstill.

INCREASE SPEED

RAISE RIGHT HAND

DECREASE SPEED

RAISE LEFT HAND

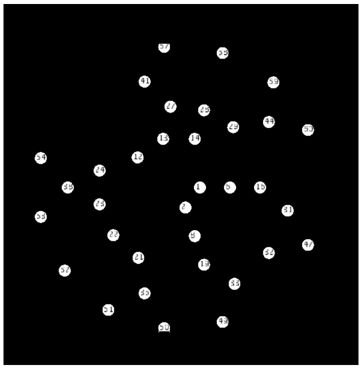
CONWAY'S GAME OF LIFE

Developed in 1970 by John Conway, Conway's Game of Life is a cellular automation that displays the evolution of a species. It requires no inputs besides a randomly assigned initial state. The survival of the dome, as represented by each light depends upon a set of very specific rules:

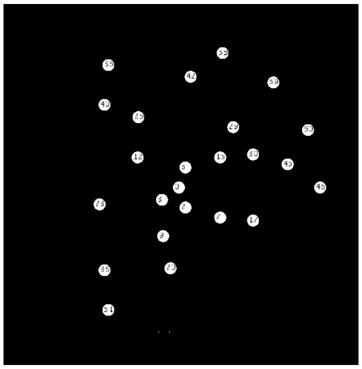
Live cells with < 2 neighbors die by under-population.
Live cells with 2 or 3 neighbors live on to the next generation.
Live cells with > 3 neighbors dies by overcrowding.
Dead cells with 3 neighbors come to life by reproduction.

The speed of each generation can be controlled in a fashion similar to Around the World by raising your right or left hand.

FIRST GENERATION



NEXT GENERATION

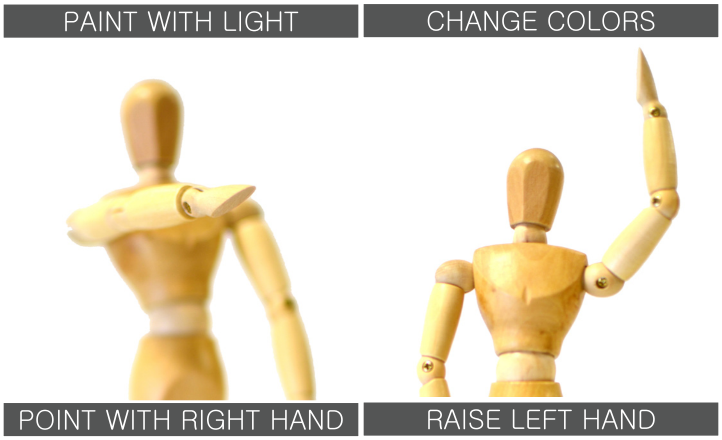


BIT DOME

LIGHT PAINTING

Now that you have had a chance to free your mind of any worries, it is time to create! Light Painting allows you to use a variety of gestures to paint the Dome in a seemingly endless number of colors and patterns.

To accomplish the feat of being able to freely paint the lights in the Dome, it is necessary to create a one-to-one relationship between your hands and the various lights in the dome. To do this, the Dome extrapolates coordinate data from your initial calibration and applies it to the front half of the Dome. As you paint these lights, the Dome automatically mirrors your creation on the back of the Dome as well.



GAMES

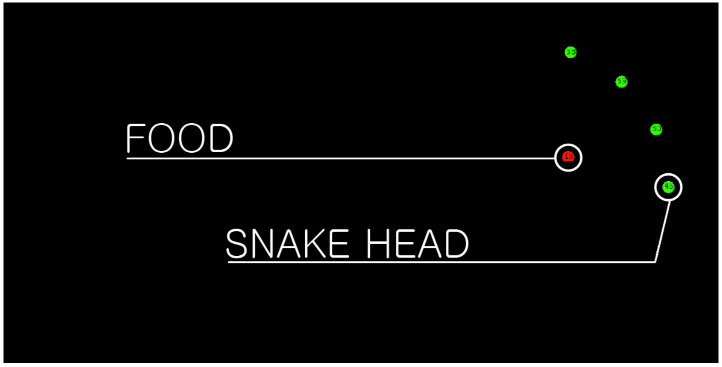
The Korean principle of harmony dictates that every facet of our life must be brought into a balance. After a long day of strenuous work and serene meditation, it only makes sense that one should relax the body and mind and enjoy a fun game or two. With this in mind, the Dome features four games that are sure to bring you happiness. These games include Snake Game, Colors in Motion, Pac Man and Space Invaders.

Using you mind and body, you can compete against the Dome as you jump, crouch, lean and concentrate to win the game! With a victory in hand, you will be able to leave the Dome knowing that you've overcome opposition and can face any of life's challenges.

SNAKE GAME

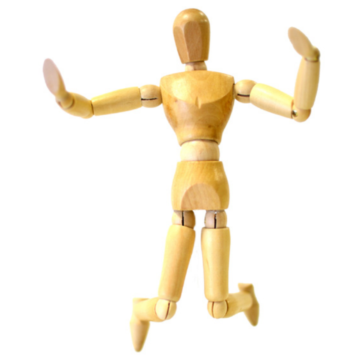
A time-honored favorite, the Snake Game challenges you to maintain spacial awareness within the Dome while trying to steer a hungry snake toward its meal. Every time the snake successfully feeds it grows in length by one segment. Steer the snake toward its destination you'll have to use a combination of timing and agility. To move the snake up in the Dome, lightly jump (and mind your head). If you momentarily crouch you'll convince the snake to go down a level. Leaning from side to side will change the snake's direction as it tirelessly makes its way around the Dome. Once you collect ten pieces of the food, the Dome rewards you with a secret!

SNAKE GAME



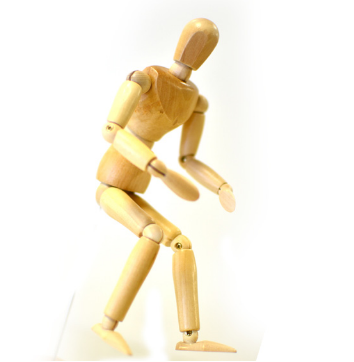
BIT DOME

MOVE SNAKE UP



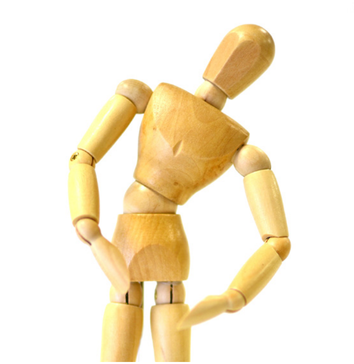
JUMP

MOVE SNAKE DOWN



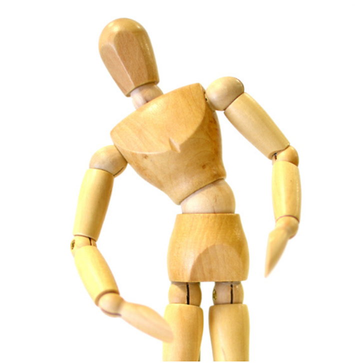
CROUCH

MOVE SNAKE LEFT



LEAN LEFT

MOVE SNAKE RIGHT

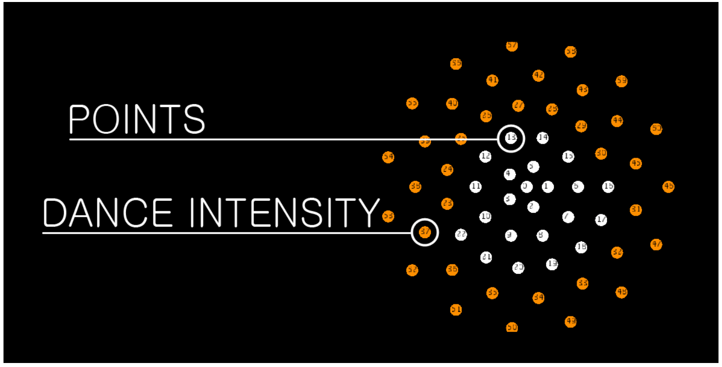


LEAN RIGHT

COLORS IN MOTION

One of the best ways to raise your spirits is to listen to some good music and move your body! With Colors in Motion the goal is to keep moving to the music to increase the color intensity of the Dome. As you dance and groove, you'll soon raise the Dome to its maximum level, white lights begin to illuminate in succession, signifying the number of points you are earning. To master this game, you must reach 60 points before the end of the song. Beware though, if you slow down, the Dome will begin to lose intensity and you'll have to begin again.

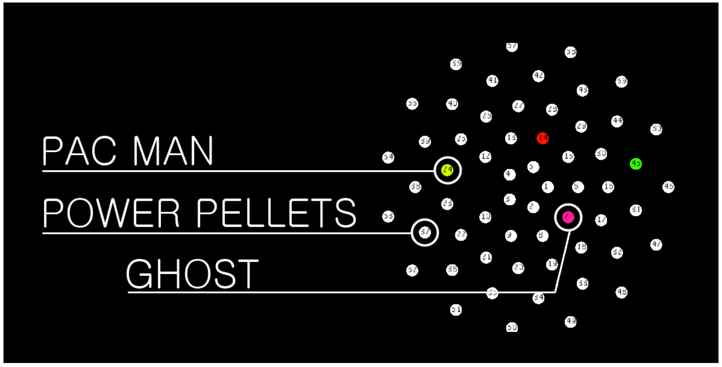
COLORS IN MOTION



PAC MAN

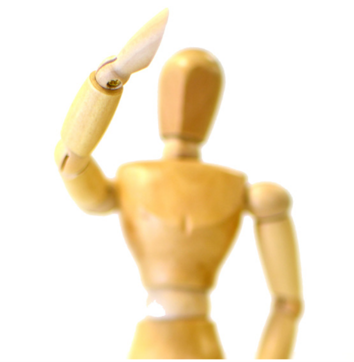
The race is on to collect all the power pellets in the Dome. With you using your right arm like an oversized joystick, you must successfully collect all the white power pellets to win this face-paced game. Watch out for the ever-persuing ghosts. If they touch you, you'll be out of the game. This game engages users by encouraging them move their bodies and minds to ensure Pac Man's safe journey through dangerous territory.

PAC MAN



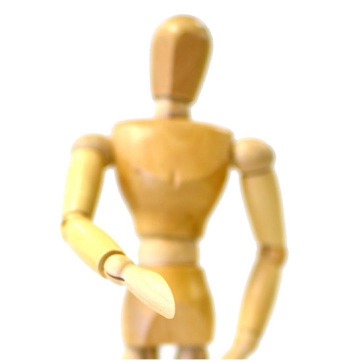
BIT DOME

MOVE PAC MAN UP



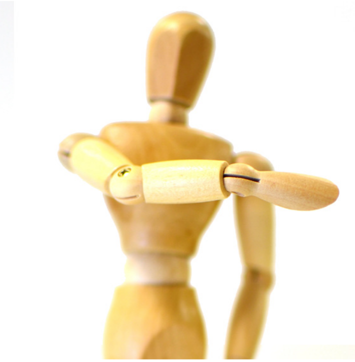
RIGHT ARM UP

MOVE PAC MAN DOWN



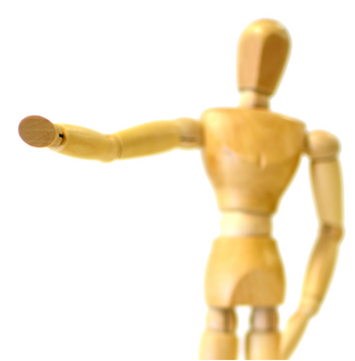
RIGHT ARM DOWN

MOVE PAC MAN LEFT



RIGHT ARM LEFT

MOVE PAC MAN RIGHT

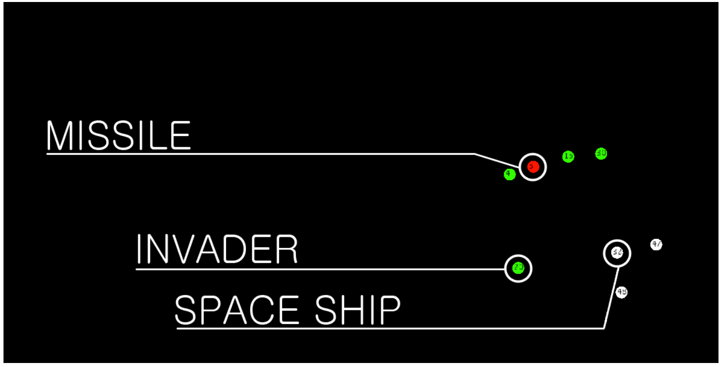


RIGHT ARM RIGHT

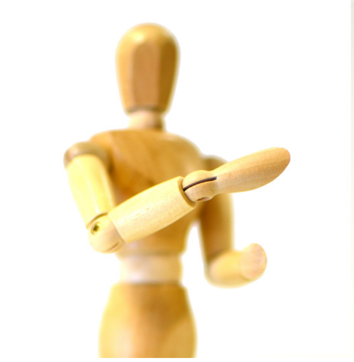
SPACE INVADERS

The Dome is under attack from space! It is your mission to pilot your faithful ship around a shoot down the villainous invaders as the fall from the sky. Using your hands like a steering wheel, you can move your ship and fire missiles by pulling your arms in close to your body. Remember, invaders can come down anywhere in the Dome so watch your back! Every time an invader lands, you lose one of your five allotted lives, so be careful and see how many invaders you can shoot down before they overtake the planet.

SPACE INVADERS

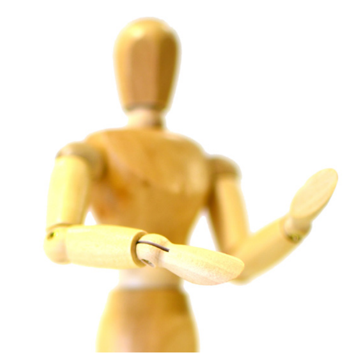


MOVE SHIP LEFT



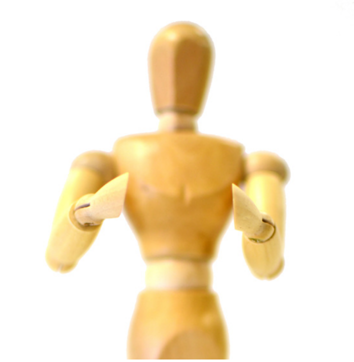
STEER LEFT

MOVE SHIP RIGHT



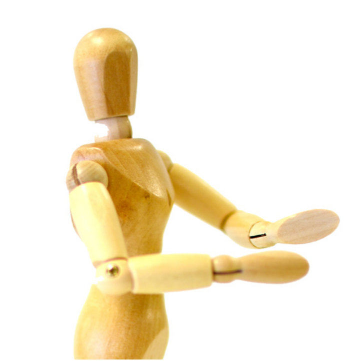
STEER RIGHT

KEEP SHIP STEADY



KEEP ARMS STRAIGHT

FIRE MISSILE



PULL ARMS TO BODY

DEVELOPMENT

The development of the Bit Dome began in September when students were asked during Berry College's Physical Computing class to develop an "imaginary expressive object" as an exercise in creative thinking. Given my recent experience in Korea and this prompt, I created an imaginary space where one could isolate themselves to meditate that could be creatively lit with a variety of lights. In this exercise, the dome could interact with its user through a 3D camera and its array of colorful lights. When this idea was well received by the class, I began to consider building a physical implementation of the Dome.

Construction began in early November and continued through December to completion. These several weeks represent a large amount of physical construction and software development, in addition to a large amount of time spent test and collaborating with others on the project. To address the individual modules of the Bit Dome the following summary will divide the process into two distinct sections: hardware and software.

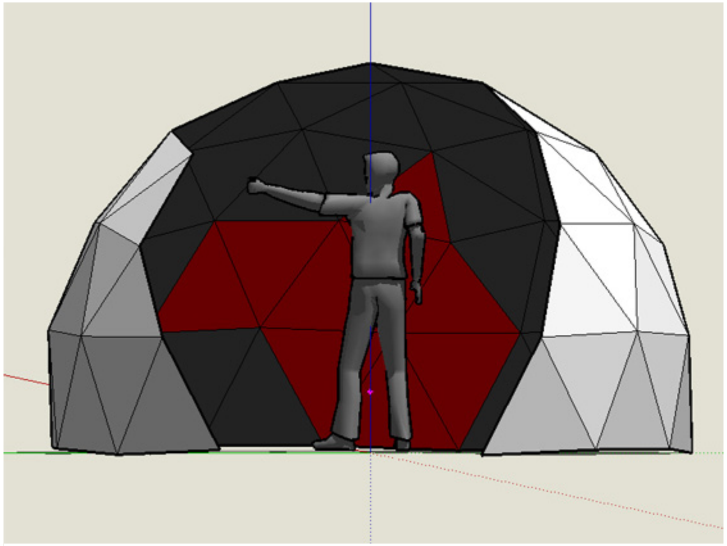
HARDWARE

The hardware used in this implementation of the Bit Dome is fairly simple. It consists of the Dome structure itself, an 8x8 matrix of RGB LED lights, a Rainbowduino microcontroller, a XBox Kinect, a speaker system and a laptop. By interfacing these various components, it is possible to immerse a user into an experience unlike any other. These systems work together to allow the user to isolate themselves and enjoy a truly one-on-one experience with the Dome.

THE DOME

Building the Dome itself presented quite a few challenges that had to be overcome early on in the development process. The first decision made in planning was whether or not to pursue a panel-type construction (made with solid sheet-type materials) or a skeletal frame construction (made with tubular framing material). After thoroughly exploring both options, the decision was made to proceed with a panel-type construction because of the immediate availability and low cost of the materials involved. The Dome's skin is composed of extruded insulation foam with foiled lining and was obtained locally in 4' x 8' sheets. In order to control costs, a variety of sizes for each of the Dome's 110 triangles was considered in order to minimize waste when cutting these large sheets. This resulted in the use of 11 large panels and a dome that now spans a 10' diameter and 7.5' height at its tallest point. These dimensions have proven to be quite satisfactory for the activities that are conducted in the Dome.

BIT DOME RENDERING



AN EARLY CONCEPT MODEL OF THE DOME

RGB LEDs & THE RAINBOWDUINO

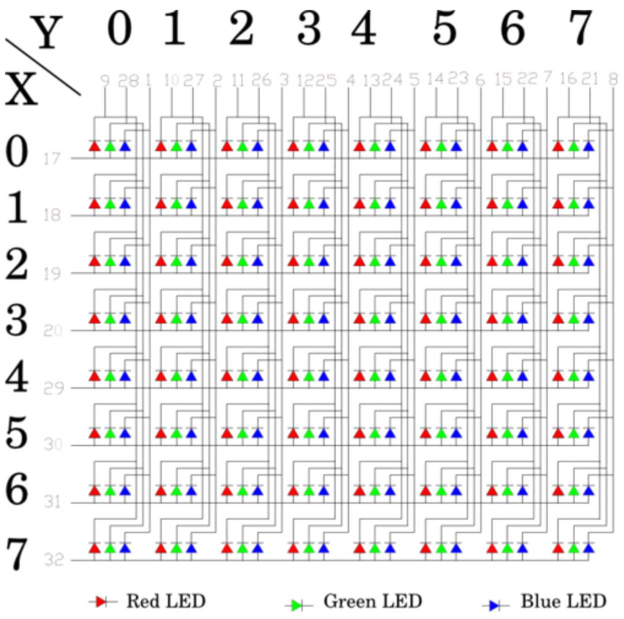
The principle method of interaction with the Bit Dome occurs through its 61 pixel (i.e. light) interface. Each pixel occurs at the vertex of each of the 110 triangles in the 3V series geodesic dome. Controlling this number of individual LEDs (183 in all when considering each R, G and B LED) can be troublesome, especially when communicating with the physical hardware over a serial port. For this reason, it was apparent early on that a traditional Arduino and shift registers would not be sufficient to match the required amount of data that would be sent (61 lights with 24-bit color resolution) and the speed (30+ frames per second).

To overcome these limitations of the traditional Arduino hardware, a Rainbowduino was procured that is specially designed to drive 64 RGB LEDs. It accomplishes this by allowing a user to arrange 64 LEDs in an 8 x 8 grid. The RGB cathodes make up the common columns, while each LED's anode makes up a row. By controlling current to these respective rows and columns, it becomes possible to individually address and control an LED.

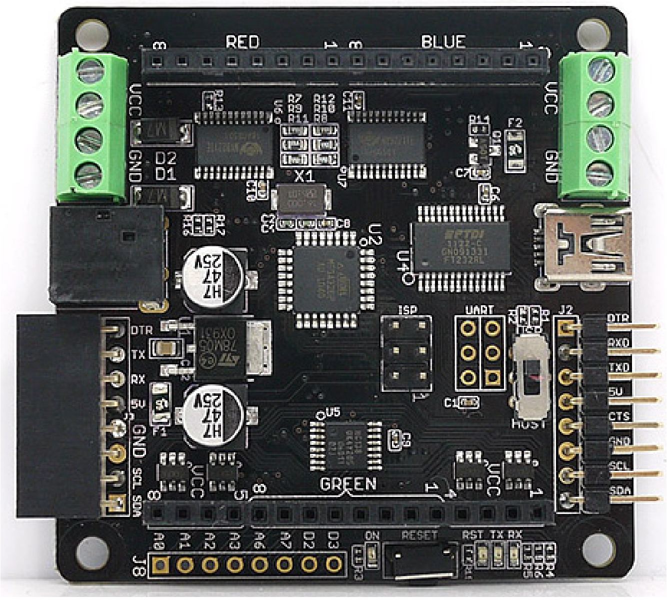
While the Rainbowduino included a very simple library that would allow for elementary communication with the device, it was necessary to build on top of this library to meet the desired performance benchmarks. A Java library had previously been written that met this criteria, though early efforts to establish communication between this library, Processing (the control software) and the Rainbowduino were fruitless. After consulting with the library's designer and the course instructor, communication was established and software development of the Dome controller could begin.

Furthermore, it became obvious that an external power supply would not be needed to power the LEDs in the dome as the amount of current supplied from the connected laptop. Attaching a power source to the Rainbowduino provided no noticeable increase in performance or brightness.

8X8 RGB LED SCHEMATIC



RAINBOWDUINO V3 MICROCONTROLLER



XBOX KINECT

In order to achieve interactivity, an input device was required to be installed inside the dome. Given the nature of the dome, it made sense that it should be capable of two things: tracking a human body and three dimensional imaging. Given the budgetary considerations (and the limit thereof) for this exercise, the very affordable and easily attainable XBox Kinect was chosen as the primary source of input.

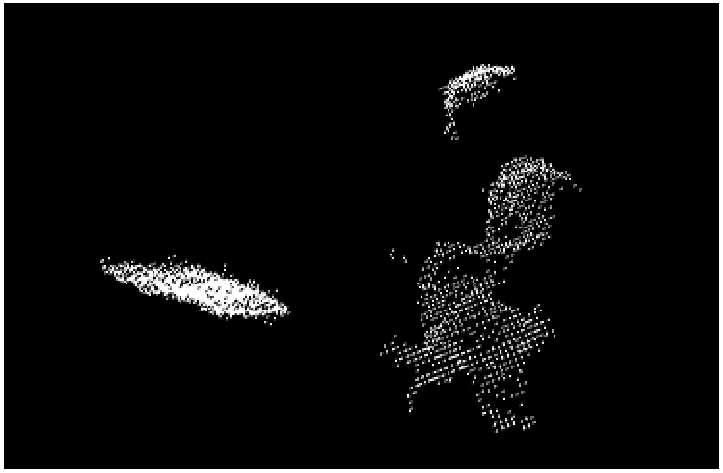
The Kinect is a device that allows a program to obtain a depth map of a scene within view of the device using the combination of a infrared camera and an infrared laser projector. Using a specified pattern of reference points, these two elements are able to determine how far away objects are that are within its operable range. The relatively small size of the dome (when compared to the average family room for which the Kinect was designed) made it safe to assume that the Kinect would function adequately in the given environment. After a variety of preliminary tests, the original concept of installing the Kinect in the top of the dome was abandoned because of the close proximity of the users head while standing in the dome to the camera eclipsed the Kinect's field of vision, rendering any depth information gathered unusable. With this in mind, the Kinect was repositioned to a spot on the inner wall of the dome with the thought that the user could stand opposite of it and interact with it freely.

AUDIO

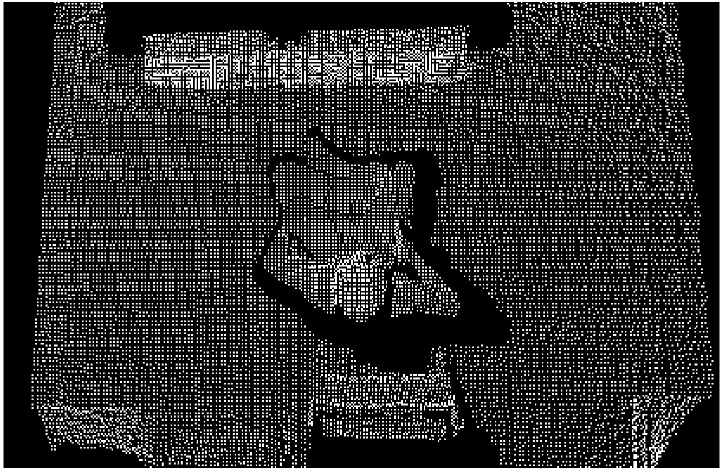
Early on in the development phase of the Bit Dome, it became apparent that in addition to interactive lights, the addition of sound and music to the Dome would support the projects goals of providing an immersive atmosphere. This proved to later be an advantageous decision because speech synthesizing was later added so that the Dome could instruct a user through the calibration phase of the dome in addition to given a variety of prompts and instructions as a user navigated between programs or accomplished certain tasks. A consumer grade amplifier and an array of speakers surrounding the dome were added to create a pleasing sound.

XBOX KINECT

PRIMARY KINECT DEPTH MAP TRIAL (SIDE VIEW)



PRIMARY KINECT DEPTH MAP TRIAL (TOP VIEW)



SOFTWARE

RAINBOWDUINO V3 STREAMING FIRMWARE

While the Dome is running, two separate systems are running alongside each other. The control computer acting as the master, handles all the heavy calculation and control programming whereas the Rainbowduino acts as a slave and merely receives an array of 64 RGB values for it to push out to the LEDs connected to the Dome. The firmware running on the Rainbowduino is distributed alongside the Rainbowduino hardware and has been slightly modified for increased performance and high speed serial communication. It features a variety of perks including buffering to decrease the likelihood of flickering or low refresh rates.

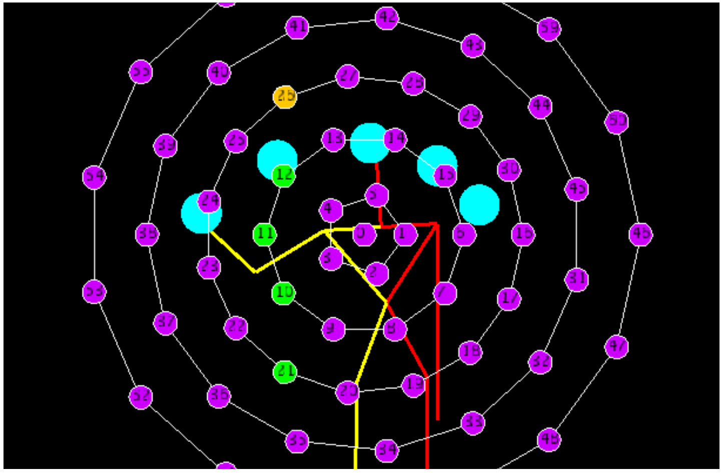
PROCESSING

To act as a control host between the input coming in from the XBox Kinect and the output to the Rainbowduino (and subsequently the Dome's light) and the speakers, a computer running Processing (version 1.5.1) is employed. Processing was an obvious choice because of the already existing body of work surrounding the gathering of input from the Kinect. Additionally it was a platform frequently discussed and used during the Physical Computing course (as was the Arduino and therefore inherently the Rainbowduino). Furthermore, because the physical building of the Dome was temporarily delayed while waiting for components from China, a simulator program was designed in Processing such that software could be developed and tested on the Processing platform before a large amount of time was invested in building the physical manifestation of the idea. This simulator remains in place as part of the final project as a way for outside observers to witness what a user inside the Dome is experiencing.

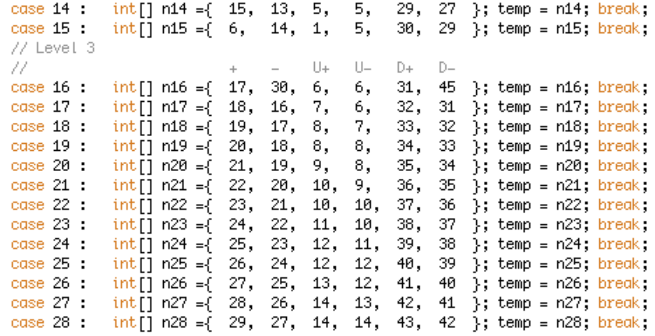
PIXEL MAPPING

Because the circuit representation of the lights as an 8 x 8 grid did not sensibly map to the topographical or virtual representation of the lights (which represented something

SKELETON CALIBRATION AND TRACKING

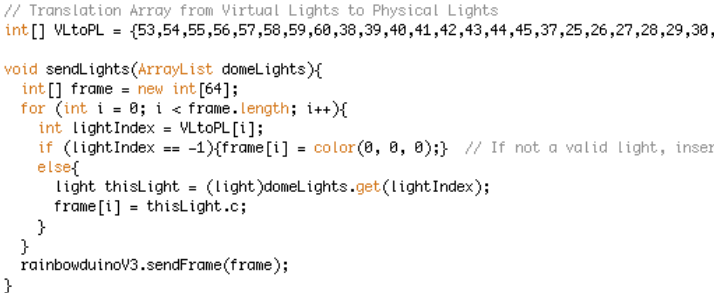


LED NEIGHBOR ARRAY CODE SAMPLE



```
case 14 :   int[] n14 ={  15,  13,  5,   5,   29,  27  }; temp = n14; break;
case 15 :   int[] n15 ={  6,   14,  1,   5,   30,  29  }; temp = n15; break;
// Level 3
//                        +    -    U+   U-   D+   D-
case 16 :   int[] n16 ={  17,  30,  6,   6,   31,  45  }; temp = n16; break;
case 17 :   int[] n17 ={  18,  16,  7,   6,   32,  31  }; temp = n17; break;
case 18 :   int[] n18 ={  19,  17,  8,   7,   33,  32  }; temp = n18; break;
case 19 :   int[] n19 ={  20,  18,  8,   8,   34,  33  }; temp = n19; break;
case 20 :   int[] n20 ={  21,  19,  9,   8,   35,  34  }; temp = n20; break;
case 21 :   int[] n21 ={  22,  20,  10,  9,   36,  35  }; temp = n21; break;
case 22 :   int[] n22 ={  23,  21,  10,  10,  37,  36  }; temp = n22; break;
case 23 :   int[] n23 ={  24,  22,  11,  10,  38,  37  }; temp = n23; break;
case 24 :   int[] n24 ={  25,  23,  12,  11,  39,  38  }; temp = n24; break;
case 25 :   int[] n25 ={  26,  24,  12,  12,  40,  39  }; temp = n25; break;
case 26 :   int[] n26 ={  27,  25,  13,  12,  41,  40  }; temp = n26; break;
case 27 :   int[] n27 ={  28,  26,  14,  13,  42,  41  }; temp = n27; break;
case 28 :   int[] n28 ={  29,  27,  14,  14,  43,  42  }; temp = n28; break;
```

VIRTUAL TO PHYSICAL LED TRANSLATION



```
// Translation Array from Virtual Lights to Physical Lights
int[] VLtoPL = {53,54,55,56,57,58,59,60,38,39,40,41,42,43,44,45,37,25,26,27,28,29,30,

void sendLights(ArrayList domeLights){
  int[] frame = new int[64];
  for (int i = 0; i < frame.length; i++){
    int lightIndex = VLtoPL[i];
    if (lightIndex == -1){frame[i] = color(0, 0, 0);}  // If not a valid light, inser
    else{
      light thisLight = (light)domeLights.get(lightIndex);
      frame[i] = thisLight.c;
    }
  }
  rainbowduinoV3.sendFrame(frame);
}
```

SOFTWARE

more closely related to a polar graph) it became necessary to abstract the relationship of lights to one another such that they could be individually addressed in software, interact with each other, and correctly be translated and packaged in a form that could be sent to the Rainbowduino Firmware. To do this an translational map of the Bit Dome was created in Google Sketchup as a flattened top view, making organization and optimization easier. After this translational implementation was established, simple functions were created such that a given light's neighbors could quickly be found, in addition to its individual properties as an object in the Light class.

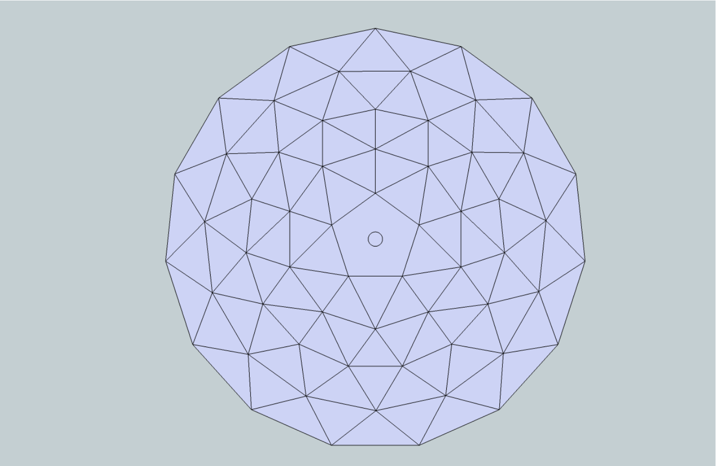
SIMPLE OPENNI

One of the perks of using the Processing platform was the inclusion and support of a wide variety of software libraries. Simple OpenNI, for example allows for seamless retrieval of depth information from the Kinect. Additionally, it supports gesture recognition and skeleton/joint tracking which greatly enhanced and sped up the process of establishing human interaction with individual pixels in the Dome. The body and gesture information gathered from OpenNI was stored in a class where it could easily be accessed by any function that was presently running.

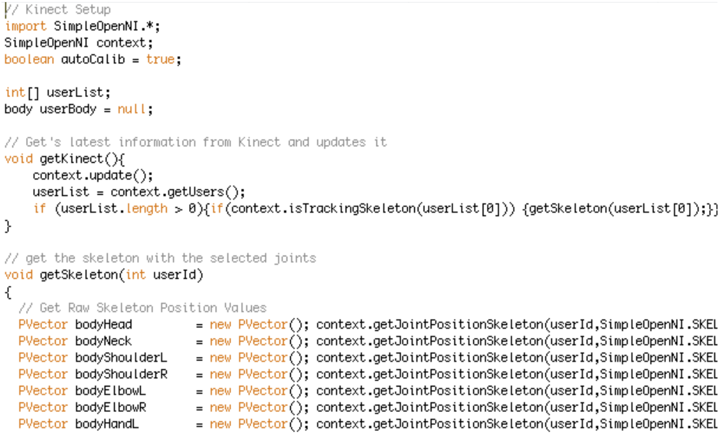
DOME TWEENING

While a variety of unique functions were developed for the Bit Dome to executive its variety of functions and visualizations, the most novel among them deserves special mention. While developing the Sunrise / Sunset program where the sky (in particular the sun) transitioned from one state to another, it became obvious that "digitally" staggering through different scenes (light patterns in the Dome) was undesirable. For example, when the sun would move from one location to the next, it would either be off or on in a given location. This led me to consider the concept of color tweening, the gradual fading from one color to another. Using this principle,

LED NEIGHBOR DOME TEMPLATE



OPENNI BODY TRACKING CODE SAMPLE



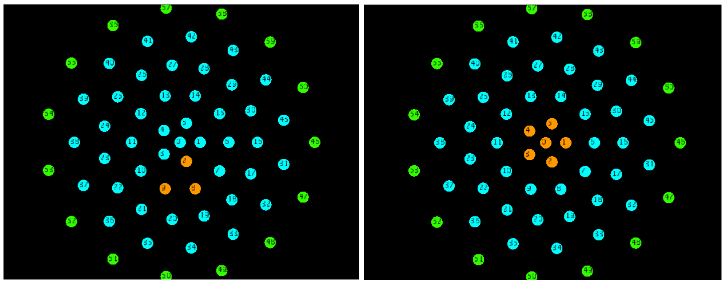
```
// Kinect Setup
import SimpleOpenNI.*;
SimpleOpenNI context;
boolean autoCalib = true;

int[] userList;
body userBody = null;

// Get's latest information from Kinect and updates it
void getKinect(){
    context.update();
    userList = context.getUsers();
    if (userList.length > 0){if(context.isTrackingSkeleton(userList[0])) {getSkeleton(userList[0]);}}
}

// get the skeleton with the selected joints
void getSkeleton(int userId)
{
  // Get Raw Skeleton Position Values
  PVector bodyHead         = new PVector(); context.getJointPositionSkeleton(userId,SimpleOpenNI.SKEL
  PVector bodyNeck         = new PVector(); context.getJointPositionSkeleton(userId,SimpleOpenNI.SKEL
  PVector bodyShoulderL    = new PVector(); context.getJointPositionSkeleton(userId,SimpleOpenNI.SKEL
  PVector bodyShoulderR    = new PVector(); context.getJointPositionSkeleton(userId,SimpleOpenNI.SKEL
  PVector bodyElbowL       = new PVector(); context.getJointPositionSkeleton(userId,SimpleOpenNI.SKEL
  PVector bodyElbowR       = new PVector(); context.getJointPositionSkeleton(userId,SimpleOpenNI.SKEL
  PVector bodyHandL        = new PVector(); context.getJointPositionSkeleton(userId,SimpleOpenNI.SKEL
```

DOME SCENE TRANSITION (NO TWEENING)



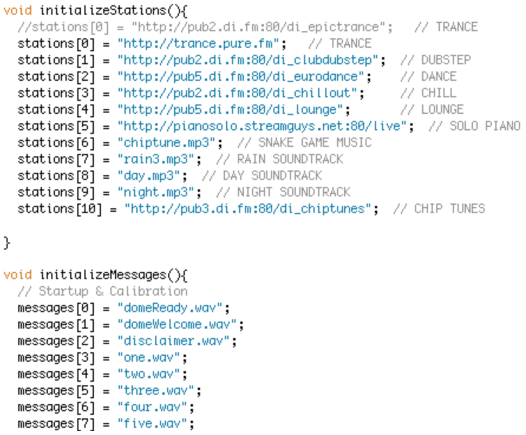
SOFTWARE

I created an abstraction that allowed me to define two dome scenes, one a start state and another the end state. The dome tweening function would then calculate intermediate domes that created a seamless transition between these two states.

MINIM

Minim is a Processing library that allows a sketch to import, play and analyze music. This library was instrumental in creating an atmosphere for almost every single program that can be run in the Bit Dome. Advanced functionality came in the form of beat and intensity detection to dynamically influence the visualizations of the Music programs, in addition to the raindrops and lightning strikes in Rainstorm. One particularly useful function discovered during the development of the Dome was the ability of Minim to not just process static sound files stored on the operating computer, but also the ability to stream Internet radio stations. This greatly expanded the offerings available in the Music section with a virtually limitless supply of songs that can be played in continuous succession.

AUDIO LIBRARY CODE SAMPLE



```
void initializeStations(){
  //stations[0] = "http://pub2.di.fm:80/di_epictrance";   // TRANCE
  stations[0] = "http://trance.pure.fm";   // TRANCE
  stations[1] = "http://pub2.di.fm:80/di_clubdubstep";  // DUBSTEP
  stations[2] = "http://pub5.di.fm:80/di_eurodance";    // DANCE
  stations[3] = "http://pub2.di.fm:80/di_chillout";     // CHILL
  stations[4] = "http://pub5.di.fm:80/di_lounge";       // LOUNGE
  stations[5] = "http://pianosolo.streamguys.net:80/live";  // SOLO PIANO
  stations[6] = "chiptune.mp3";  // SNAKE GAME MUSIC
  stations[7] = "rain3.mp3";  // RAIN SOUNDTRACK
  stations[8] = "day.mp3";  // DAY SOUNDTRACK
  stations[9] = "night.mp3";  // NIGHT SOUNDTRACK
  stations[10] = "http://pub3.di.fm:80/di_chiptunes";  // CHIP TUNES

}

void initializeMessages(){
  // Startup & Calibration
  messages[0] = "domeReady.wav";
  messages[1] = "domeWelcome.wav";
  messages[2] = "disclaimer.wav";
  messages[3] = "one.wav";
  messages[4] = "two.wav";
  messages[5] = "three.wav";
  messages[6] = "four.wav";
  messages[7] = "five.wav";
```

THANK YOU

I would like to thank Dr. Nadeem Hamid and Markus Lang for their many contributions in establishing communication between Processing and the Rainbowduino. Thank you to Dr. John Grout for his constant advice and suggestions in building the dome. Finally, thank you to the many visitors who continuously made suggestions, tested and enjoyed the dome while it was being developed.