

University
of Idaho

S.N.A.P.

Andrew Rose

Dustin Fox

Dylan Carlson

Project S.N.A.P.'s Problem Statement

Our goal was to create an easily distributable standard testing environment for gathering data and metrics to find the best possible acoustic navigation algorithm.

Audio Algorithm Specifications

Visual-Audio-Engine

- Allow multiple audio navigation algorithms to be tested on a single platform.
- Optimized shared memory block communication.

Configurations

- Create configuration files that can be saved, loaded and shared.
- Create a configuration menu to adjust settings.

Unity Test Bed Specifications

Logging

- Log information from each test session.
- Allow easy data export for analysis.

Maps/Tests

- Create stochastic maps that can be used to test navigation.

Installation

- Must include an easy installation/build script that minimizes the need for user input and installs all required .dll files.

Our Goals

- Modular and Modifiable for future teams.
- Well documented.
- Versatile and ready for future technologies and research.
- Clean and well set up project repository to keep all resources in one place.

Constraints

- Restricted ourselves to using purely open source plugins/libraries/tools to allow for simple licensing.
 - OpenCV
 - OpenAL
- Free version of Unity is acceptable for solo/indie developers that are making below \$100,000.



Main Menu - Initial Design



- No set color scheme
- Not as user friendly
- Very buggy
- Not able to open the file browser window
- Couldn't load audio configurations



Main Menu - Final Design

S.N.A.P.

New Configuration

Load Configuration

Map Settings

Logs

About

Start

Quit

- Set color scheme
- Very user friendly
- Users can easily create new audio configurations, load previous ones, alter different aspects of the map, view logs from previous tests and start a new test.

Logging



Logs

Logs:

Date: 04/25/2018	Time: 10.67377
Hallway: Hallway Map	Collisions: 0
Date: 04/25/2018	Time: 9.729893
Hallway: Hallway Map	Collisions: 0
Date: 04/25/2018	Time: 13.02383
Hallway: Hallway Map	Collisions: 4
Date: 04/25/2018	Time: 13.04382
Hallway: Hallway Map	Collisions: 0
Date: 04/25/2018	Time: 10.79847
Hallway: Hallway Map	Collisions: 3
Date: 04/25/2018	Time: 10.81847
Hallway: Hallway Map	Collisions: 0

Filter:

All

[Back](#)

- After each test is completed a log file is created with information about your test.
- Information includes:
 - Date of test
 - Map used
 - How many collisions
 - Time it took to complete test

Audio Configurations



Configuration

Frequency Min:

Frequency Max:

Horizontal Resolution:

Vertical Resolution:

Field of View:

Sample Length:

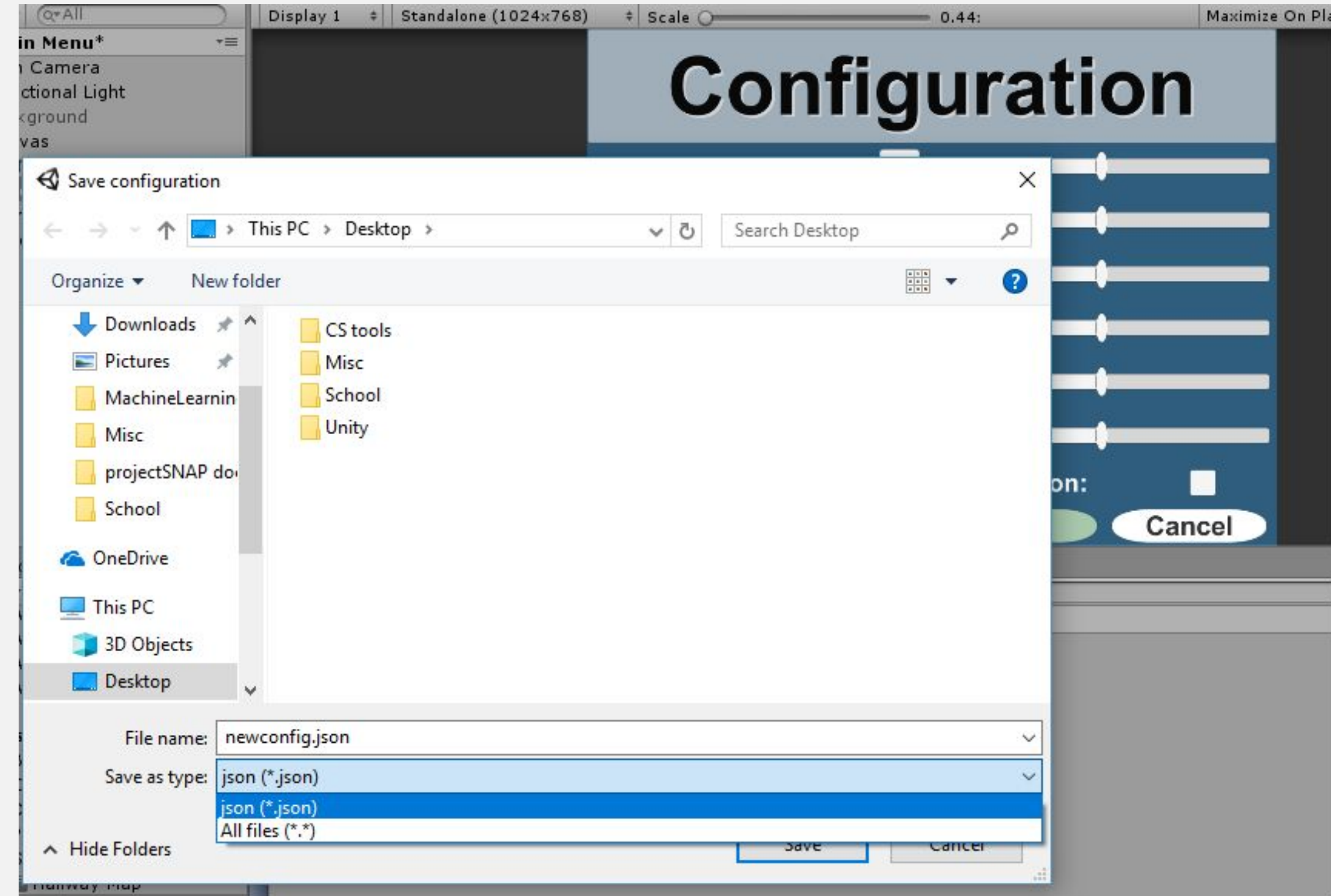
Scanning Type:

- Allows users to alter how they view the world through sound.
- Audio configurations are saved as JSON files, that can be shared.
- You can save and load different audio configurations.

Audio Configurations - Save/Load



```
newconfig.json
1 {"savedFrequency":25.0,
2  "savedFrequencyIncrement":5.0,
3  "savedHorizontalSteps":5.0,
4  "savedStepDelay":5.0,
5  "savedAudioSpreadDegree":45.0,
6  "savedAudioVolumeRollOff":25.0,
7  "savedScanningType":0,
8  "savedVision":false }
```



- A look at how audio configuration files look in .JSON format and how easily they can be saved.

Map Settings

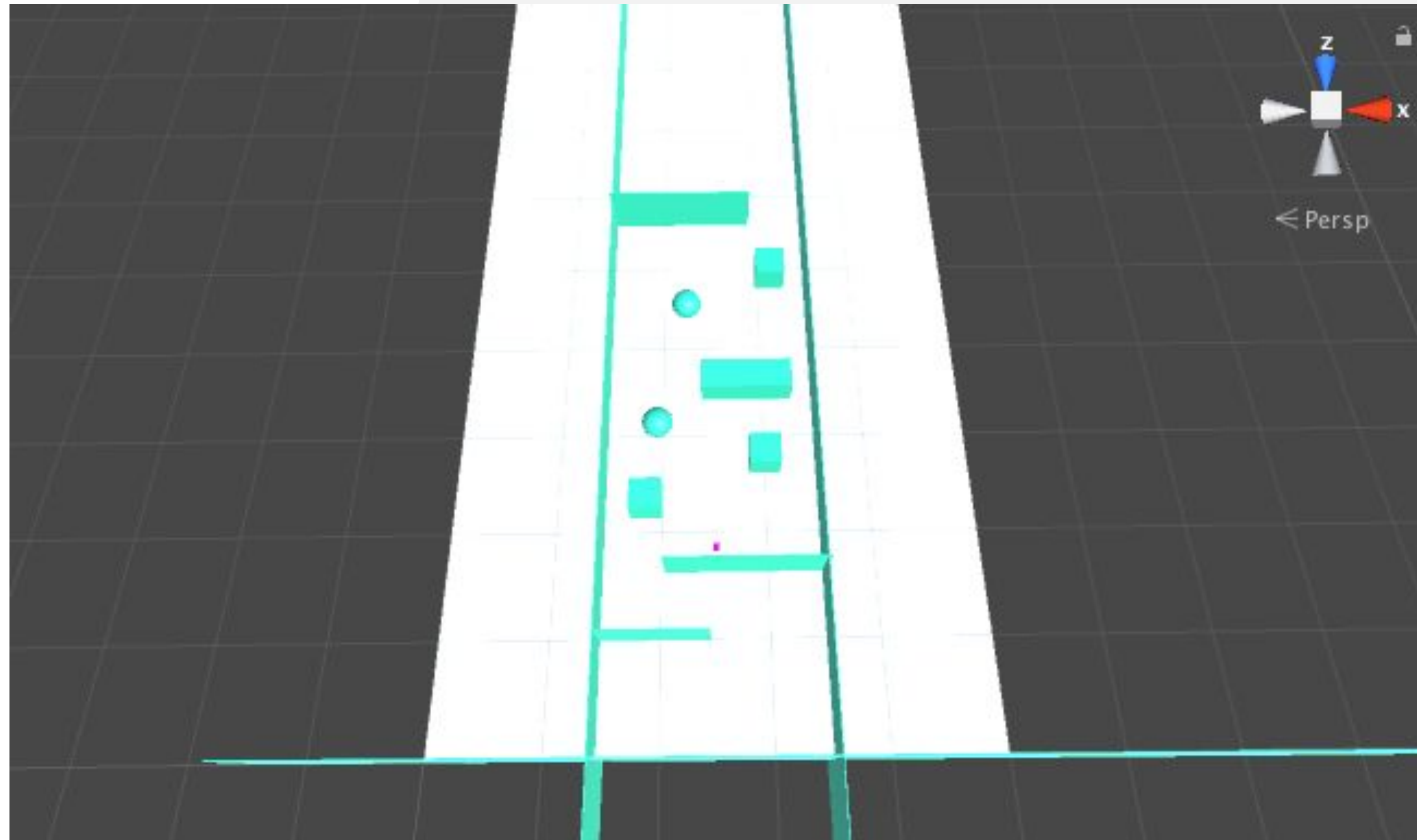


Map Settings

Cubes Spawned:	15	<input type="range"/>
Spheres Spawned:	15	<input type="range"/>
Minimum Cube Size:	2	<input type="range"/>
Maximum Cube Size:	5	<input type="range"/>
Minimum Sphere Size:	7	<input type="range"/>
Maximum Sphere Size:	5	<input type="range"/>
Map Selection:	Hallway	<input type="button" value="Apply"/> <input type="button" value="Cancel"/>

- Allows users to change the map before they enter a test.
- Users can also select which map they would like to load.

Testing Map - Initial Design

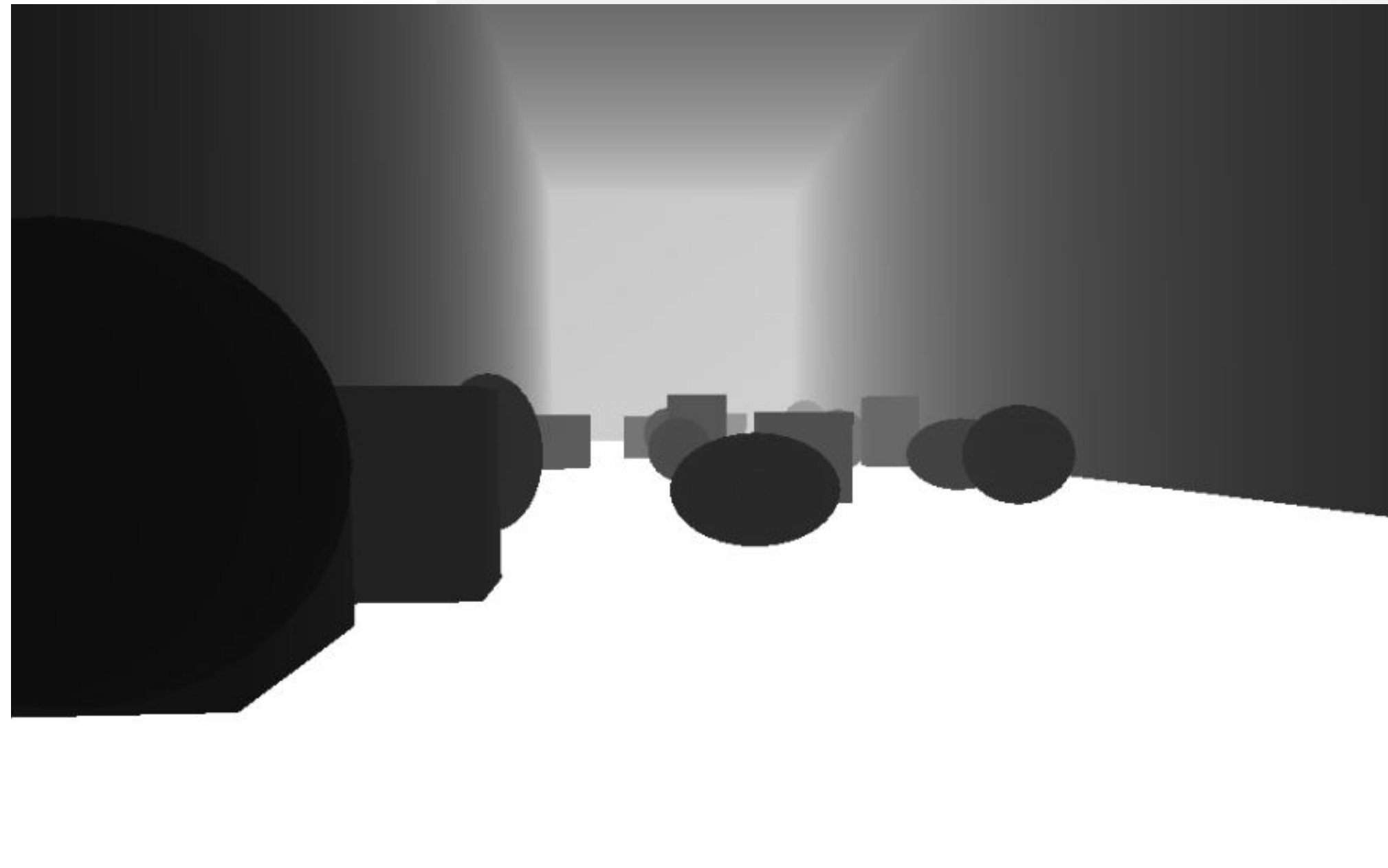


- No randomly generated obstacles.
 - Users can easily memorize a map
- Map was way too short.
- Didn't implement a scene change once the map was complete.

Testing Map - Final Design

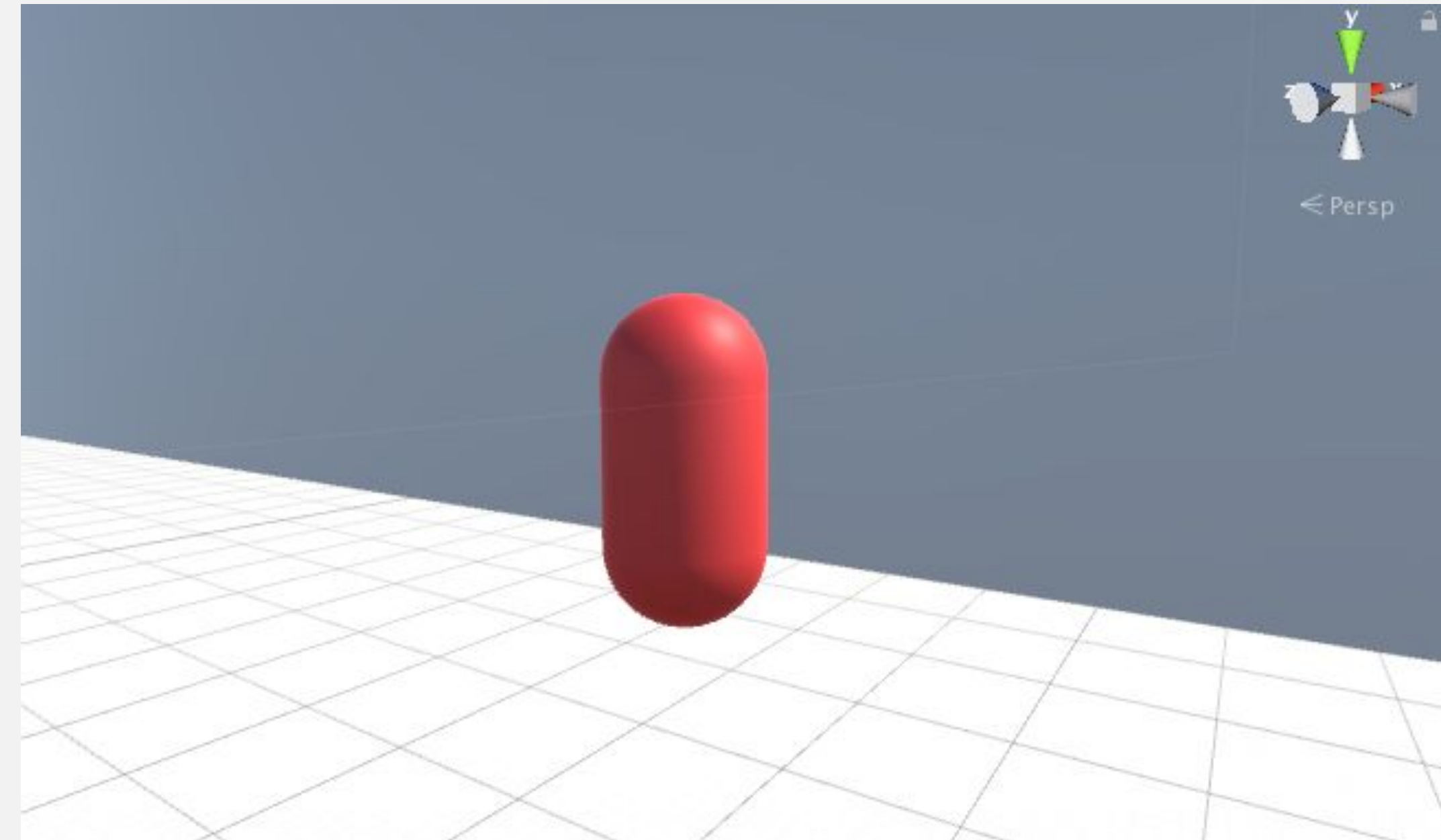


Static Map



- Spawns randomly generated object so that each test is different

Dynamic Map



- Spawns randomly moving cylinders to simulate avoiding other people.
- Also includes randomly generated static objects.

Testing Map - Final Design



Test Complete

Results

Collisions: 0

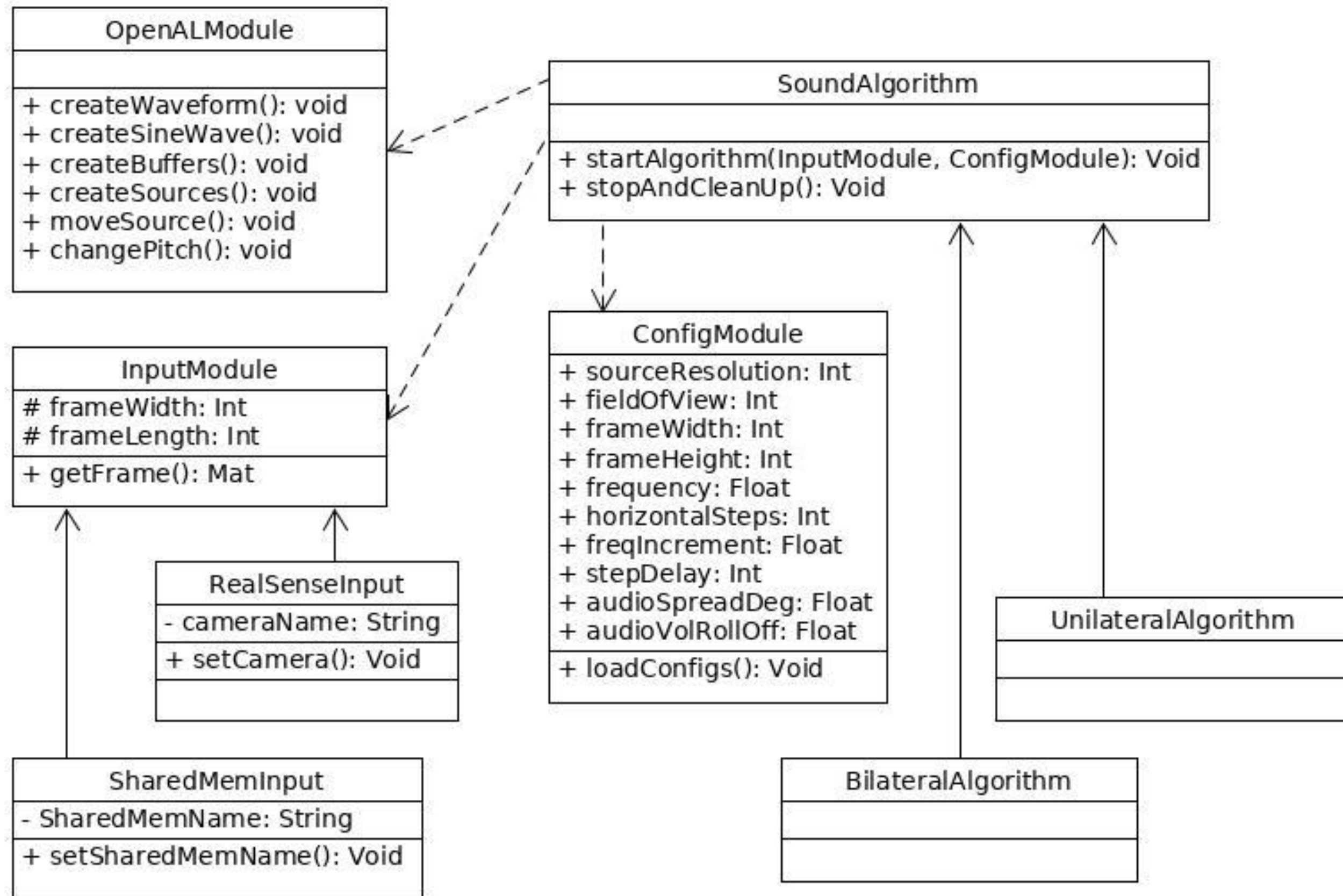
Time: 10.06 sec

New Test

Main Menu

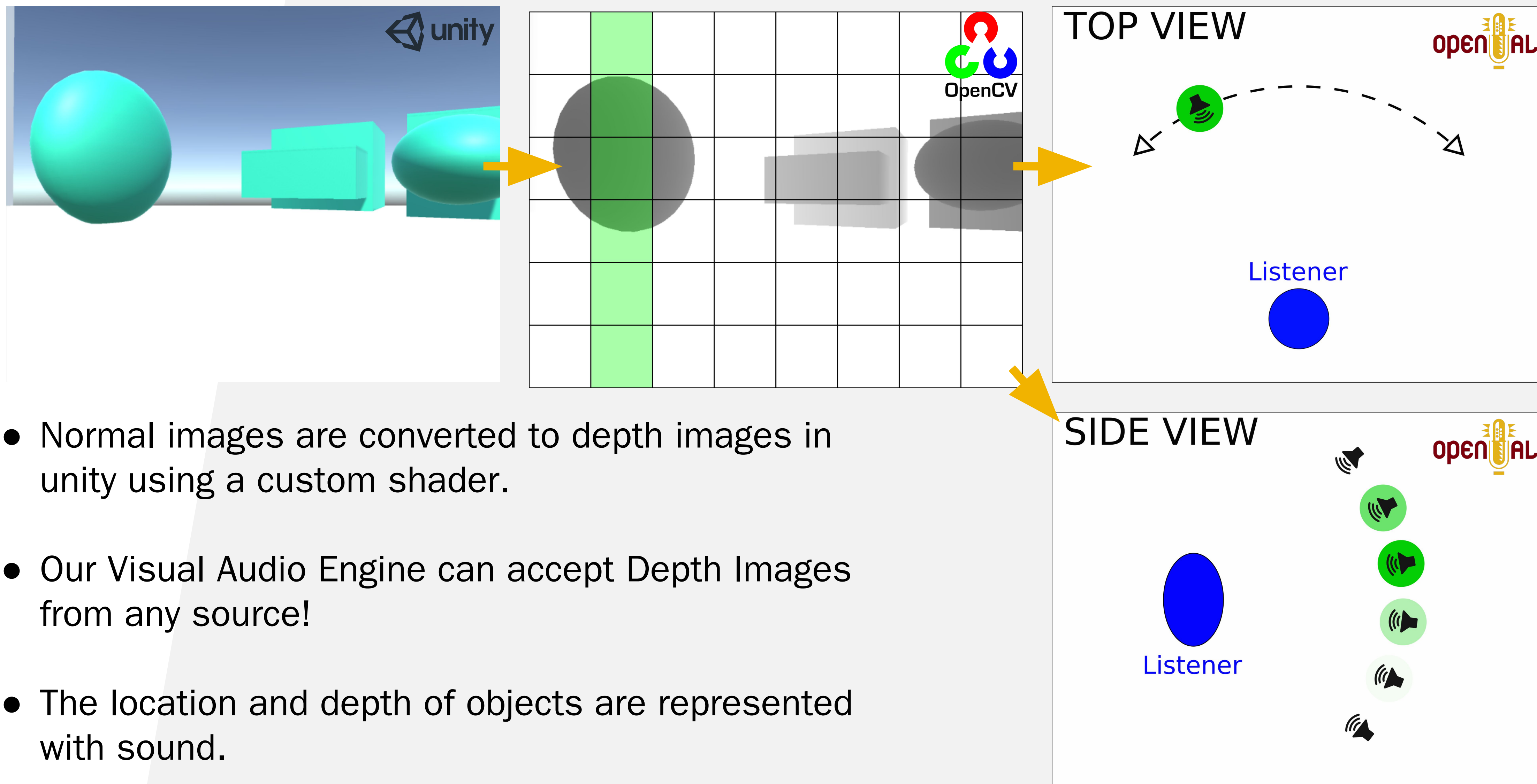
- At the end of every map a test complete window now appears.
- The test complete window prompts the user:
 - Total collisions
 - Total time of test
 - Ability to perform a new test
 - Ability to return to main menu

Visual Audio Engine (V.A.E)



- A modular framework for building and testing vision to sound algorithms utilizing openCV and openAL.
- Reimplemented the previous group's algorithm with more configurability.
- Can accept depth information from cameras or other programs such as our test-bed.
- Built to make it easier to continue development and research.

Visual Audio Engine (V.A.E)



Easy Installation Process



```
64Bit32 - Inno Setup Compiler 5.5.9 (64)
File Edit View Build Run Tools Help
-- 64Bit.iss --
; Demonstrates installation of a program built for the x64 (a.k.a. AMD64)
; architecture.
; To successfully run this installation and the program it installs,
; you must have a "x64" edition of Windows.
; SEE THE DOCUMENTATION FOR DETAILS ON CREATING .ISS SCRIPT FILES!

[Setup]
AppName=My Program
AppVersion=1.5
DefaultDirName={pf}\My Program
DefaultGroupName=My Program
UninstallDisplayIcon={app}\MyProg.exe
Compression=lzma2
SolidCompression=yes
OutputDir=userdocs:Inno Setup Examples Output
; "ArchitecturesAllowed=x64" specifies that Setup cannot run on
; anything but x64.
ArchitecturesAllowed=x64
; "ArchitecturesInstallIn64BitMode=x64" requests that the install be
; done in "64-bit mode" on x64, meaning it should use the native
; 64-bit Program Files directory and the 64-bit view of the registry.
ArchitecturesInstallIn64BitMode=x64

[Files]
Source: "MyProg-x64.exe"; DestDir: "{app}"; DestName: "MyProg.exe"
Source: "MyProg.chm"; DestDir: "{app}"
Source: "Readme.txt"; DestDir: "{app}"; Flags: isreadme

[Icons]
Name: "{group}\My Program"; Filename: "{app}\MyProg.exe"
```

- Through the use of Inno Setup we are able to create an easy installation process for future users
- Allows for the installation of necessary plugins such as:
 - OpenAL
 - OpenCV

Conclusion



Over the course of this project our group has learned:

- The Unity engine.
- Intensive version control using Git.
- Utilizing a shared memory space between concurrently running executables.
- Designing and Implementing full scale object oriented projects.
- Agile Development

Recommendations



For future development on this project, our group recommends investigating into:

- Different types of depth cameras for the hardware component.
- Experimenting with other sound libraries (non-open source).
- Different techniques for analyzing depth images.



Thank you!

Any Questions?