LYU 1702 AR Game with Tango

TANG DING 1155047074
TANG XUN 1155046996

1

Combines real and virtual

- Connect virtual world with real world, both spatially and cognitively. The virtual information appears to become part of the real world

2

Interactive in real time

- Human–computer interface operates in a tightly coupled feedback loop

3

Registered in 3D

- Precise alignment of corresponding virtual and real information

Introduction to AR

Introduction to Tango

01

Augmented reality computing platform, developed and authored by Google.

02

It uses **computer vision** to enable mobile devices to detect their position relative to the world around, especially indoor space.

03

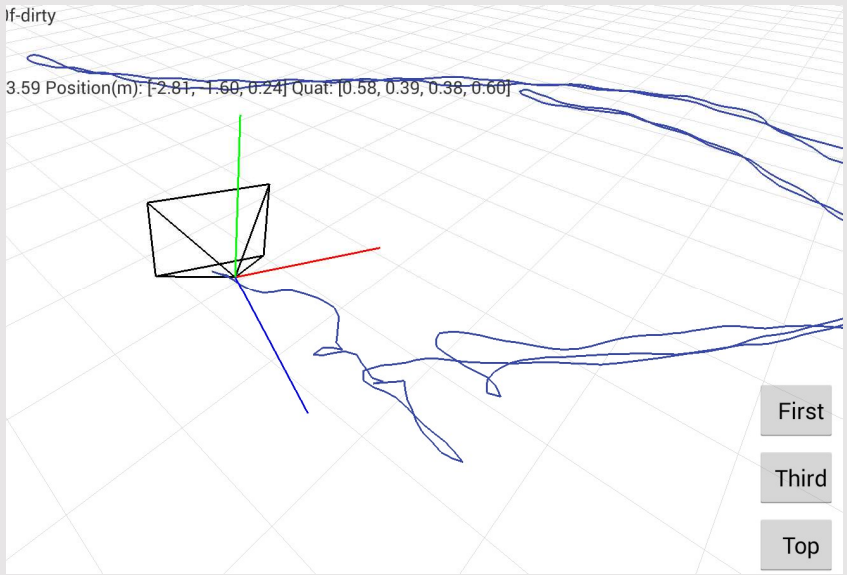
Provide C, Java and Unity API

04

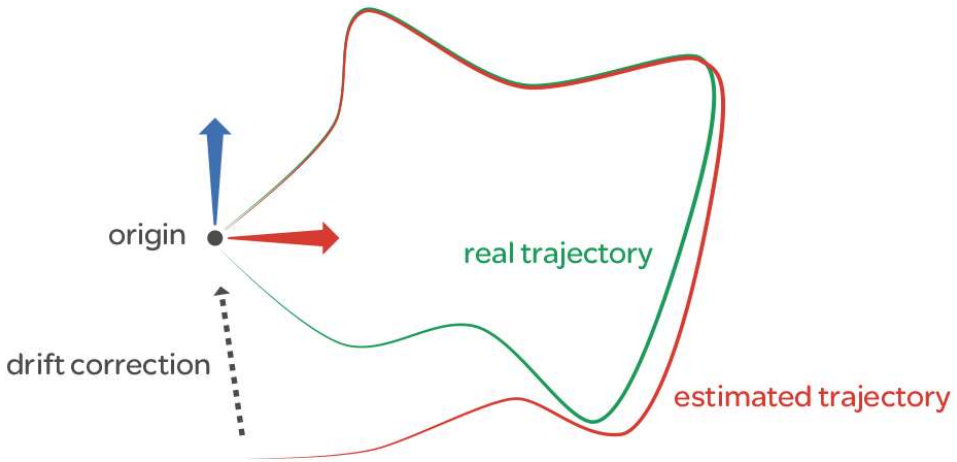
Need Tango-supported device with special hardware.

Motion Tracking

- Overview: Tango device can track its own movement and orientation through 3D space.
- How: visual-inertial odometry
  - Standard visual odometry: using camera images to determine a change in position by looking at the relative position of different features in those images
  - Inertial motion sensors: tracking a device's rotation and acceleration
- Limitation
  - No memory
  - Error Accumulation



Area Learning



- Overview: Tango device can recognize where they are in an environment by noticing the previously-seen features around them.
- How:
  - Generating a mathematical description of the edges, corners, other unique visual features
  - Inertial motion sensors: tracking a device's rotation and acceleration
  - drift corrections: realizing it has traveled in a loop and adjusting its path to be more consistent with its previous observations when the device sees a place it knows it has seen earlier in your session

Area Description Picker

lab1
39aaa609-f1b0-25e6-88a9-ba0f3c6f6fa5

Unnamed
876fb6a7-8f10-209e-8c53-9665ba97287d

1021
b09650bd-6aa3-2a45-8b20-bdc87effa199

1021b
b09650c5-6aa3-2a45-886b-75c81b228f1f

09:01:14
d16e7154-bc79-2be2-8ba9-8780c09203bb

room
ea5cb954-eb24-2e4c-8b69-beac99d8f9a2
Has Mesh Data

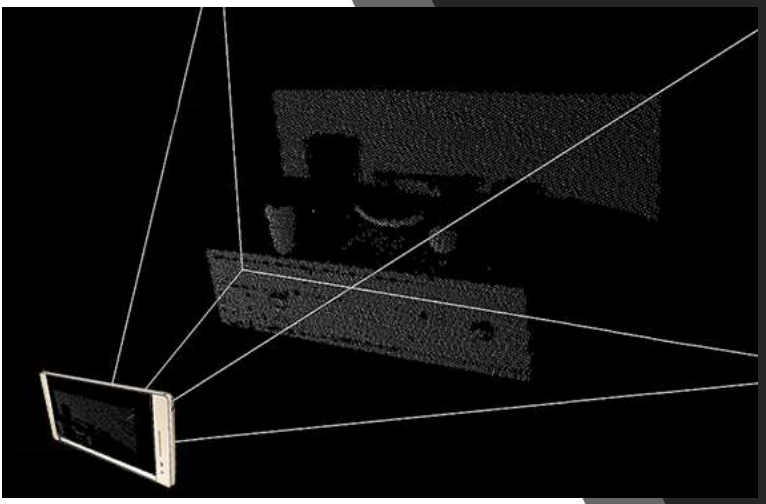
Delete All Meshes

Create new mesh and new Area Description

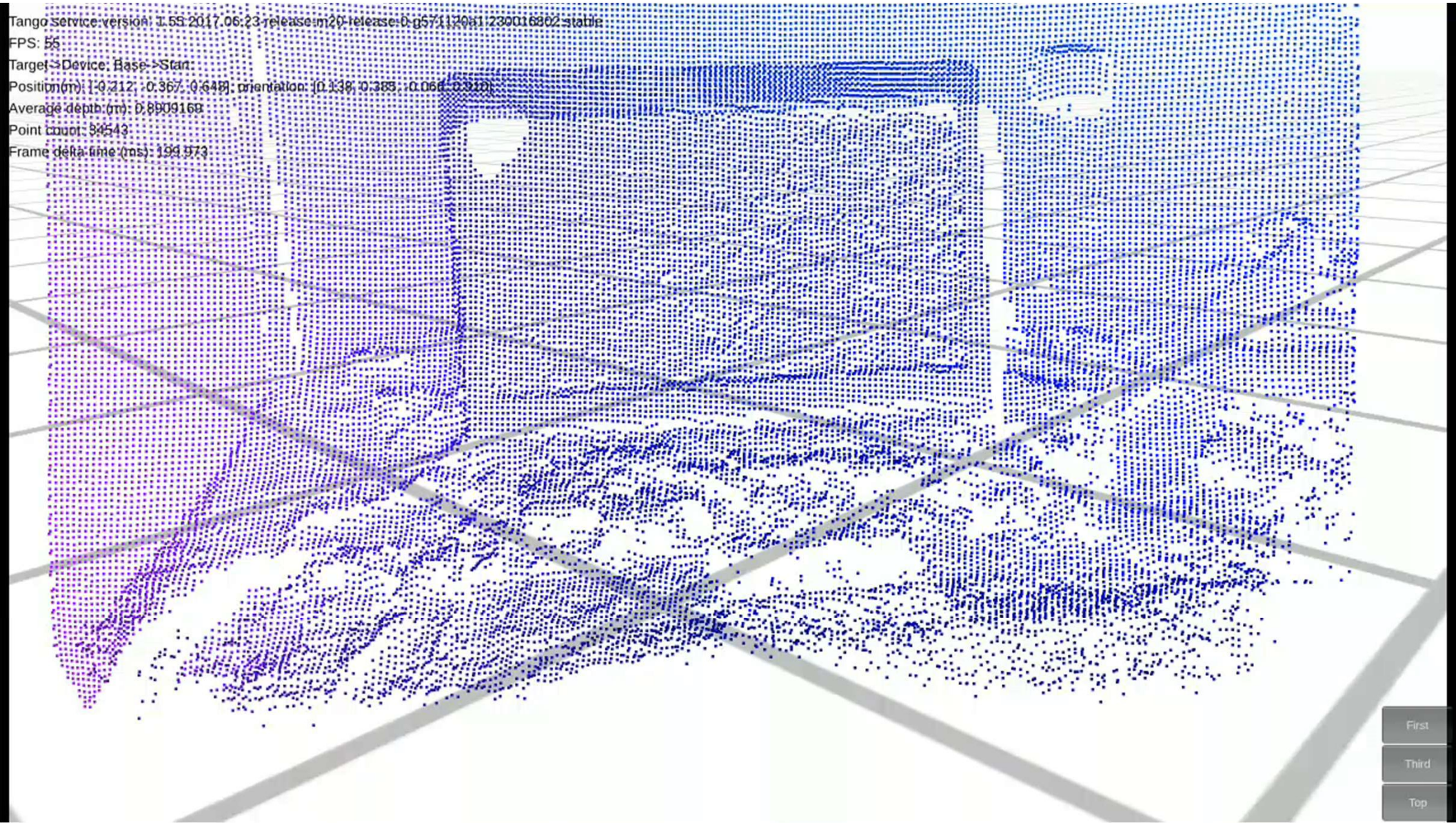
Create new mesh for selected Area Description

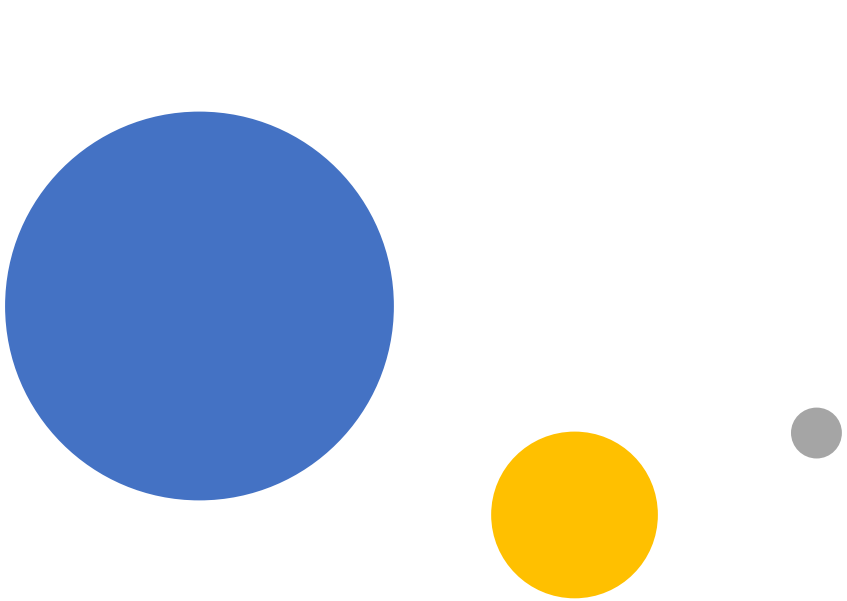
Start Game using Area Description with mesh

Depth Perception



- Overview: Tango device can estimate the distance to objects.
- How: depth technologies
  - Stereo, Structured Light:
    https://www.youtube.com/watch?v=mSsnf5tqXnA
  - Time of Flight
    https://www.youtube.com/watch?v=fzUIDIM3EsA
- Point Cloud
  - Data structure which stores depth information


Tango service version: 1.55 2017-06-23 release/m20 release-0-g57f1120a1 2300168d2 stable
FPS: 55
Target->Device; Base->Start:
Position(m): [-0.212, -0.367, 0.648]; orientation: [0.138, 0.388, -0.066, 0.910]
Average depth(m): 0.8909168
Point count: 34543
Frame delta time (ms): 199.973
First
Third
Top



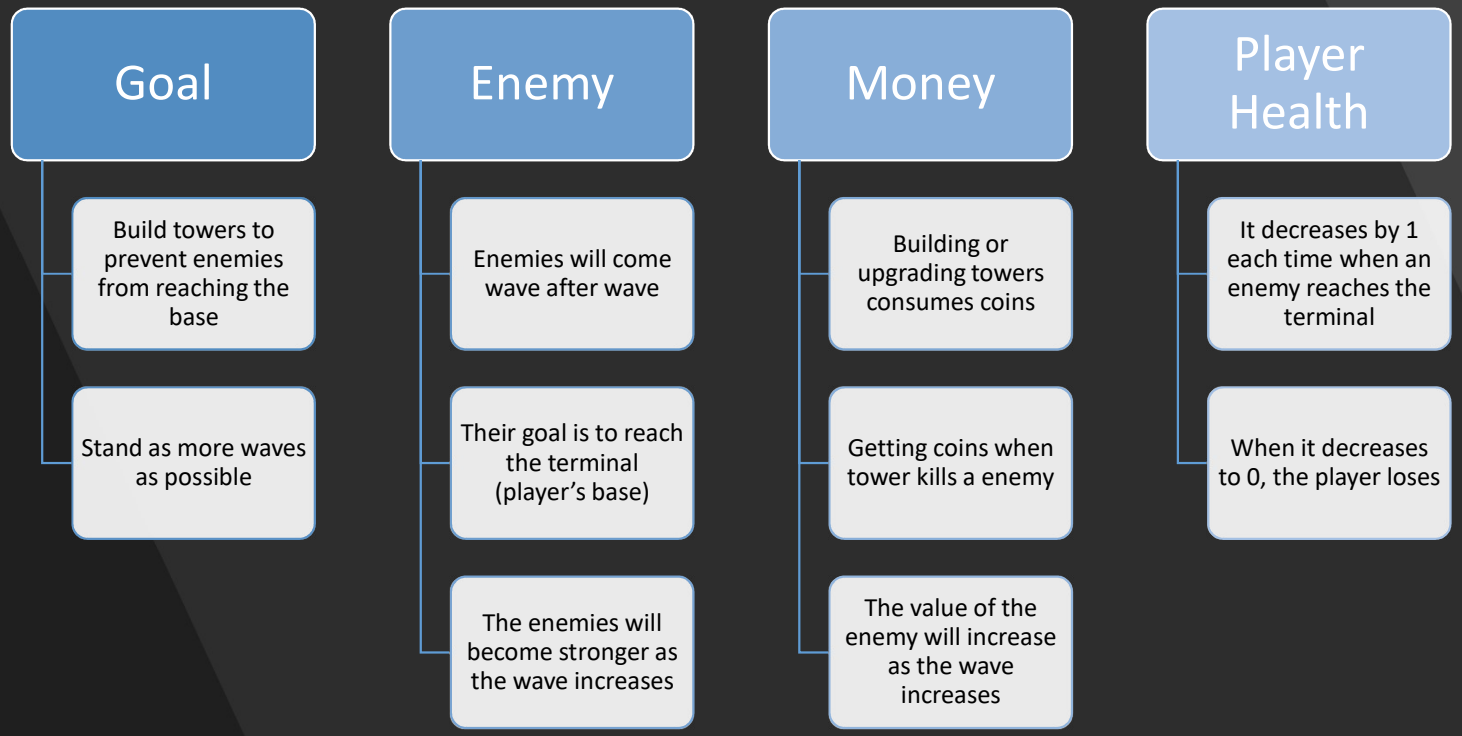
Demo

AR Tower Defense

Start


unity

Game Design


Goal
Build towers to prevent enemies from reaching the base
Stand as more waves as possible
Enemy
Enemies will come wave after wave
Their goal is to reach the terminal (player's base)
The enemies will become stronger as the wave increases
Money
Building or upgrading towers consumes coins
Getting coins when tower kills a enemy
The value of the enemy will increase as the wave increases
Player Health
It decreases by 1 each time when an enemy reaches the terminal
When it decreases to 0, the player loses

Map Generation

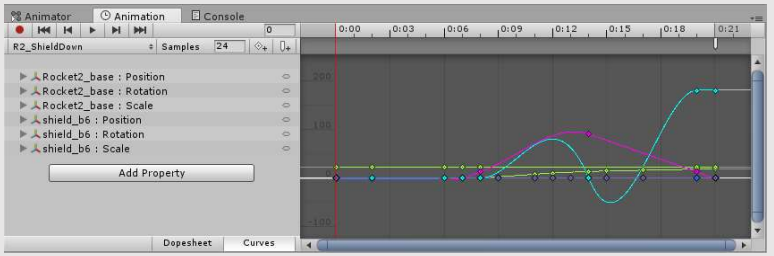
- Requirement:
  - Random
  - Fixed start and end
  - No dead end
- How
  - Initial map with cells and walls(there's wall between each two cells)
  - Depth-First-Search(DFS) from start to end, and record all dead ends
  - For each dead end, choose a random direction to the nearest non-dead end cell, and remove wall in path
- https://www.youtube.com/watch?v=S-1Eq6NL-NE

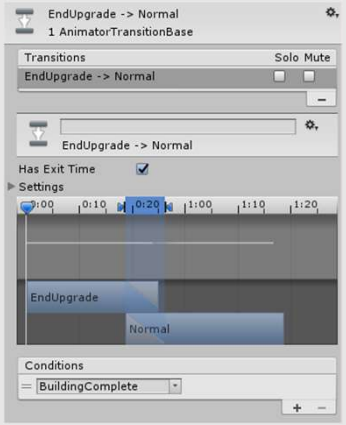
Navigation

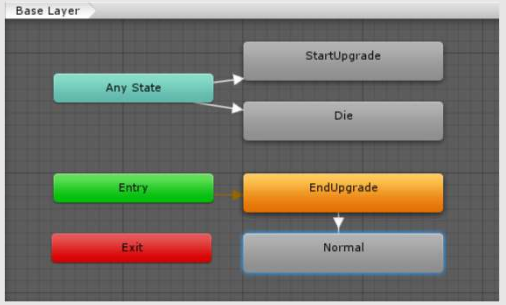
- D* lite
  - Compared to A*:
    - More efficient
    - Reacting to dynamic environment
  - Every time environment changes, A* will throw the previous information away, while D* will keep it.
- Not enough
  - Distributed the calculation tasks to multiple frames to make the rendering smoother.

Animation States Control

- The building and destroy of towers
  - An easy implementation of Animation Controller
    - Most of the animations are accessed by a simple curves of the position/rotation/scale
    - The transition of the states are controlled by the trigger, i.e. “EndUpgrade” -> “Normal” will start transition after the trigger “BuildingComplete” fires

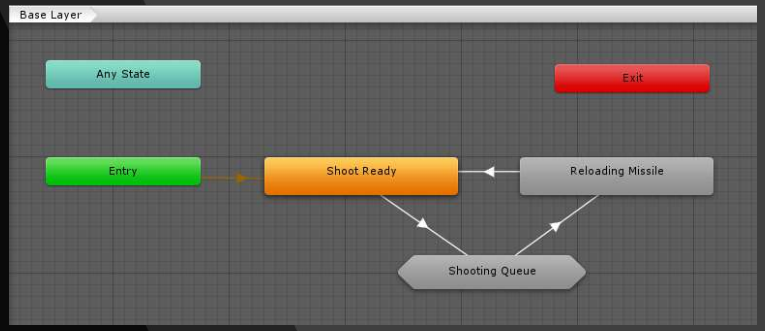


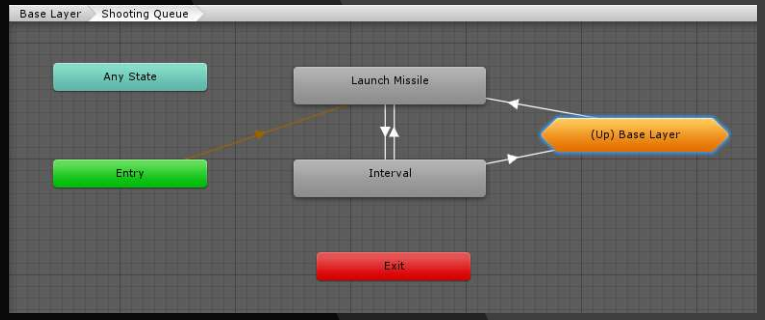




Animation States Control

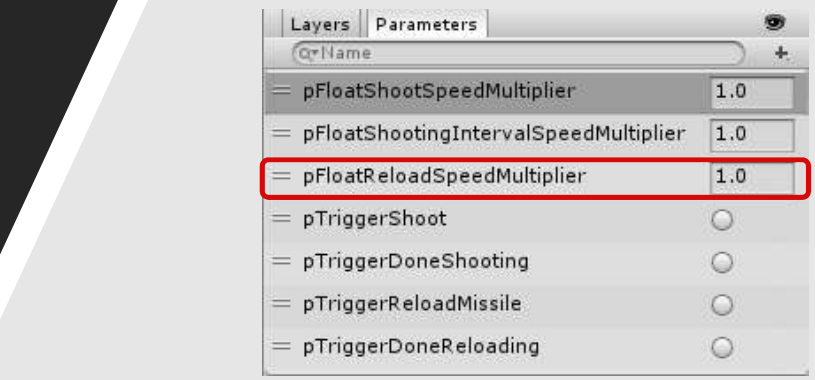
- The shooting of the Rocket Tower & Basic Tower
  - By using the Sub-State Machine, the sequence becomes easier to understand & manage

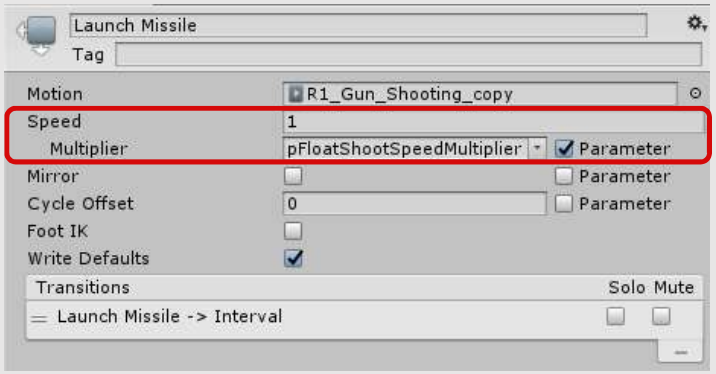


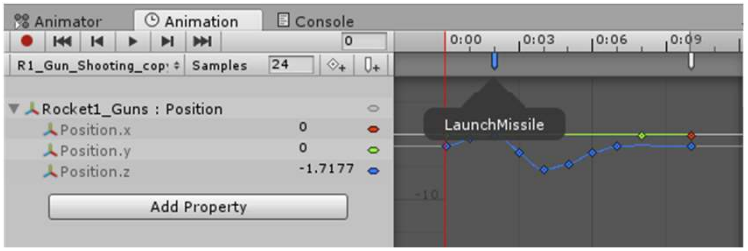


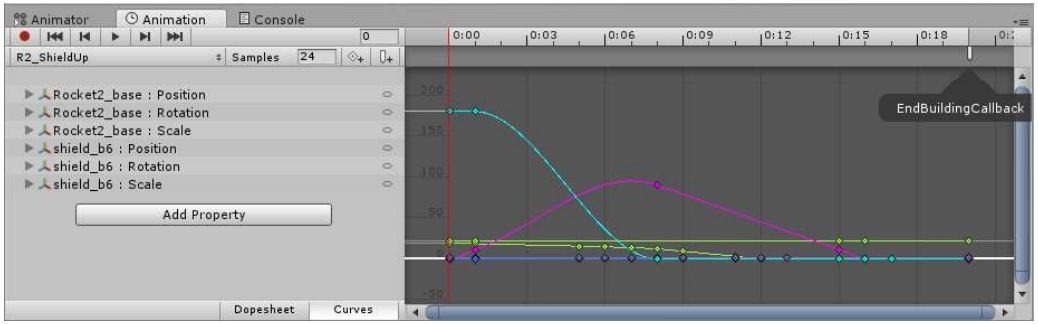
Animation States Control

- Use parameter to control the speed of animation play
- And the parameter can be modified through scripts, so the duration of animation play is dynamically changeable







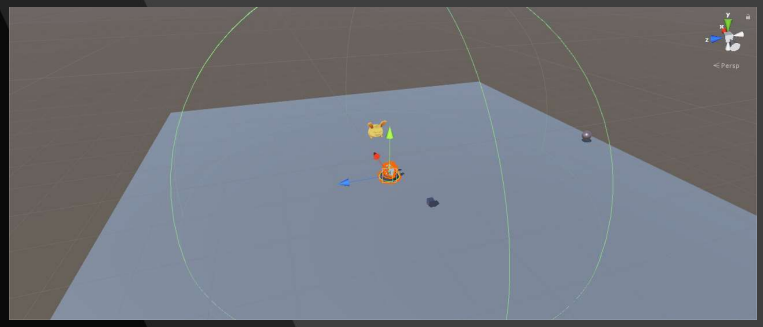


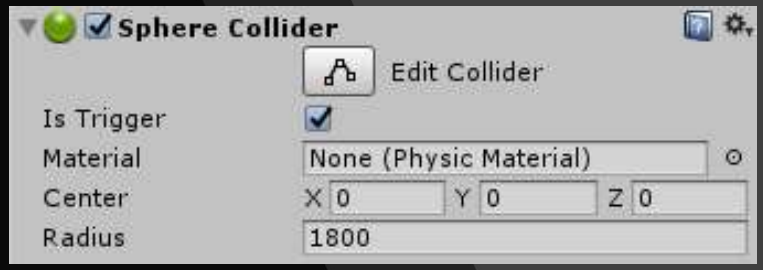
- We add events to the animations to make it happen at the correct time and look reasonable
  - Animation event will call a script function regardless the unstable frame rate, so it's reliable
- And through this method we can implement callbacks of animation play

Animation Event

Enemy Monitor

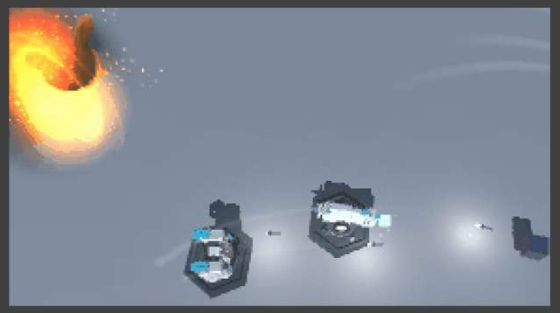
- Each tower has its monitor (attack) range
- We implement this by trigger collider and onTriggerStay() method
- We make it can tell which enemies are the most close to the terminal, and give them the highest priority to be aimed





Rotation to aim target

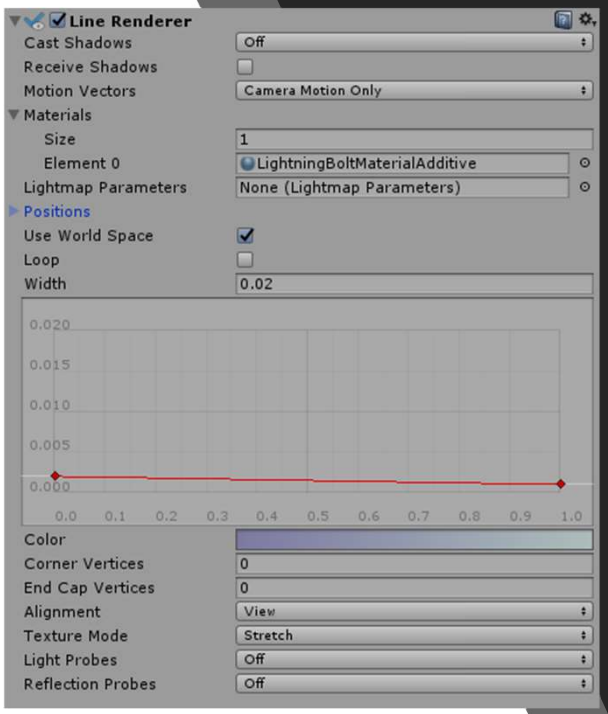
- We implemented smooth rotation for the towers
  - When the tower change its target, we want to make it rotate to the new target smoothly.
    - The angular speed should not mutate
    - The angular speed should have maximum
    - Use angular acceleration to change angular speed
  - We simplify the problem to:
    - If the angular distance is large, at each frame, should the tower accelerate or decelerate, so that it takes the shortest time to aim target and keep the same speed with it?
  - The known conditions:
    - The target's velocity
    - The tower's current angular speed
    - The angular distance between



Special Effect

- We implement many effect to make the graphic attractive
  - Line Renderer
  - Particle System

Use Line Renderer to create lightning



- The Line Renderer of Unity is a powerful component to generate polyline
- We can modify width, material and color for the polyline
- The “Positions” contains dozens of 3D points in x, y, z coordinates
- We use program to randomly and reasonably update the points every several frames to make it behave like a lightning

Use Line Renderer to create lightning

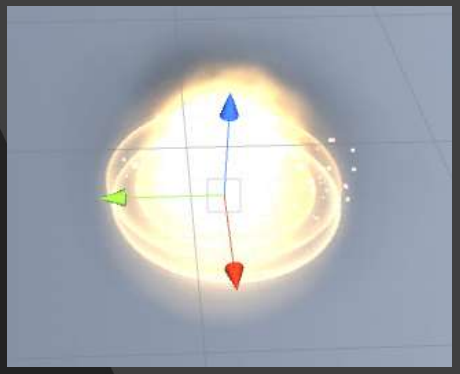
- The start point & end point should be the pole of tower and the target
- The array size is related to the distance
- We make the adjacent points not too apart away from each other to look reasonable

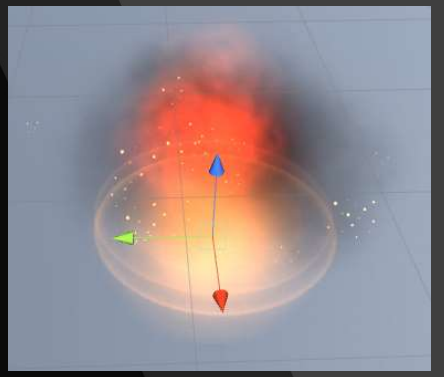
Positions

Size	33					
Element 0	X	3.92	Y	0.02758	Z	9.06
Element 1	X	3.926069	Y	0.0282515	Z	9.058283
Element 2	X	3.931817	Y	0.0294277	Z	9.055875
Element 3	X	3.93814	Y	0.0294077	Z	9.056334
Element 4	X	3.944391	Y	0.0286039	Z	9.057012
Element 5	X	3.949445	Y	0.0287815	Z	9.060855
Element 6	X	3.954752	Y	0.0294298	Z	9.064282
Element 7	X	3.960858	Y	0.0304736	Z	9.065922
Element 8	X	3.967224	Y	0.0311844	Z	9.065784
Element 9	X	3.973329	Y	0.0289183	Z	9.064999
Element 10	X	3.979532	Y	0.0295168	Z	9.06295
Element 11	X	3.985098	Y	0.0310962	Z	9.060115
Element 12	X	3.989482	Y	0.0338401	Z	9.056273
Element 13	X	3.995253	Y	0.0359393	Z	9.05462
Element 14	X	4.001478	Y	0.0359407	Z	9.055923
Element 15	X	4.007473	Y	0.0370071	Z	9.056356
Element 16	X	4.01356	Y	0.0374684	Z	9.056296
Element 17	X	4.019424	Y	0.0368418	Z	9.054017
Element 18	X	4.024163	Y	0.0386045	Z	9.050221
Element 19	X	4.030462	Y	0.0392535	Z	9.049366
Element 20	X	4.035285	Y	0.0401937	Z	9.045281
Element 21	X	4.041015	Y	0.0408311	Z	9.042608
Element 22	X	4.045501	Y	0.0415601	Z	9.038165
Element 23	X	4.050343	Y	0.0380613	Z	9.035537
Element 24	X	4.055157	Y	0.0377762	Z	9.031141
Element 25	X	4.060915	Y	0.0355868	Z	9.029146
Element 26	X	4.065945	Y	0.0315198	Z	9.028864
Element 27	X	4.072272	Y	0.0305634	Z	9.027681
Element 28	X	4.078611	Y	0.0310470	Z	9.029069
Element 29	X	4.084995	Y	0.0317009	Z	9.029093
Element 30	X	4.090883	Y	0.0295848	Z	9.030521
Element 31	X	4.096968	Y	0.0311746	Z	9.029519
Element 32	X	4.103058	Y	0.0300991	Z	9.028

Explosion

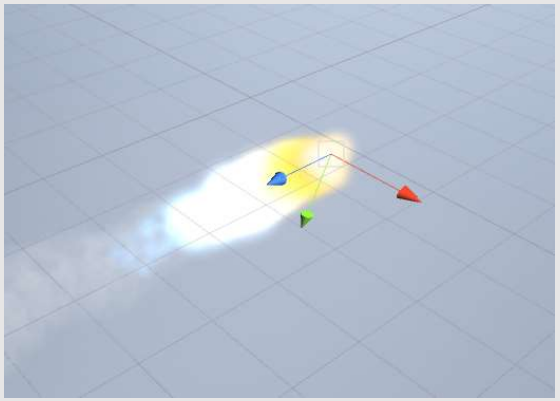
- The explosion is combined with 5 Particle Systems
  - The flame at the center of the explosion
  - The black smoke
  - The glow ball (the heat and the light) at the center of the explosion
  - The rings presenting the blast waves
  - The splashing sparkles
- Their size/color/velocity will change over time

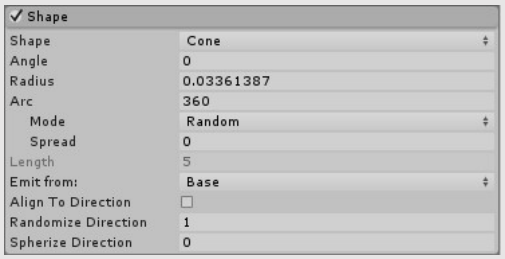


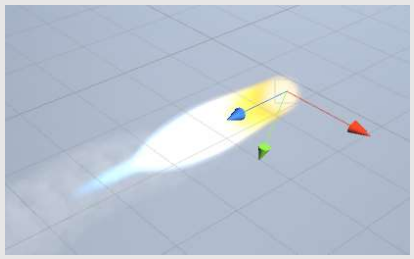


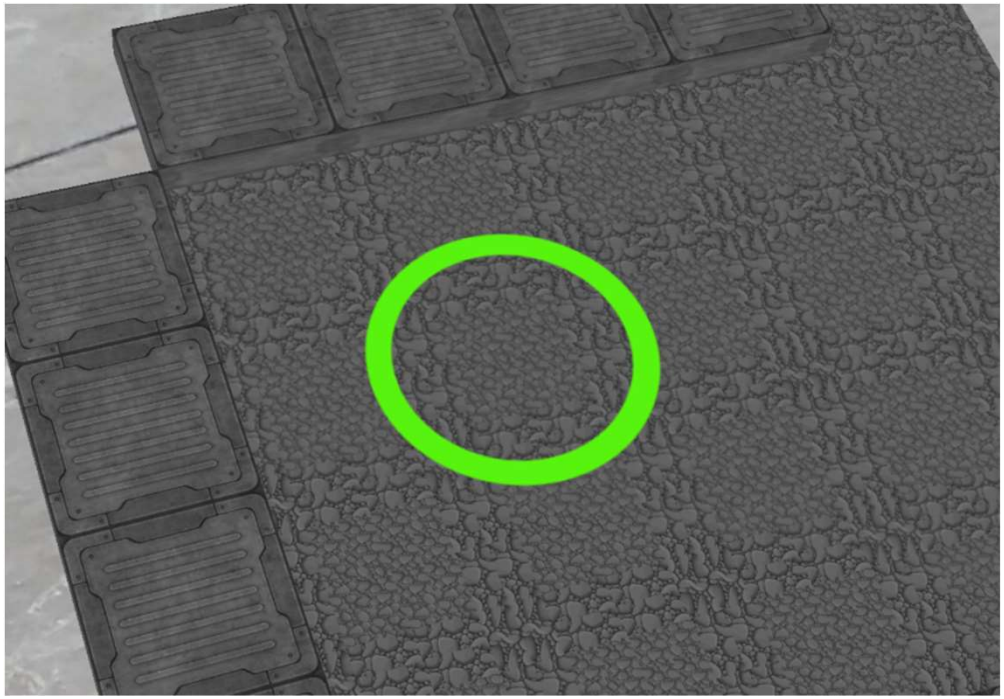
The flame tail of missiles

- This consists of the flame and the smoke
  - Each smoke particle should retain at where it is generated, so the Simulation Space should be “World” to prevent it moves with the missile
  - While the fire’s Simulation Space should be “Local” to follow with the missile
  - To simulate a real flame
    - We use color change over time to make the its yellow-to-white color
    - We use Cone Shape’s Random Arc to simulate the wobbling. Without this the flame will behave like the right







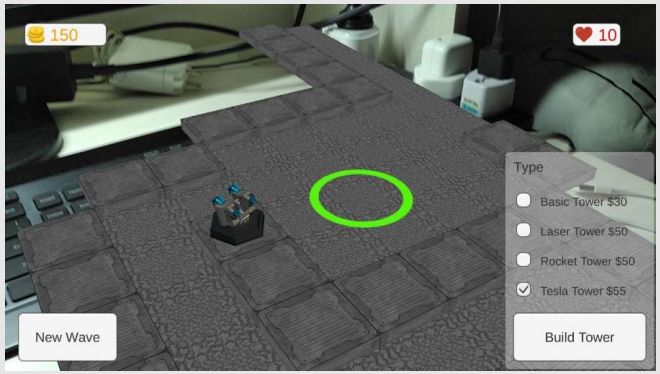


- As we are using Augmented Reality, the virtual objects will be bound with the real world, so we use camera movement to behave like the mouse(focus)
  - We highlight the tile at the center of the screen

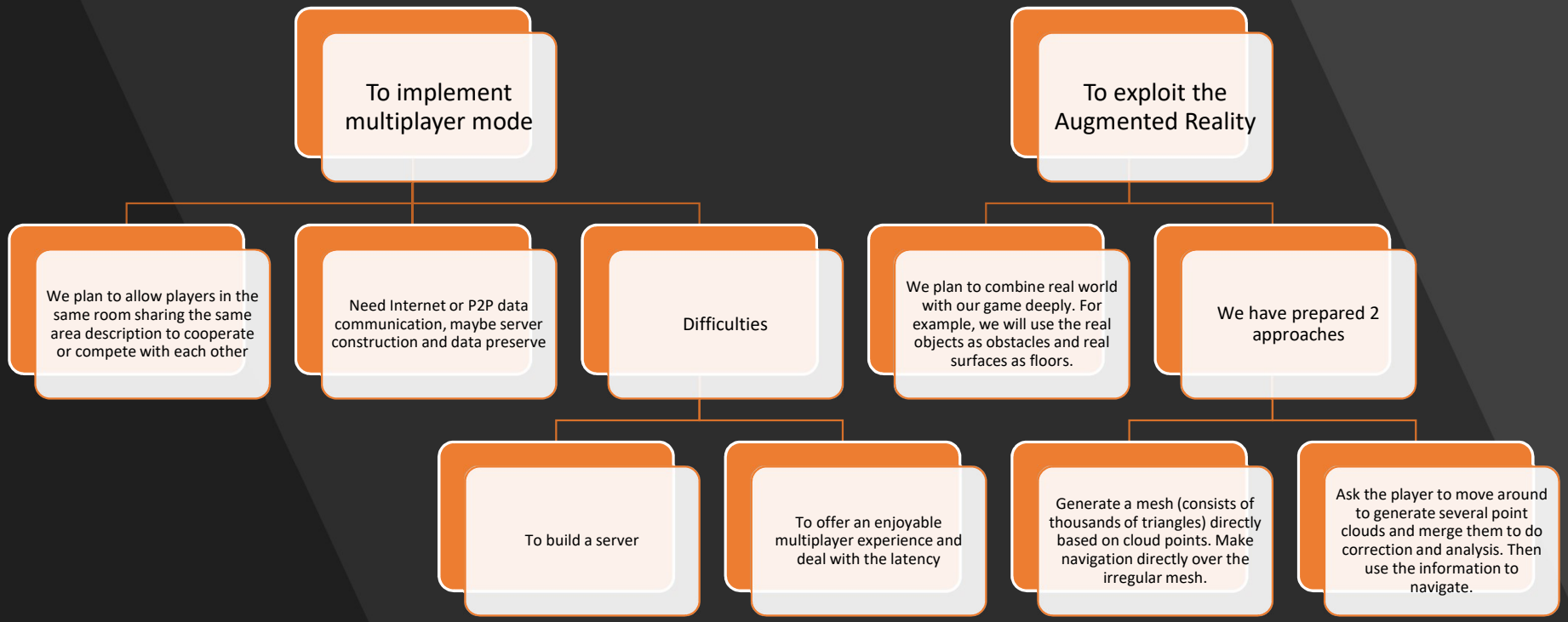
User Interface

User Interface

- The player can click the buttons on the screen to build tower right above the highlighted tile or upgrade the tower selected
- The coin counter
  - It shows player the money remains
- The player health
  - It shows how many enemies can be omitted until player's failure
- New Wave Button
  - By clicking this to start new wave of enemies



Future Plan


To implement multiplayer mode
We plan to allow players in the same room sharing the same area description to cooperate or compete with each other
Need Internet or P2P data communication, maybe server construction and data preserve
Difficulties
To build a server
To offer an enjoyable multiplayer experience and deal with the latency
To exploit the Augmented Reality
We plan to combine real world with our game deeply. For example, we will use the real objects as obstacles and real surfaces as floors.
We have prepared 2 approaches
Generate a mesh (consists of thousands of triangles) directly based on cloud points. Make navigation directly over the irregular mesh.
Ask the player to move around to generate several point clouds and merge them to do correction and analysis. Then use the information to navigate.

Q & A

Thank you!