

Ruiyang Xu (NOPER)

<https://noperproduction.com>

Portfolio



1 / Relief

An Immersive Experience of Cyberchondria



2 / Tracer

A Cinematic Video Game of Love, Duty & Sacrifice



3 / Protest

A Strategic Board Game of Modern Conflict



4 / I Wanted You

A Thesis Proposal on the Evolving Patterns of Global Conflict



5 / Fourth of July

A Virtual Reality Exploration of Post-War Traumatic Disorder

RELIEF

An Immersive Experience of Cyberchondria



(Exhibition | Shanghai | May 2025)

Intro

Relief is a deconstruction and reconstruction of some of my most significant experiences (outside of pain) during cancer treatment process ---Cyberchondria.

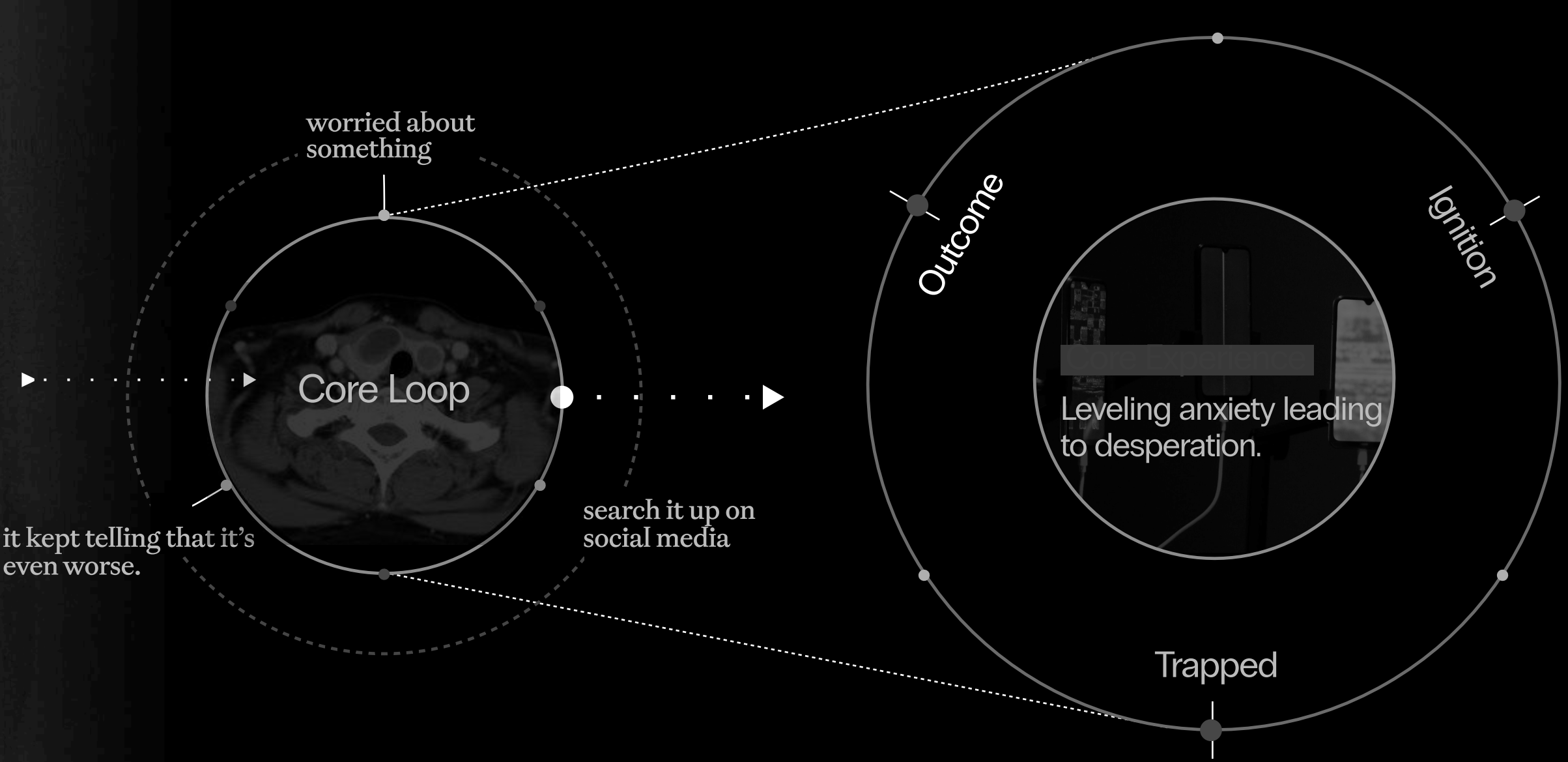
Info:

2025.05 | Individual Project | Davinci Resolve, Figma



Deconstruction Cyberchondria

- Social Media
- Induced
- Anxiety
- Desperation

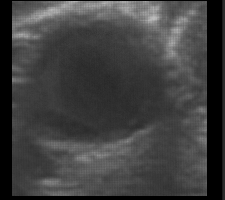


*Ignition

Noticed some
uncomfort (call it X)



→ worried about it



→ Search it up
on social media



→ View posts about diseases with X, with
some diseases (call it Y) more threatening.



What if I got Y?

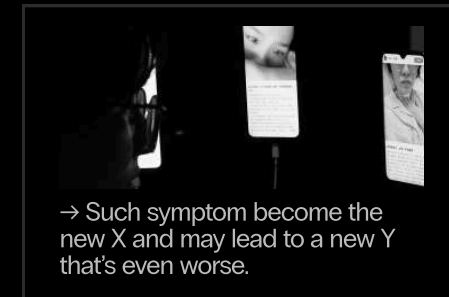
Trapped



→ Search about
symptoms of Y



→ Compare symptoms
to confirm.



→ Such symptom become the
new X and may lead to a new Y
that's even worse.

Outcomes

→ One would be desperate, since they would be convinced
that they had severe diseases, where as they actually not.

/Reconstruction/

Experience to Recreate

Search on social media posts about some symptom, and
refreshing for new posts that hints wrose diseases. During
the process, anxiety gradually morph into desperation

Method

We reconstucts the patient as different components of the room, with
the audience sitting in the middle. We then immerse the audience as
the patient by letting audience do what patient do and feel patient's
emotions real-time.

General Setting:

The patient had a symptom, they searching on socia media,
yet the algorithm is leading them in believing they have cancer.

Sept 2023

2023 Nov

2024 Dec

2024 Aug

2024 Sept

2025 Mar

First Symptoms:

I had sometimes light fever during the night.

Worsening:

I begin to feel pressure around my throat, and the night fever became severe.

Worst Times:

I went to the emergencies multiple times for uncontrollable heart rate and chest pain.

After 8 months of diagnostics, it was finally found to be cancer.

Extrapolate

I was fearing about **death**, **anxiety** became **my routine**.

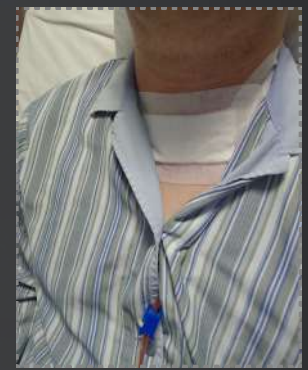
I seeked for relief by searching for people with similar cases to me, on the social platform "XiaoHongShu"

However, it only worsen my **anxieties**, as the platform's algorithm keeps hinting me of worse outcomes

overflown me, I could hardly eat anything



I had my surgery.



The treatment process was nothing like what I was anxious about, it was painful, but successful.

Back on my feet again, I decided to help others that's underwenting similar conditions

Notice that I cannot control how one gets cancer.

However, I can relief one of their anxieties, no matter one being cancer patient or not

Hence, I created this piece to turn restless scrolling into a moment of calm.

/ Deconstruction & Reconstruction / Through of My Experiences.

Immersive Experience Design

Reconstructing Patient's Actions

Reconstructing Patient's Emotions

Social Media:

The most obvious action during the cause is surfing through social media on personal terminals. With the necessary element being the personal terminal, which are mostly cellphones. Therefore, five phones are positioned before the audience, they will display interactable social media content.

Posts:

Having the hardware, we now focus on the software aspect. We wish the audience to immerse into the experience of surfing social media posts on their most familiar platform. Hence, we gathered info on real posts from the social media platform "XiaoHongShu" and recreated the application's interface for a one to one recreation.

Hinting Worse Diseases:

Those five phones will refresh posts periodically. With each refresh, each of five phones will display posts hinting even worse disease than the previous refresh.

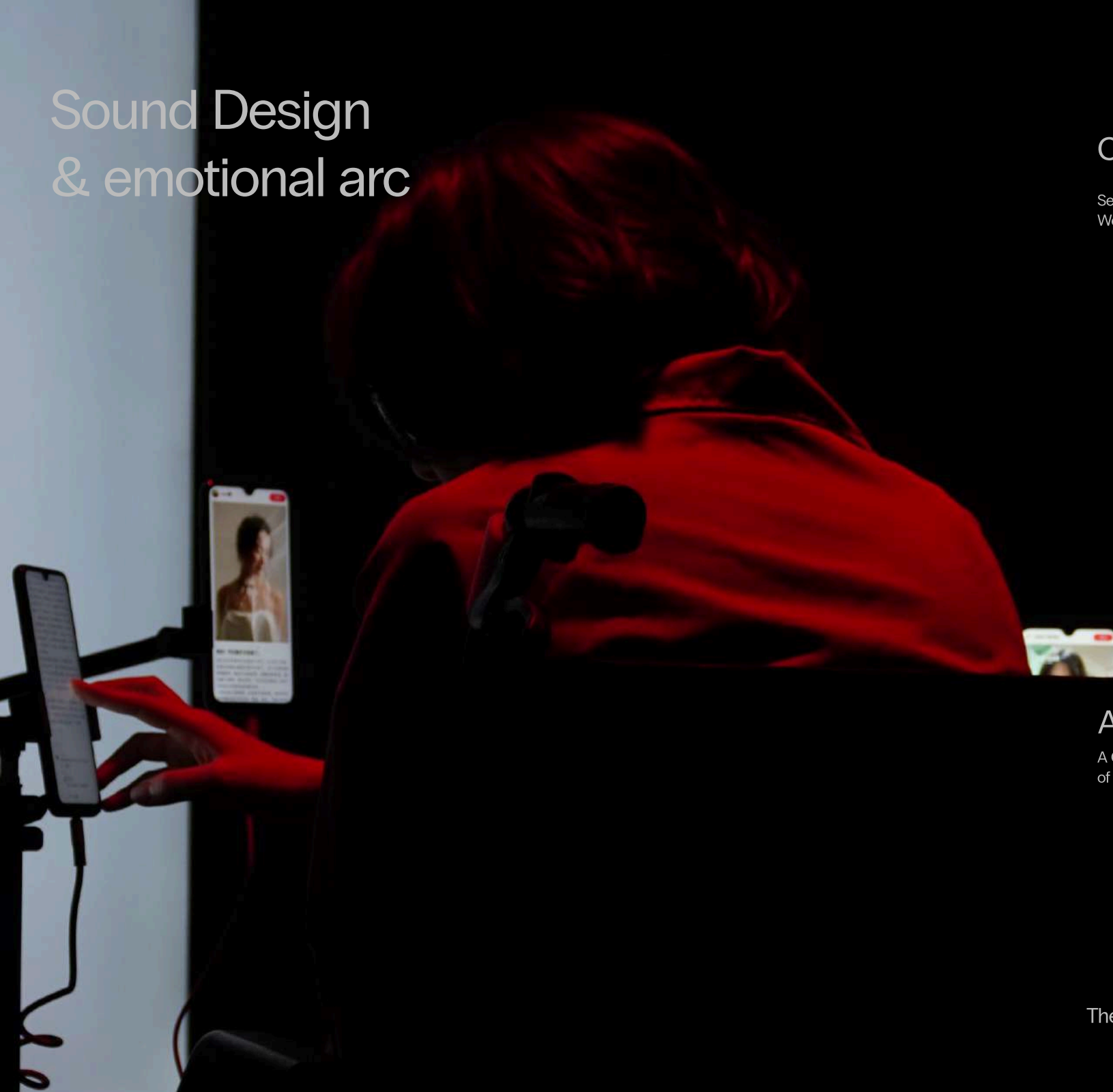
Anxiety:

- * A large screen uprightly positioned in front of the audience displays artificial video content, recreating the patient's abstract silks of thought during the action.
- * A set of duo speakers will play edited audio throughout the process, setting the mood through music.
- * The top red light is for mood setting

Desperation:

With each refresh, each of five phones will display posts hinting even worse disease than the previous refresh.

Sound Design
& emotional arc



CALM

Seat: Wheelchair
Wear: with brace

ANXIETY

A Conceptual film that give the idea
of the theme

The narrative flow of Sound

Stage 1

Stage 2

Stage 3

Stage 4

Stage 5

Stage 6

Uncomfortable & Confused

Half-believing, searching
the web for answers.

Torn between
certainty and
denial.

Then, utterly convinced.

Desperate

Hopeful

Acceptance

A glimpse of false hope

The Approaching Death

Suffer

illness

Adult

Teenager

Child

Infant



Digital Noise surrounds the audience, enter
themselves into a hyperscaped world



Noise turns into repetitive signals, doubts arise

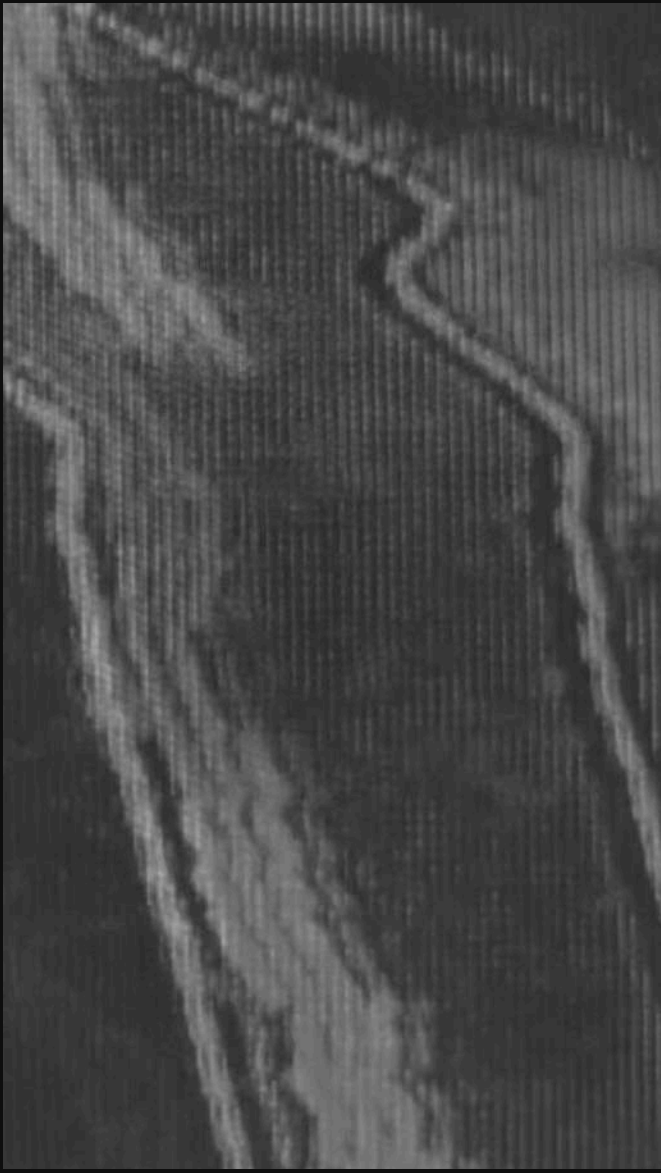
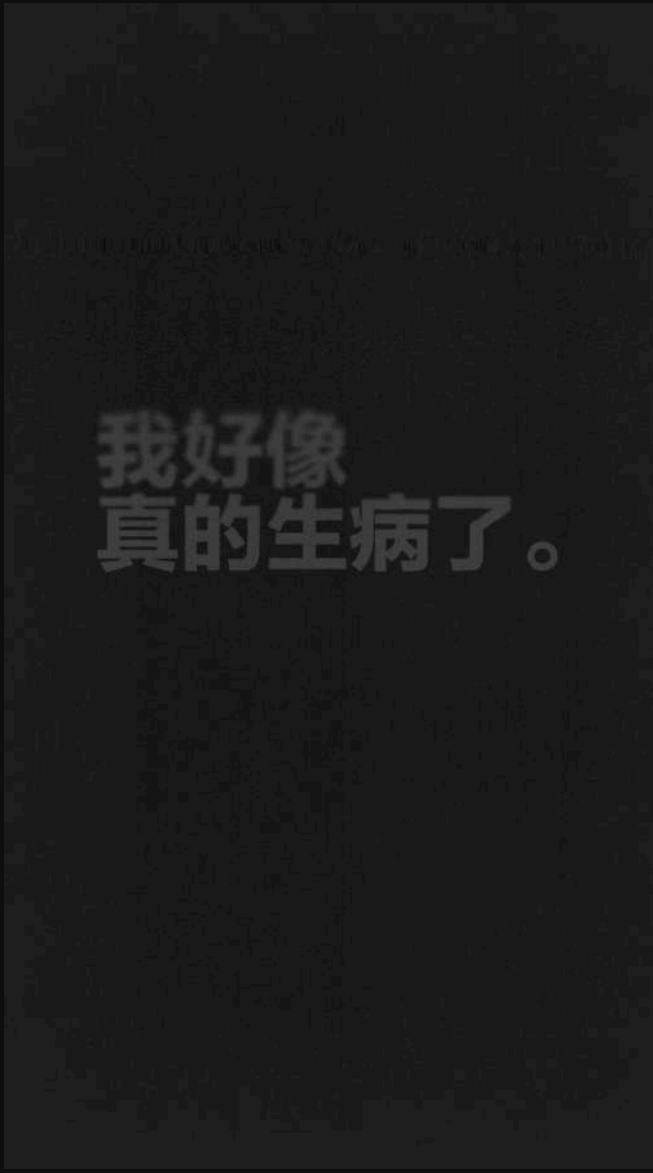
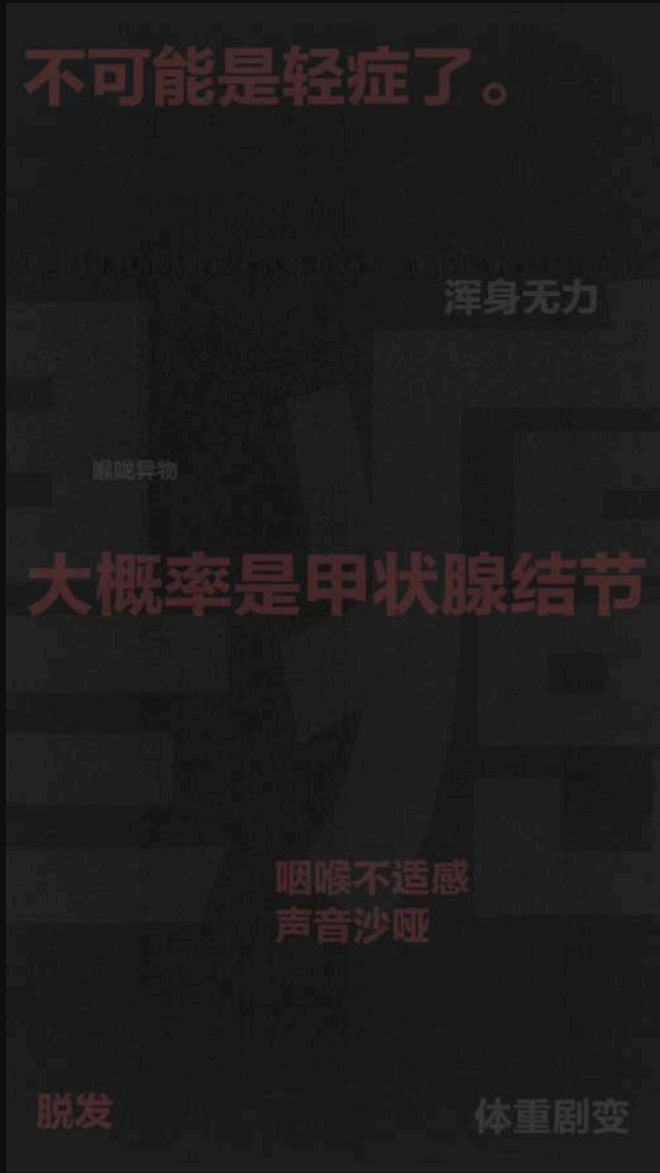
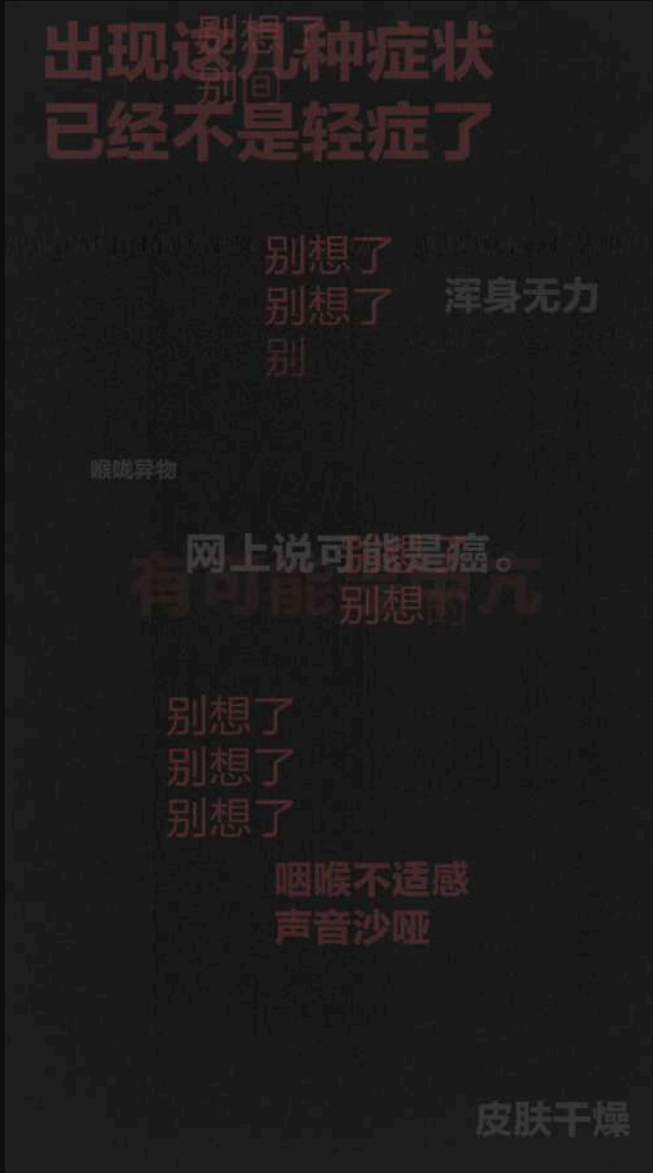
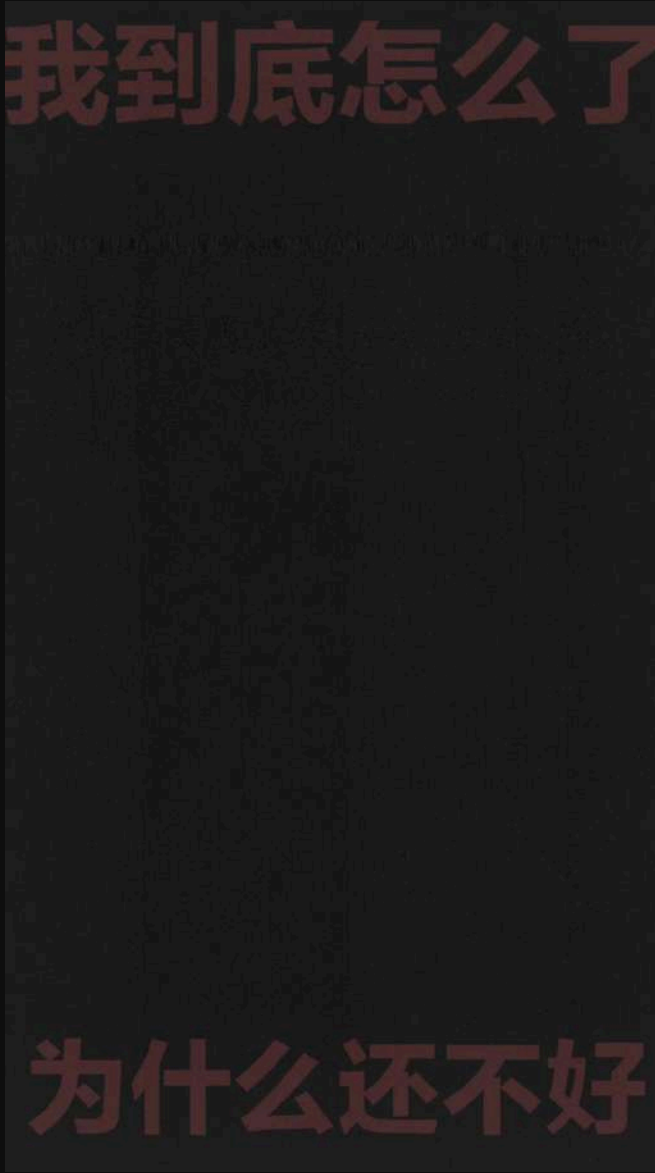
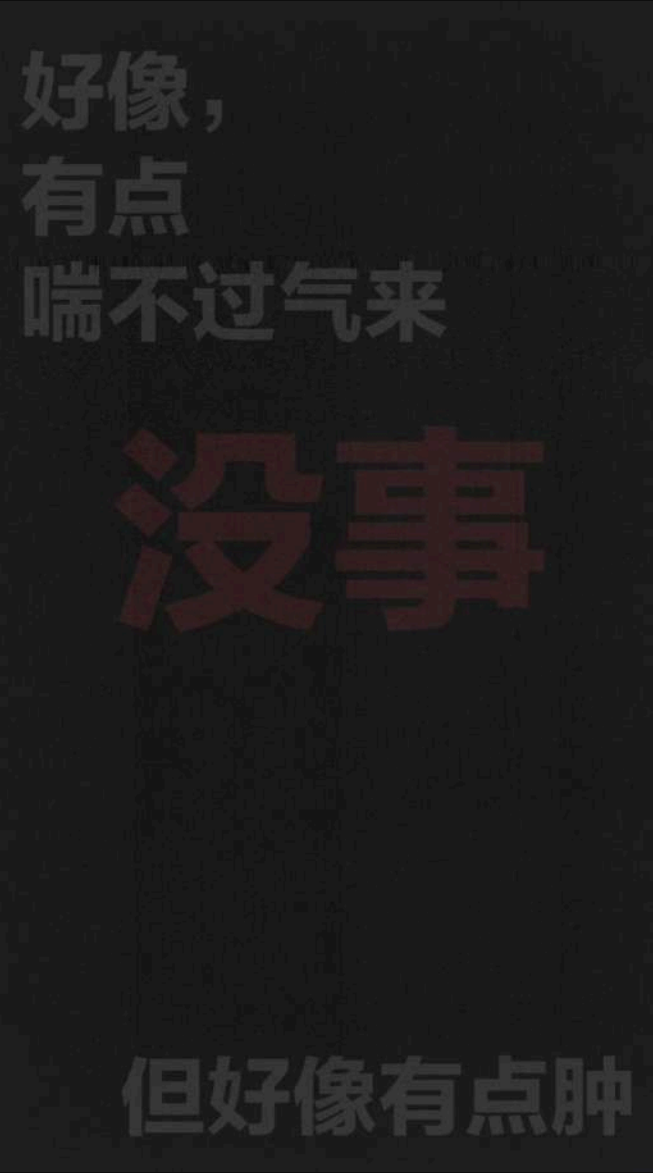
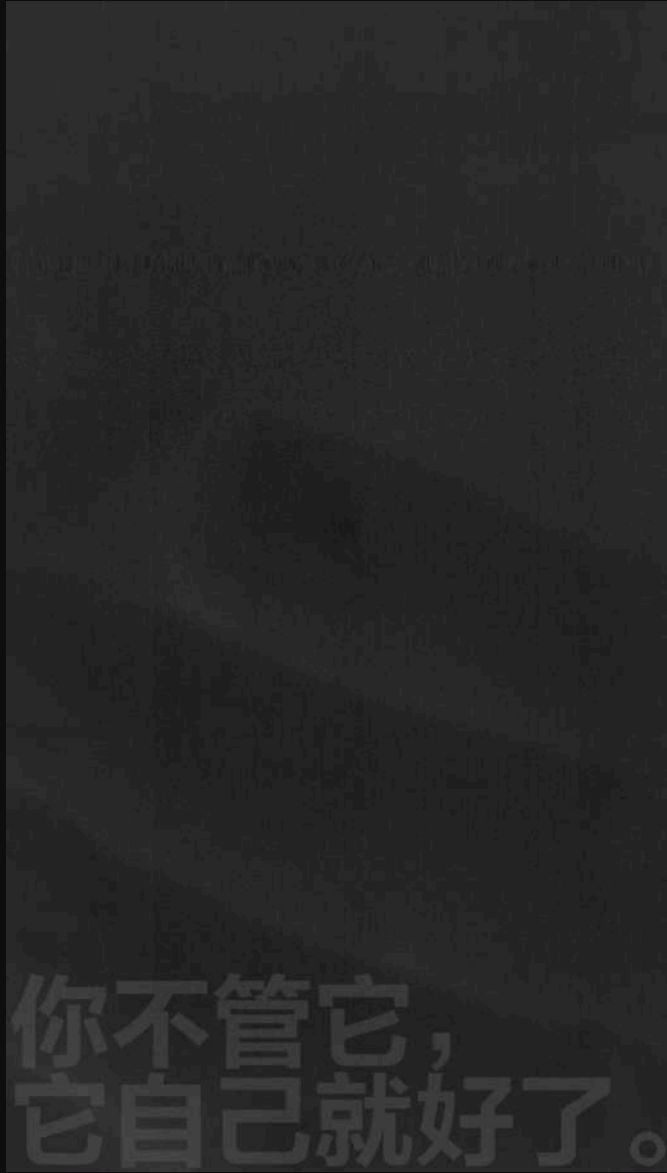


Repeat between E and F bass chords, potential
crisis emerges



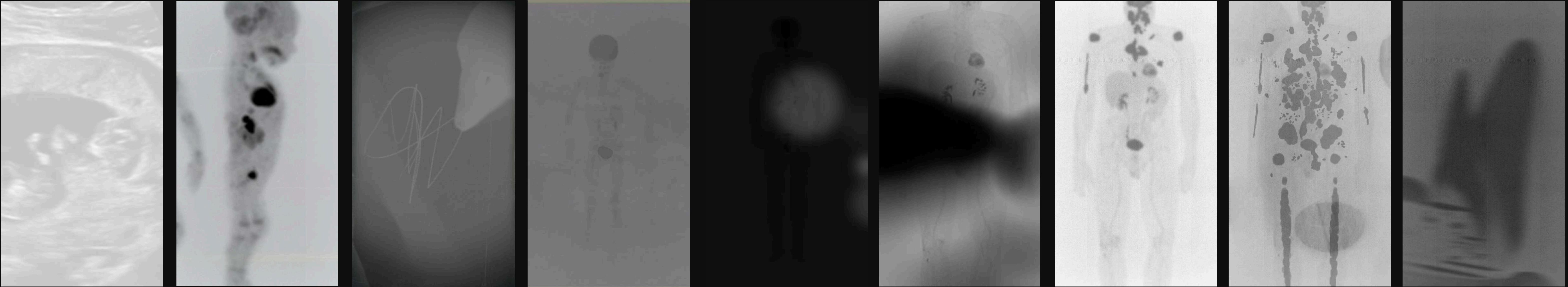
Nothing but ear ringings, interrupted by bursts of
haunting thoughts
Emotional Numbness





Filmmaking

The chinese phrases are the patient's own thoughts when trapped by social media. for example: “我好像真的生病了” stands for “It looks like I really got the disease”



Filmmaking

In stage 6, while the patient is imagining their failing treatment and having a flashback of their life, a PET-CT scan will illustrate the progression of cancer.

Programming

*System Flow Chart

Android App: Build a native Android container to load a web application.

Server: A WebSocket-based device status synchronization system built with Node.js.

Client: Built with Vue 3 to support multi-device collaborative interaction.

Full-Screen Immersive Experience

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowManager.LayoutParams.FLAG_FULLSCREEN);
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    webView = findViewById(R.id.webview);
    WebSettings webSettings = webView.getSettings();
    webSettings.setJavaScriptEnabled(true);
    webSettings.setPluginState(WebSettings.PluginState.ON);
    webSettings.setAllowFileAccess(true);
    webSettings.setAllowFileAccessFromFileURLs(true);
    webSettings.setSupportZoom(true);
    webSettings.setLowBandwidthViewMode(true);
    webSettings.setCacheMode(WebSettings.LOAD_NO_CACHE);
    webSettings.setDomStorageEnabled(true);
    webSettings.setMediaPlaybackRequiresUserGesture(false);
}
```

Advanced WebView Configuration

```
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
    webSettings.setMixedContentMode(WebSettings.MIXED_CONTENT_ALWAYS_ALLOW);
}

// 允许内容加载时显示地址栏
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.P) {
    WindowManager.LayoutParams lp = getWindow().getAttributes();
    lp.layoutInDisplayCutoutMode = WindowManager.LayoutParams.LAYOUT_IN_DISPLAY_CUTOUT_MODE_SHORT_EDGES;
    getWindow().setAttributes(lp);
}

getWindow().getDecorView().setSystemUiVisibility(
    View.SYSTEM_UI_FLAG_LAYOUT_FULLSCREEN | View.SYSTEM_UI_FLAG_LAYOUT_STABLE
);

webView.setWebViewClient(new WebViewClient());
webView.loadUrl(url);

String url = "http://120.24.227.118:8081/?t=" + System.currentTimeMillis();
webView.loadUrl(url);
});
```

Full-Screen Immersive Experience

```
let deviceList = [
  { id: "0", status: "0", yon: "0" },
  { id: "1", status: "0", yon: "0" },
  { id: "2", status: "0", yon: "0" },
  { id: "3", status: "0", yon: "0" },
  { id: "4", status: "0", yon: "0" },
  { id: "5", status: "0", yon: "0" },
];

const deviceConnections = new Map();
const wss = new WebSocket.Server({ port: WS_PORT });
wss.on("connection", (ws) => {
  ws.on("message", (message) => {
    const msgList = JSON.parse(message.toString("utf-8"));
    msgList.forEach((msg) => {
      deviceConnections.set(ws, msg.id);
    });
  });

  ws.on("message", (message) => {
    const msgList = JSON.parse(message.toString("utf-8"));
    deviceList = deviceList.map((item) => {
      msgList.forEach((msg) => {
        Object.assign(item, msg);
      });
    });
    return item;
  });
});

wss.clients.forEach((client) => {
  if (client.readyState === WebSocket.OPEN) {
    client.send(JSON.stringify(deviceList));
  }
});
});
```

Device Status Synchronization Mechanism

```
ws.on("close", () => {
  const currentDevice = deviceConnections.get(ws);

  var stamp = new Date().getTime() + 8 * 60 * 60 * 1000;
  var beijingTime = new Date(stamp)
    .toISOString()
    .replace(/T/, " ")
    .replace(/\.\d+/, "")
    .substring(0, 19);

  console.log("有设备断线了!! ", currentDevice, "时间:", beijingTime);

  if (currentDevice !== undefined) {
    deviceList = deviceList.map((item) => {
      if (currentDevice === item.id) {
        item.status = "0";
        item.yon = "0";
      }
    });
    return item;
  }

  you, 上个月 * first commit

  wss.clients.forEach((client) => {
    if (client.readyState === WebSocket.OPEN) {
      client.send(JSON.stringify(deviceList));
    }
  });
});
```

Distributed Communication Architecture

```
ws.onmessage = (event) => {
  try {
    const data = JSON.parse(event.data);
    if (data[0].status === "2") {
      if (isLock) {
        startCountDown();
        isLock = true;
      }
    }
    const hasho = data.some((item) => item.yon === "n");
    if (hasho) {
      isSelectNo.value = true;
    }
    if (data[1].status === "200") {
      blackFullMask.value = true;
    } else if (data[1].status === "202") {
      countdownTime.value = s_t-1;
      pageons.value = d300Sidist[0];
      blackFullMask.value = false;
      playWMS.value = false;
      timerId = null;
      isLock = false;
      isSelectNo.value = false;
      showMask.value = true;
      videoRef.value.currentTime = 0;
      videoRef.value.pause();
    }
  } catch (error) {
    console.error("数据解析失败: ", error);
  }
};
```

Dynamic Content Delivery

```
//设备主内容内容
const d300Sidist = ["1", "2", "3", "10", "15", "21"];
const d200Sidist = ["2", "6", "11", "16"];
const d300Sidist = ["4", "7", "12", "17", "20", "22"];
const d400Sidist = ["8", "13"];
const d500Sidist = ["9", "14", "18", "19", "23", "24"];

//响应式内容标识
const pageWMS = ref(d300Sidist[0]);

//动态组件
<XHSComponent
  :showId="item.id"
  :authing="item.authing"
  :authName="item.authName"
  :sliders="item.sliders"
  :noteTitle="item.noteTitle"
  :noteDesc="item.noteDesc"
  :noteDate="item.noteDate"
  :isSelectNo="item.isSelectNo"
  :commentCount="item.commentCount"
  @ohs-select-no="handleXhsemit"
  :isSelectNo="isSelectNo"
  :currentDevice="currentDevice"
/>
```

Full-Screen Image Modal Interaction System

```
<div class="auth-name">{{ comment.commentName }}</div>
<div class="auth-content">{{ comment.commentContent }}</div>
<div>
  <div class="auth-commenting"
    @click="showMask(comment.commenting)"
  >
  <img:src="comment.commenting" alt="" />
</div>
<div class="auth-date">{{ comment.commentDate }}</div>
```

Windsurf: Refactor | Explain | Generate Function Comment | X

```
function showMask(img) {
  const mask = document.createElement("div");
  mask.classList.add("mask-img");
  mask.innerHTML = ``;
  document.body.appendChild(mask);
  mask.addEventListener("click", () => {
    document.body.removeChild(mask);
  });
}
```

Bistable Like State Management Engine

```
<div>
  v-if="comment.commentLikeing === 0"
  @click="comment.commentLikeing = 1"
  class="like"
  >
    <div class="like-img">
      
    </div>
    <div>{{ comment.commentLike }}</div>
  </div>
  v-else @click="comment.commentLikeing = 0" class="like"
  <div class="like-img">
    
  </div>
  <div>{{ -comment.commentLike + 1 }}</div>
</div>
```

```
const levelComments = Array.from(
  { length: list.value.length },
  (_, i) => photoool[i % photoool.length]
);

const levelComments = Array.from(
  { length: props.commentCount - list.value.length },
  (_, i) => photoool[(1 + list.value.length) % photoool.length]
);
```

Timeline Synchronization Engine

```
function startAddingMsg() {
  setTimeout(() => {
    sendMsg(1);
  }, 4.1 * 1000);

  setTimeout(() => {
    sendMsg(2);
  }, 4.2 * 1000);

  setTimeout(() => {
    sendMsg(3);
  }, 4.3 * 1000);

  setTimeout(() => {
    sendMsg(4);
  }, 4.4 * 1000);

  setTimeout(() => {
    sendMsg(5);
  }, 4.5 * 1000);

  setTimeout(() => {
    sendMsg(6);
  }, 4.6 * 1000);

  setTimeout(() => {
    sendMsg(7);
  }, 4.7 * 1000);

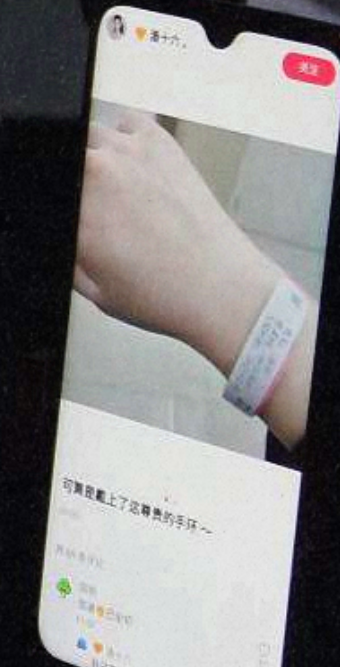
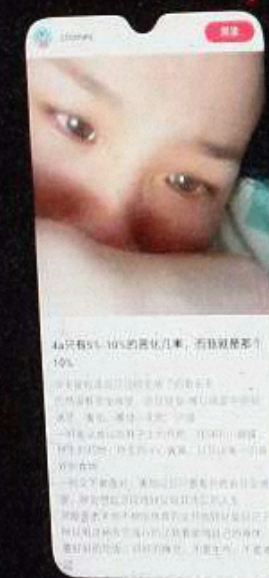
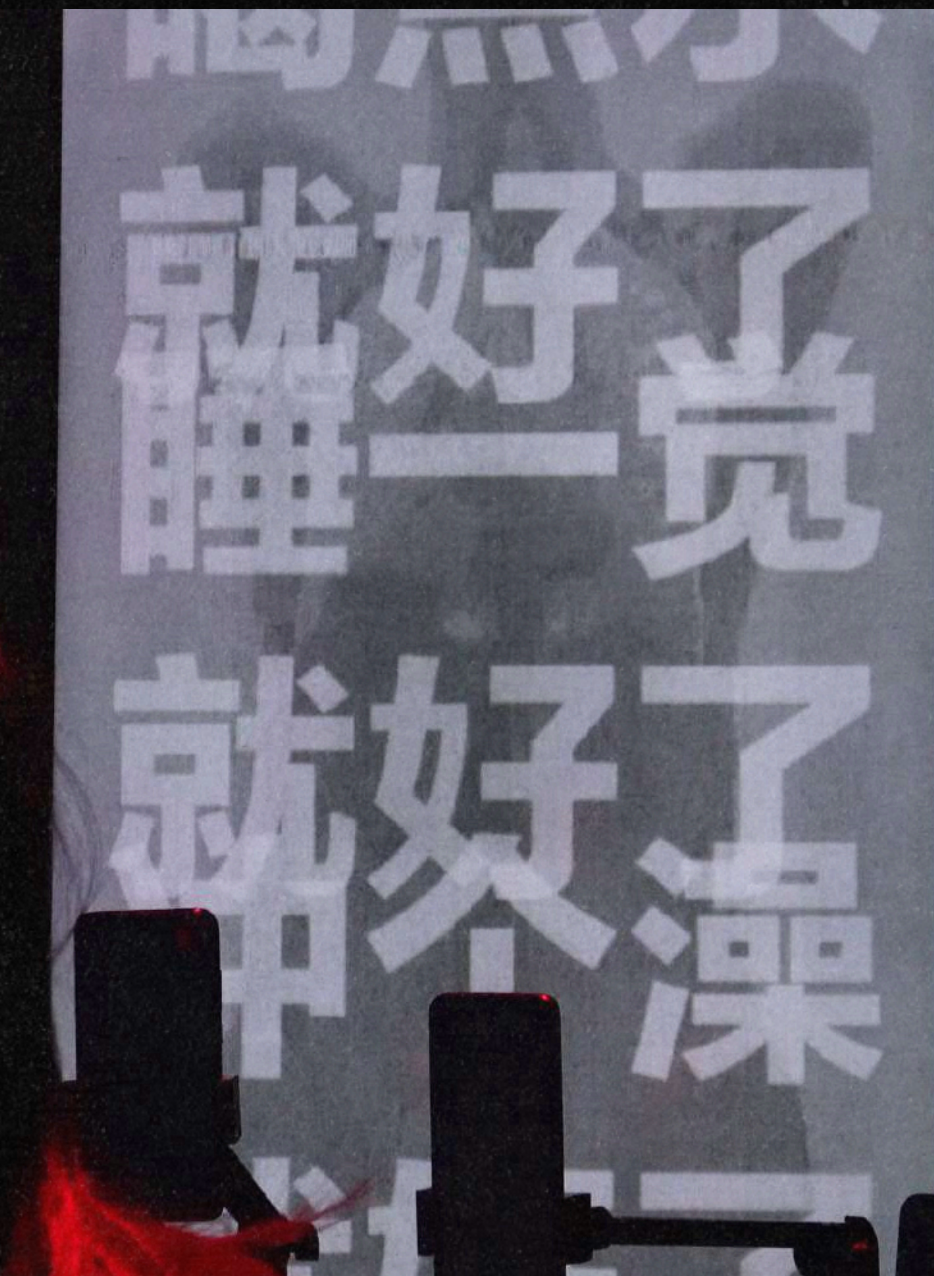
  setTimeout(() => {
    sendMsg(8);
  }, 4.8 * 1000);

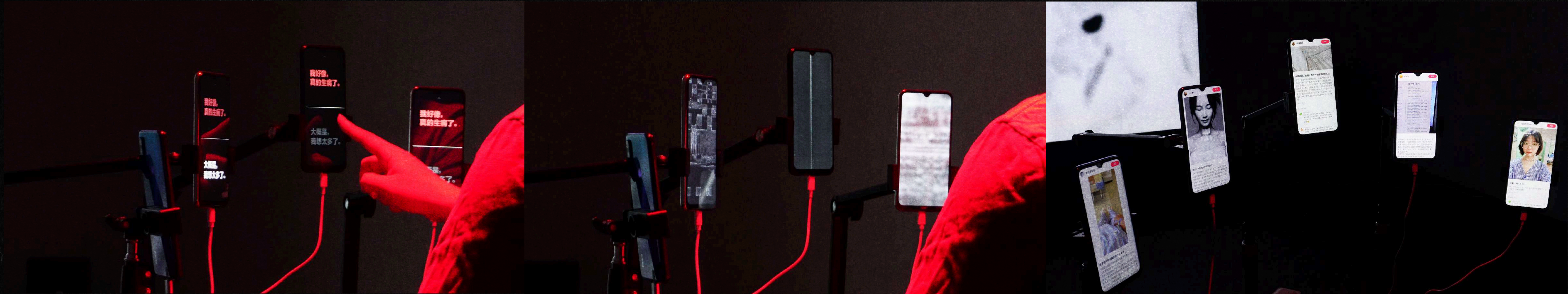
  setTimeout(() => {
    sendMsg(9);
  }, 4.9 * 1000);
}
```

link: video for the full flow

*Final Display

Seeking solace from the algorithm,
we crave comfort in self-diagnosis,
only to slip unknowingly into an escalating cycle of self-intimidation.





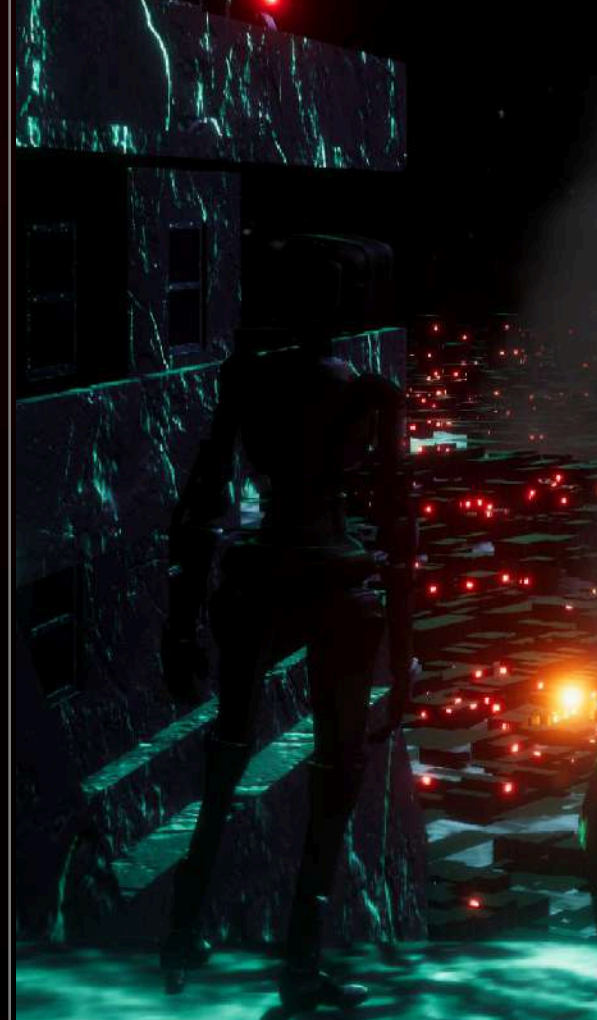
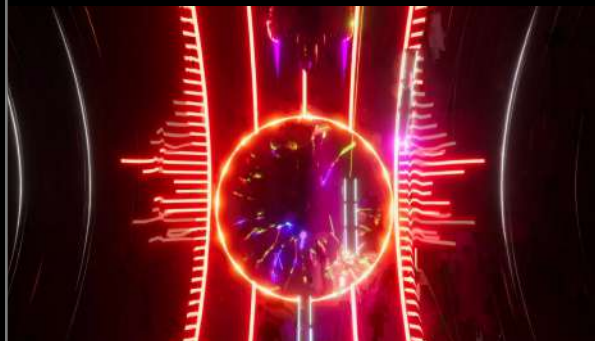
A self-fulfilling prophecy unfolds:
— each click reshapes our cognitive landscape,
as vague discomfort mutates into vivid visions of impending doom.



The end.



TRACER



No Methods Home Coming Exists,
Before Getting Burned To Ashes During Atmospheric Reentry,
Alan Finished Himself With A Pistol.

Intro:

"Tracer" seamlessly blends cinematic storytelling with game mechanics to deliver a deeply immersive and emotionally resonant experience. It follows the tragic yet heroic journey of a warrior father who willingly embraces death to safeguard peace.

Unfolding from the perspective of his wife and child, the story places players in the role of the warrior, where they must navigate intense emotional choices and unwavering convictions.

Project Info:

2025.05 | Individual Project | Unity, Davinci Resolve, Figma

Video Game

Preface

On February 25th, 2024, Aaron Bushnell — a 25-year-old U.S. Air Force cyber-defense specialist — walked alone to the gates of the Israeli Embassy in Washington, D.C.

He wore his uniform.
He carried no weapon.
He had made his choice.
In a world where most convictions are whispered into the safety of screens, Aaron chose the most irreversible stage imaginable.

He pressed a livestream button, looked into the lens, and said:
"I will no longer be complicit in genocide."
Then he set himself alight, shouting "Free Palestine!" as the flames consumed him.

It was not a bid for survival — it was a final stand, endured until his body could endure no more.
And yet, when it was over, the world had not changed.
The cause that drove him remained unanswered. The powers he opposed moved on, unmoved.
But the image stayed — seared into the conscience of those who saw it.

**IN MEMORY OF Aaron Bushnell
1999-2024**

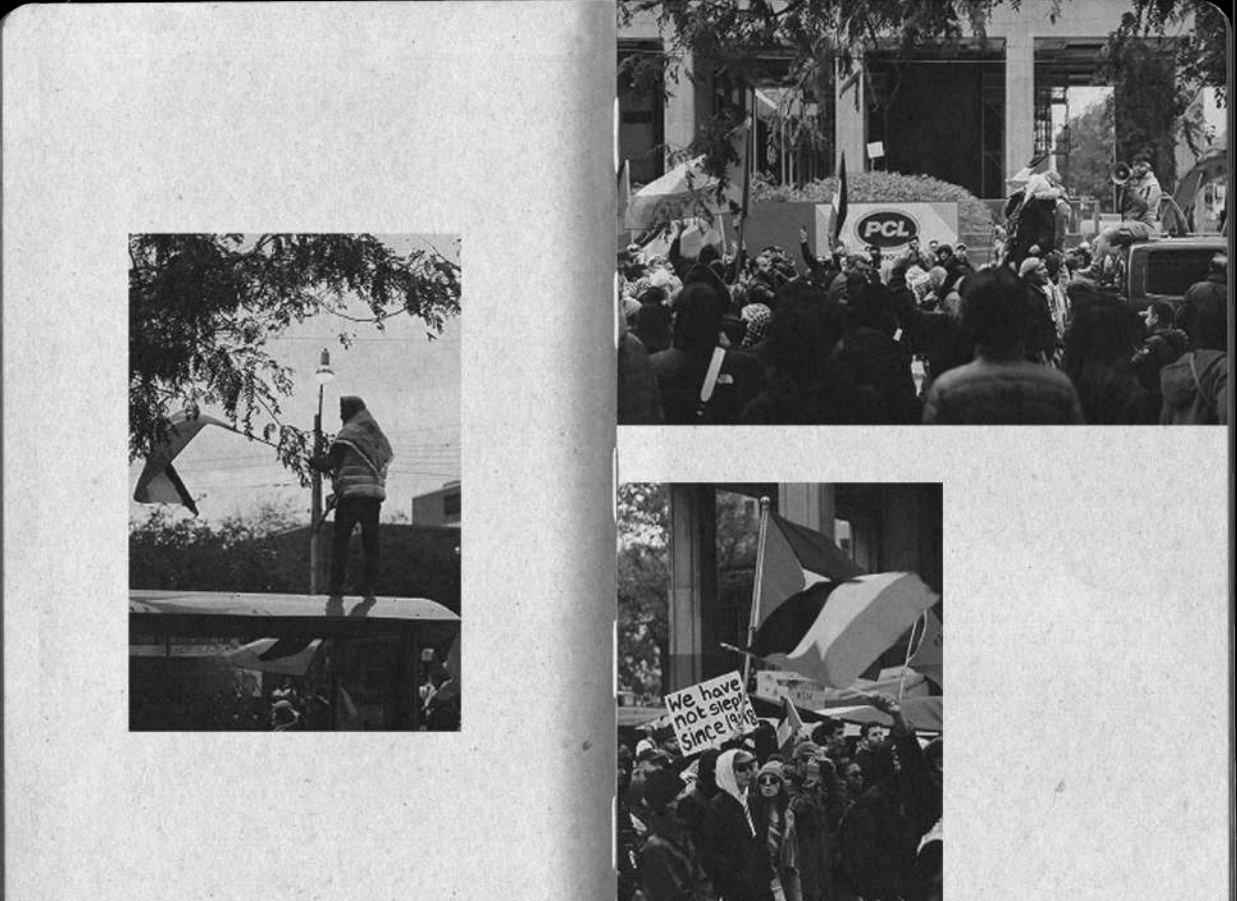
I couldn't shake it.
The image of someone walking willingly into an ordeal they could not survive — not out of desperation, but out of conviction — kept replaying in my mind.

It was the knowledge that he faced unbearable pain for a cause he believed was worth more than his own life... and that, in the end, the world he tried to change remained the same. That gap between courage and outcome is what lodged in my chest and refused to leave.

This matters to me because I was in Toronto, photographing Gaza protests.

I saw grief etched into faces, heard voices worn thin from shouting, and witnessed images of suffering so raw they stayed with me.

In those moments, I understood why someone might give everything to try to end it.



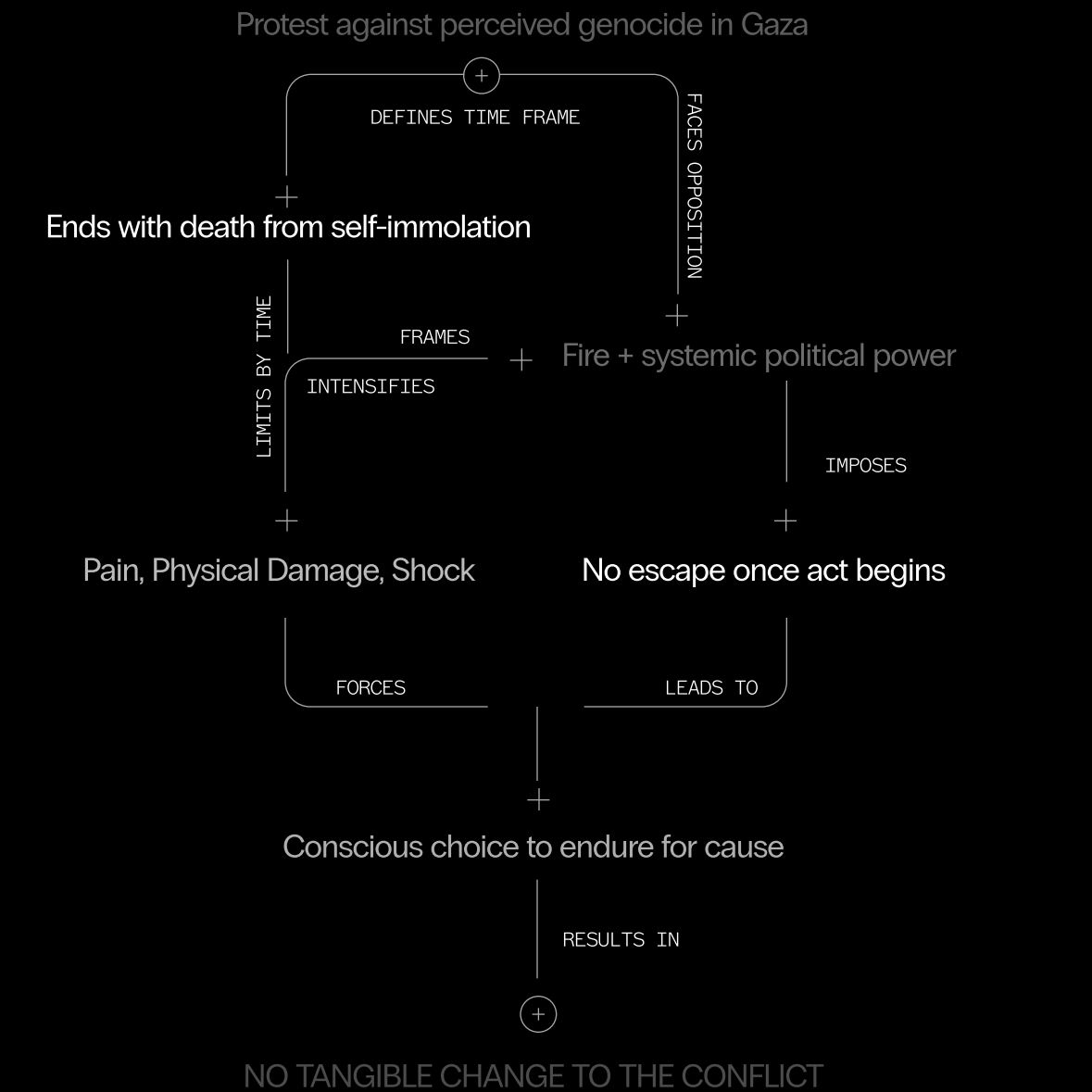
In memory of Aaron Bushnell—and in sorrow that such a profound sacrifice seemed to change so little—I began TRACER. Because Aaron served in the Air Force, the story is told through jets: an interceptor holding the line.

In TRACER, that uniform is given agency; endurance doesn't vanish into silence—it reshapes the world.



Deconstruction

Aaron Bushnell — A U.S. Air Force serviceman who set himself on fire outside the Israeli Embassy in protest of the war in Gaza, enduring until death for a cause he believed in.



Before TRACER could be built, I needed to understand why this kind of story hits so hard.

What makes certain moments of sacrifice unforgettable — in life, in history, in fiction? To answer that, I had to strip these moments down to their bones and study the structure underneath.

*Cause

The driving reason that makes the risk worth taking.

*Finite Survival Window

The set span of time the ordeal must be endured.

*Asymmetric Power

The overwhelming force arrayed against the protagonist.

*Escalation Curve

The final result, whether victory or failure.

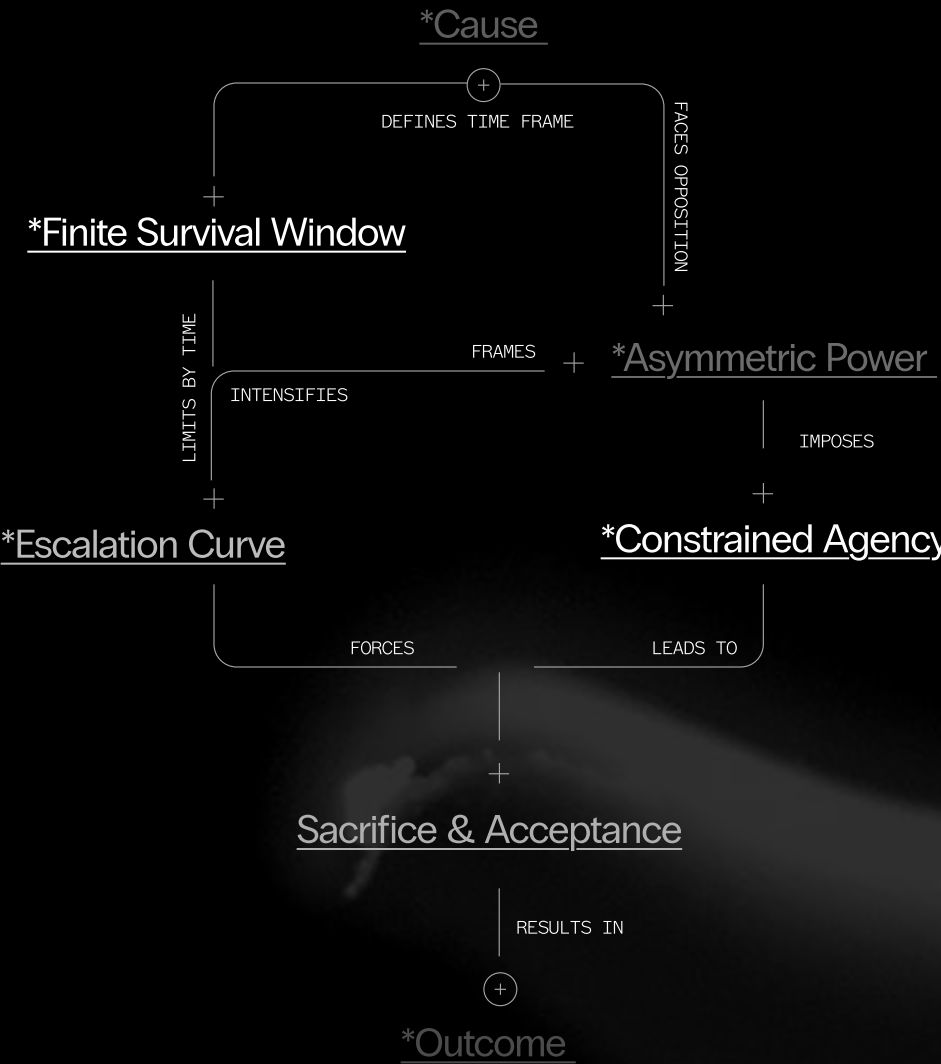
*Constrained Agency

The limitations on what the protagonist can do.

Sacrifice & Acceptance

The embrace of personal loss for the greater goal.

Identify and abstract
the key elements:



*Outcome

The final result, whether victory or failure.

Reconstruction

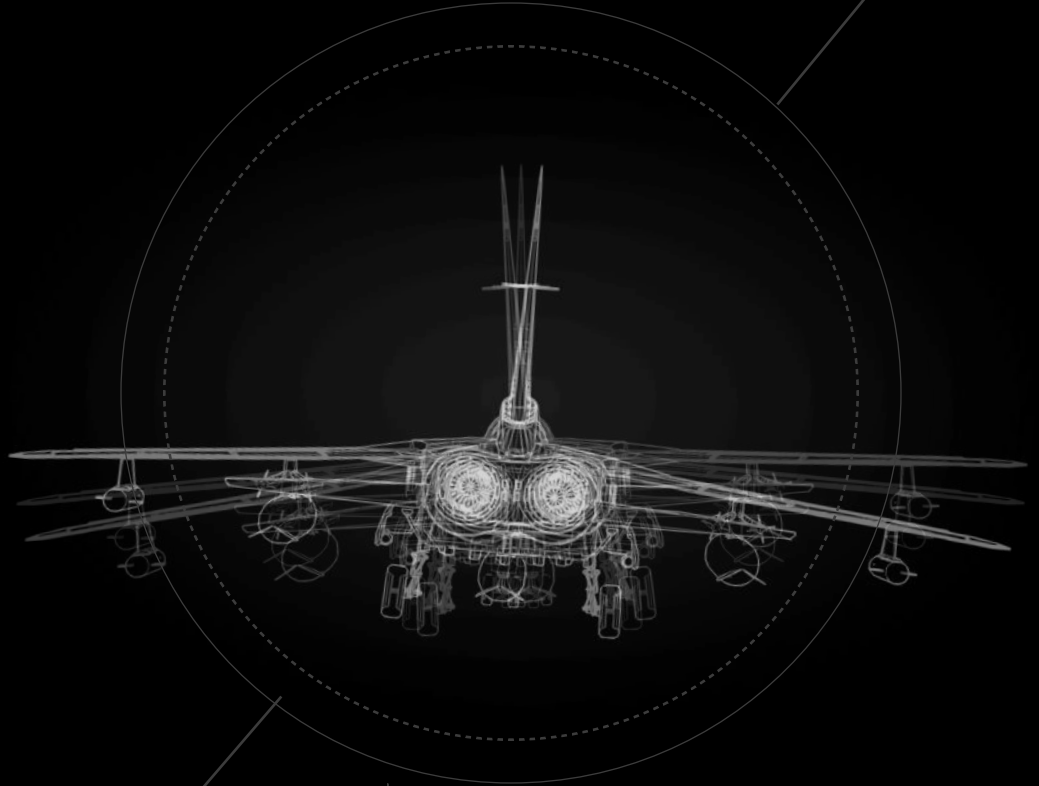
THROUGH CINEMATIC GAMEPLAY DESIGN



For immersiveness, Each element can be reflected either mechanically or narratively, with some elements intersecting both

Constrained Agency

Movement
The jet moves only left and right within a fixed horizontal lane at the bottom of the screen, making precise positioning the key to survival.



Aiming & Shooting

Guns fire straight forward; aiming is done entirely by shifting left or right to align with threats, combining attack and evasion into the same action.

Asymmetric Powers

Challenges

Obstacle Spawning

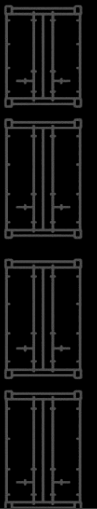
Five fixed points at the top of the screen release obstacles at random intervals, ensuring no two runs feel exactly the same.

Obstacle Behavior

Obstacles drop straight down and deal heavy damage on impact, forcing quick positioning or accurate fire to survive.

Player Interaction

Shooting obstacles destroys them before they reach the jet, but each missed shot risks narrowing safe space.



Escalation

Spawn rate and fall speed steadily increase over time, turning manageable patterns into chaotic barrages.

Boss attacks combine with falling obstacles, forcing players to dodge layered threats while the clock runs down.



Boss

Stage 1 – Top Boss

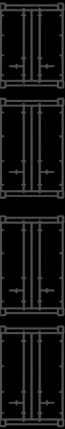
Appears at the top of the screen, using sweeping cone lasers and vertical tracking beams to control large portions of the play area.

Stage 2 – Clamp Formation

A second boss joins at the bottom, mirroring the top boss's movements to trap the player in a deadly crossfire

Black Hole Attack

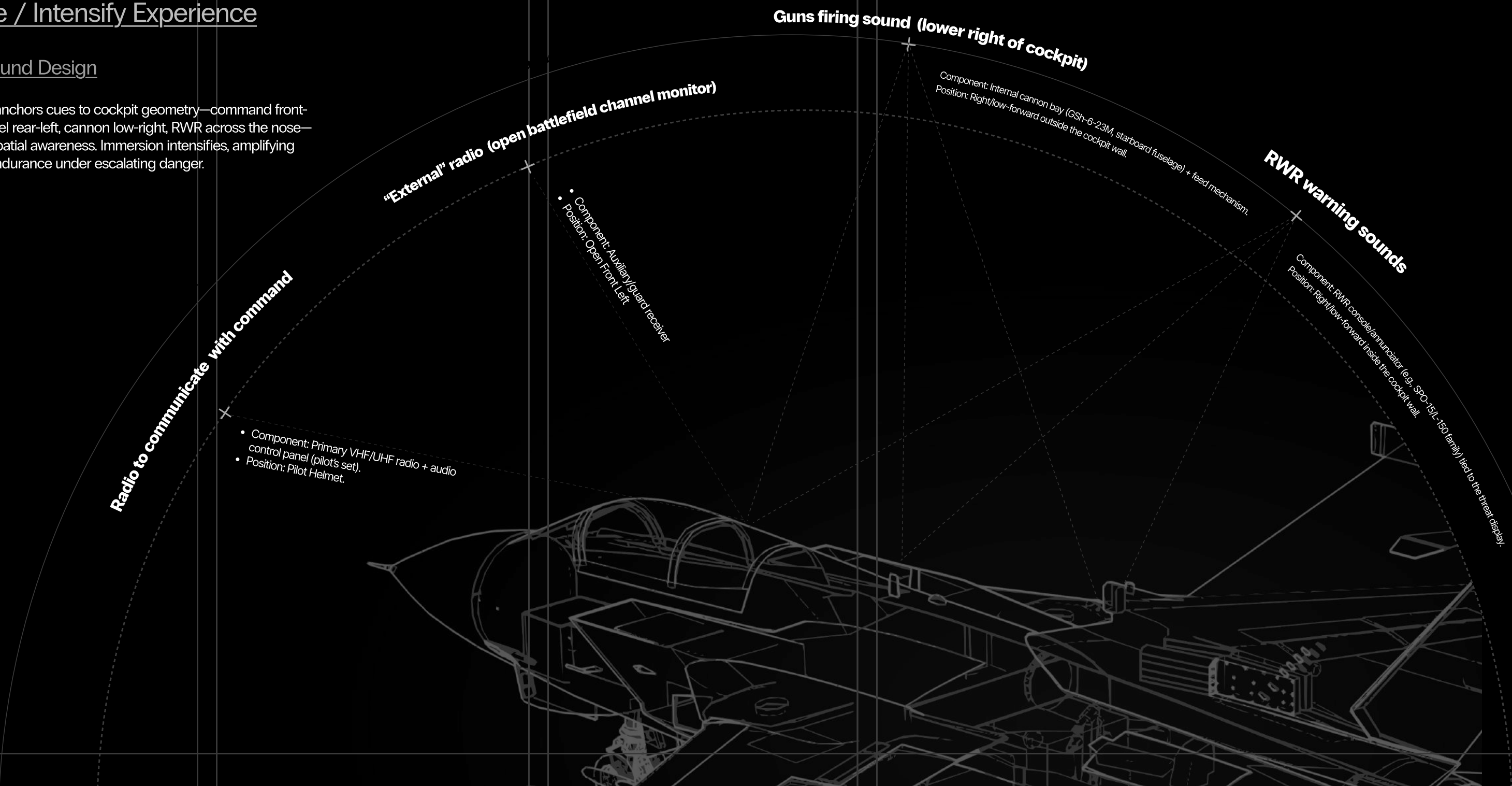
Both bosses periodically spawn a black hole between them, pulling in all objects — instant death for anything caught.



Immersive / Intensify Experience

Surround Sound Design

Surround sound anchors cues to cockpit geometry—command front-right, open channel rear-left, cannon low-right, RWR across the nose—building instant spatial awareness. Immersion intensifies, amplifying TRACER's core endurance under escalating danger.



Narrative Design

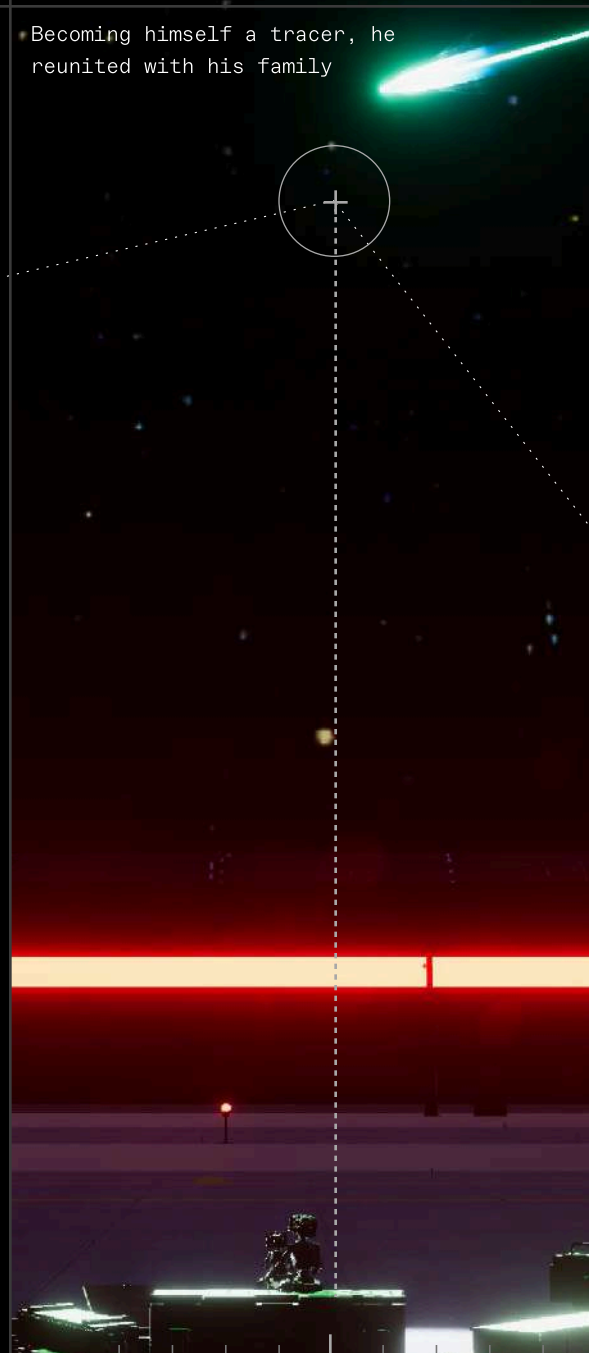
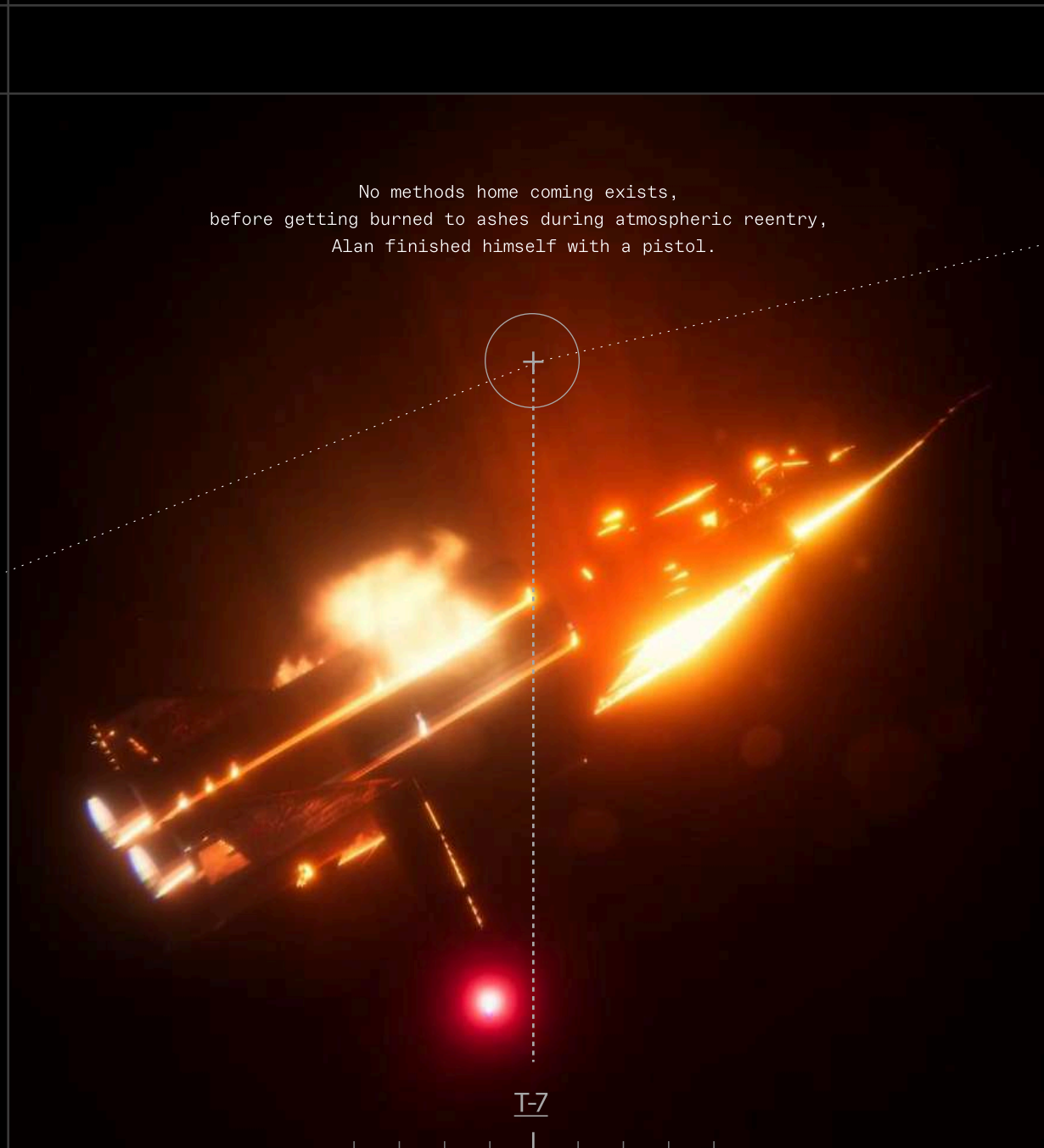
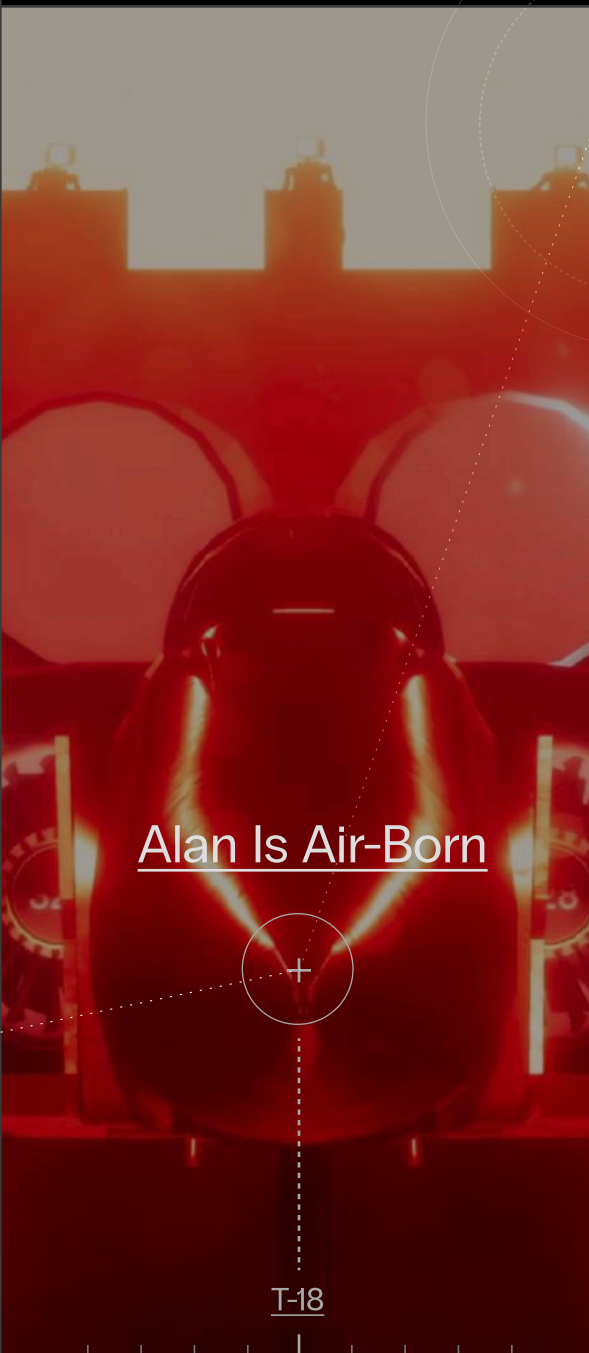
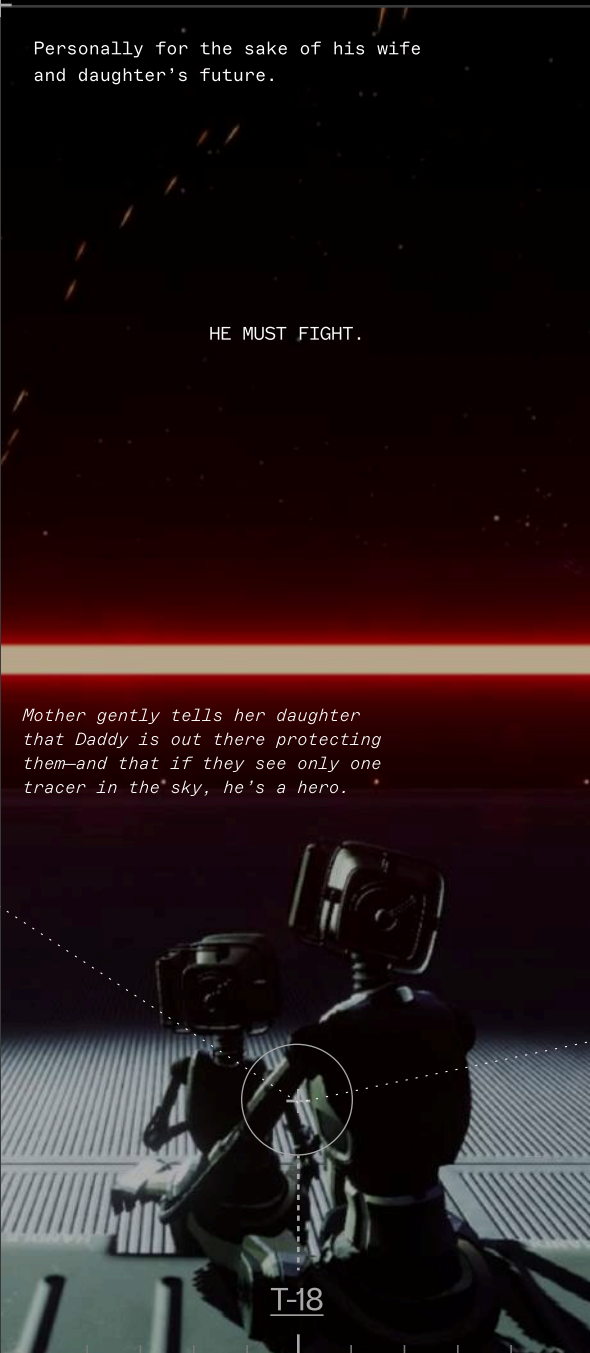
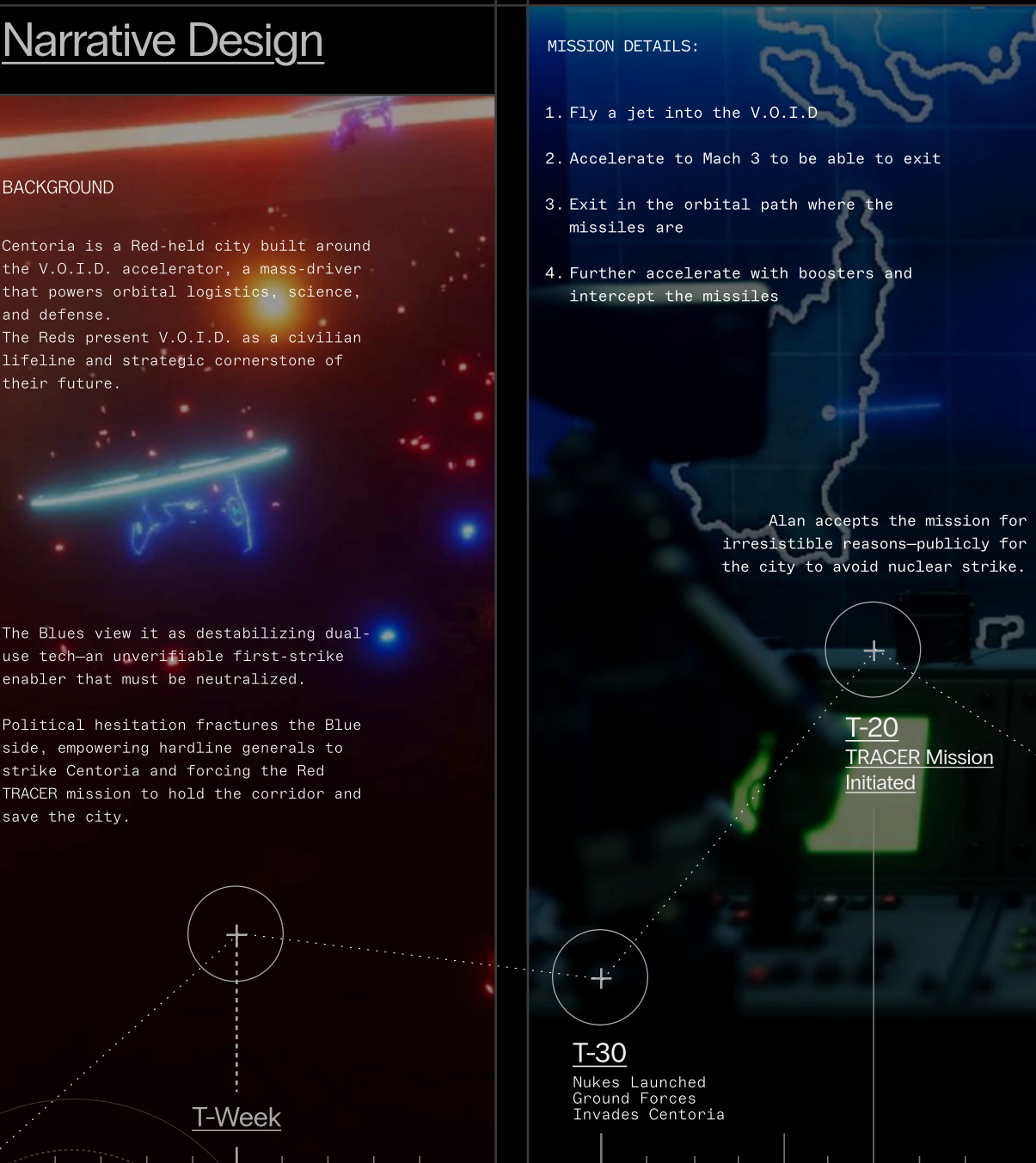
BACKGROUND

Centoria is a Red-held city built around the V.O.I.D. accelerator, a mass-driver that powers orbital logistics, science, and defense. The Reds present V.O.I.D. as a civilian lifeline and strategic cornerstone of their future.

The Blues view it as destabilizing dual-use tech—an unverifiable first-strike enabler that must be neutralized.

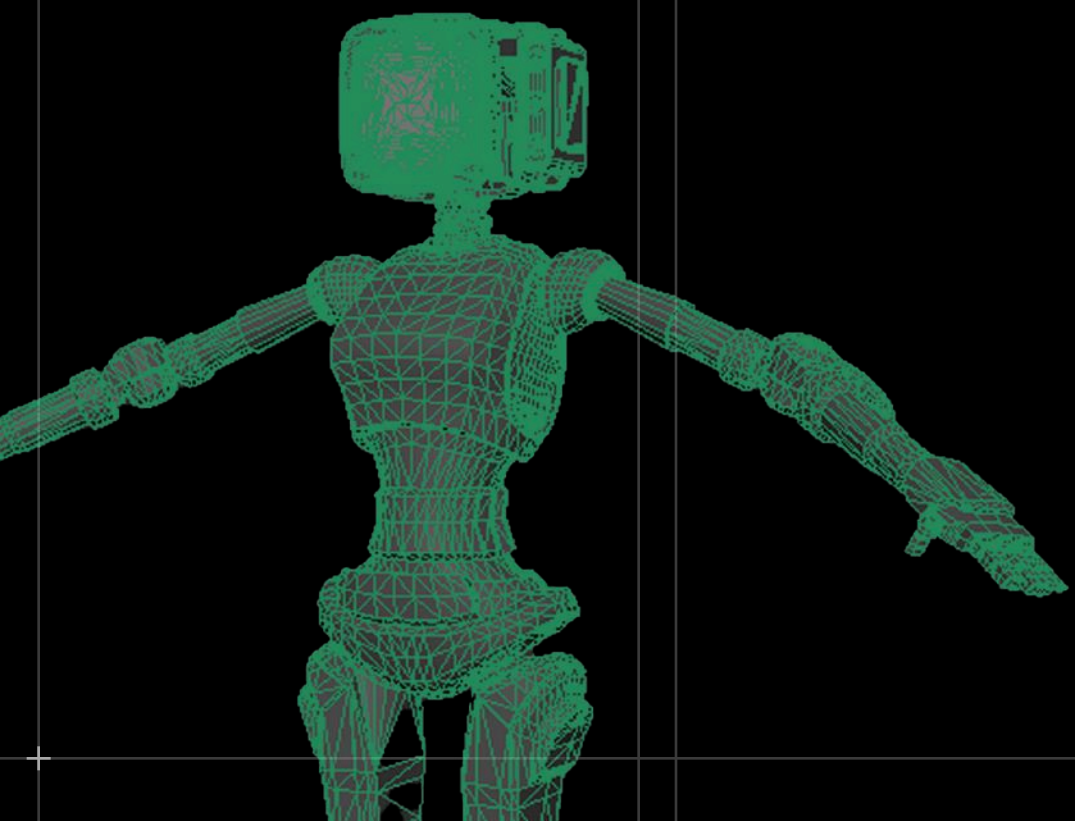
Political hesitation fractures the Blue side, empowering hardline generals to strike Centoria and forcing the Red TRACER mission to hold the corridor and save the city.

Cause:
The driving reason that makes the risk worth taking



Artistic Development

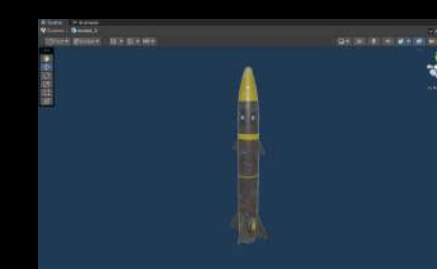
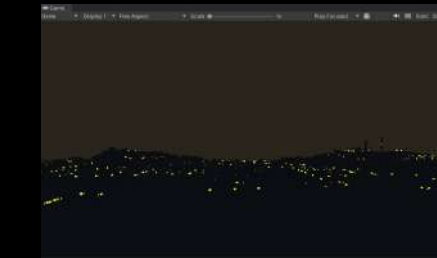
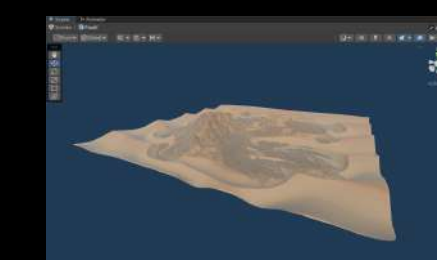
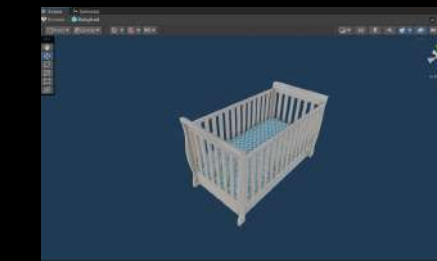
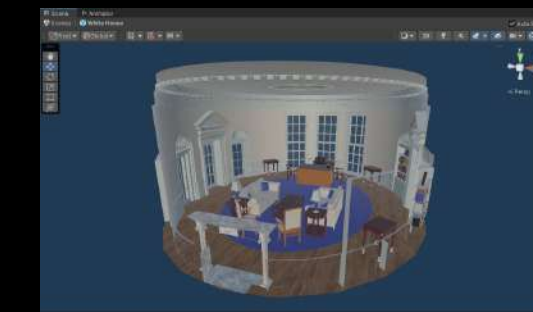
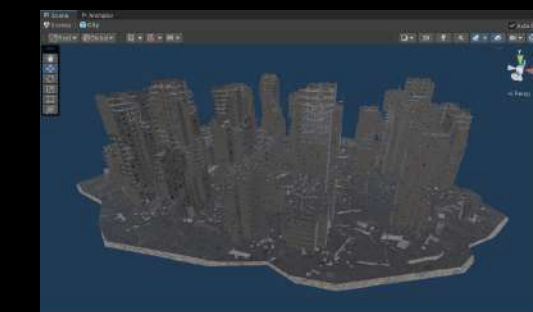
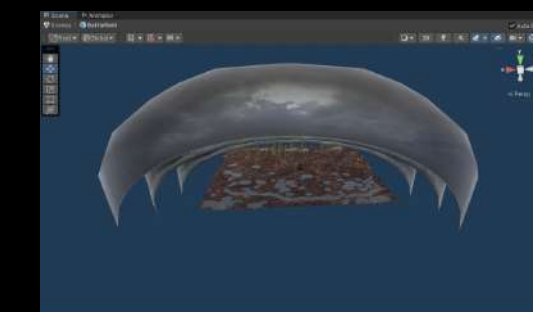
1. Character Design



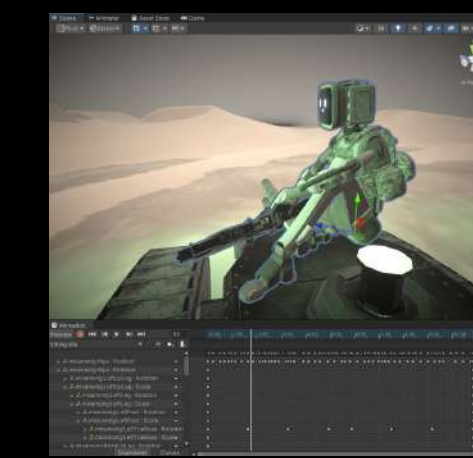
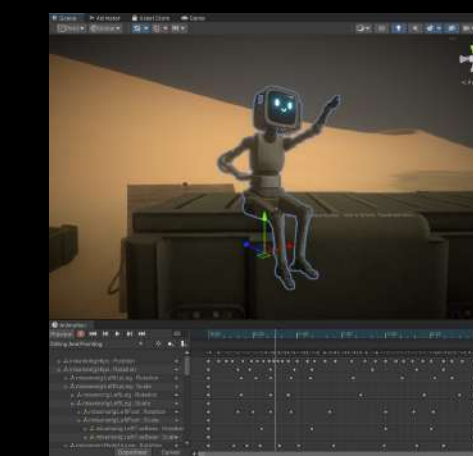
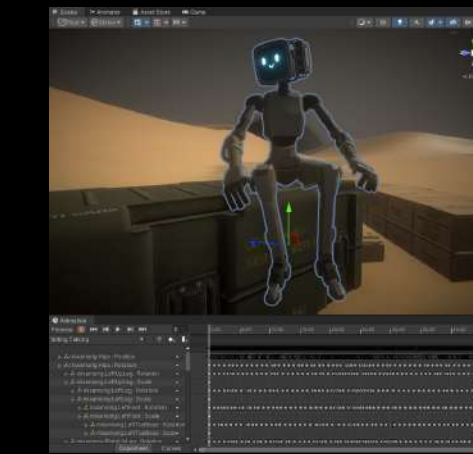
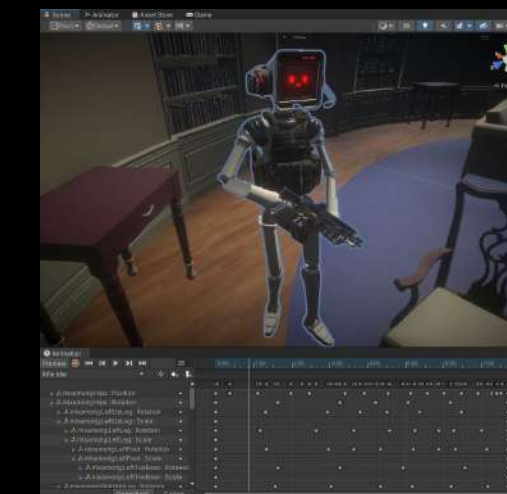
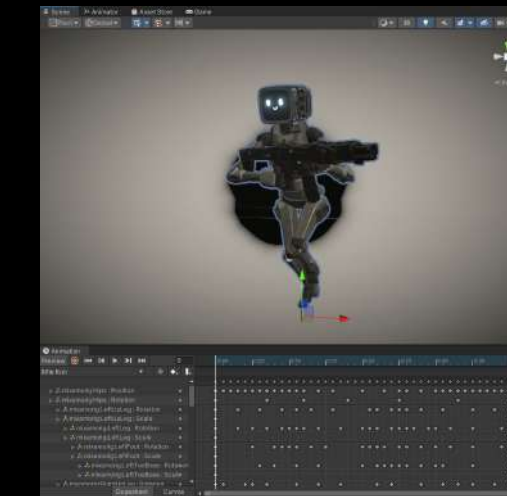
2. Key Assets Design



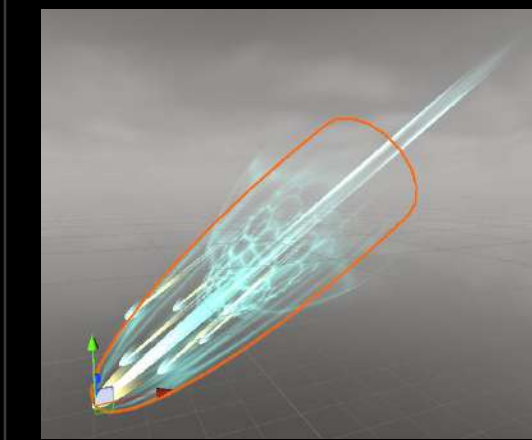
3. Other Materials From Assets Store



4. Animation



4. VFX Assets & Adjustment



Production:

AI-Generated Voiceover

The voice lining in TRACER is done through an open source neural network model called "GPT-SoVITS".

GPT模型列表

GPT-SoVITS/pretrained_models/s1bert25hz_2h4

SVITS模型列表

GPT-SoVITS/pretrained_models/s2G488k.pth

请上传并填写参考信息

请上传1-10秒内参考音频，超过10秒模型会自动截取。

打开无参考文本模式。不填写参考文本相当于开启。

参考音频的语种

中文

请填写需要合成的目标文本和语种模式

需要合成的文本

需要合成的语种

中文

怎么切

不切

按空格一切

按中文句号、切

按英文逗号、切

按标点符号切

文本切分工具。太长的文本合成出来效果不一定好，所以太长建议先切。合成会根据文本的断行分开合成再拼起来。

需要合成的音频文本

凌西盼一切

凌西盼一切

按中文句号、切

按英文逗号、切

按标点符号切

切分机文本

合成语音

凌西盼一切

凌西盼一切

按中文句号、切

按英文逗号、切

按标点符号切

切分机文本

top_A: 5

top_p: 1

temperature: 1

Adler T1-e5.cpkt	2024/10/30 22:15	CKPT 文件	151.453 KB
Adler T1-e10.cpkt	2024/10/30 22:15	CKPT 文件	151.453 KB
Adler T1-e15.cpkt	2024/10/30 22:16	CKPT 文件	151.453 KB
COO Jabber T1-e5.cpkt	2024/10/30 17:44	CKPT 文件	151.453 KB
COO Jabber T1-e10.cpkt	2024/10/30 17:45	CKPT 文件	151.453 KB
COO Jabber T1-e15.cpkt	2024/10/30 17:46	CKPT 文件	151.453 KB
FemaleKid3-e5.cpkt	2024/7/2 7:36	CKPT 文件	151.453 KB
FemaleKid3-e10.cpkt	2024/7/2 7:36	CKPT 文件	151.453 KB
FemaleKid3-e15.cpkt	2024/7/2 7:36	CKPT 文件	151.453 KB
Karlach2-e5.cpkt	2024/7/7 22:29	CKPT 文件	151.453 KB
Karlach2-e10.cpkt	2024/7/7 22:29	CKPT 文件	151.453 KB
Karlach2-e15.cpkt	2024/7/7 22:29	CKPT 文件	151.453 KB
Mother T2-e5.cpkt	2024/11/3 15:33	CKPT 文件	151.453 KB
Mother T2-e10.cpkt	2024/11/3 15:33	CKPT 文件	151.453 KB
Mother T2-e15.cpkt	2024/11/3 15:34	CKPT 文件	151.453 KB
Mother1-e5.cpkt	2024/7/8 0:11	CKPT 文件	151.453 KB
Mother1-e10.cpkt	2024/7/8 0:11	CKPT 文件	151.453 KB
Mother1-e15.cpkt	2024/7/8 0:11	CKPT 文件	151.453 KB

Each character's voice corresponds to a custom model trained by myself using existing data online.

For spawning obstacles in sync with music

The obstacle class, handles health and player interactions with it

Production: Programming

Bosses attack score, defined with each axis' angle and facing reference

Script

LaserAttackController

Radar

Spot Light

Spot Light 2 (Light)

Attack Angles

0

Element 0

X: 135

Y: 90

Z: 180

W: 0

Element 1

X: 225

Y: 90

Z: 180

W: 1

Element 2

X: 140

Y: 100

Z: 190

W: 0

Element 3

X: 220

Y: 100

Z: 170

W: 1

Element 4

X: 140

Y: 100

Z: 100

W: 0

Tracer Angles

Element 0: 15

Element 1: 0

Rotate Duration: 0.1

Laser Rotate Duration: 0.3

Tighten Duration: 0.1

Initial Spot Angle: 30

Slow Time: SlowTimeManager (Slow Time)

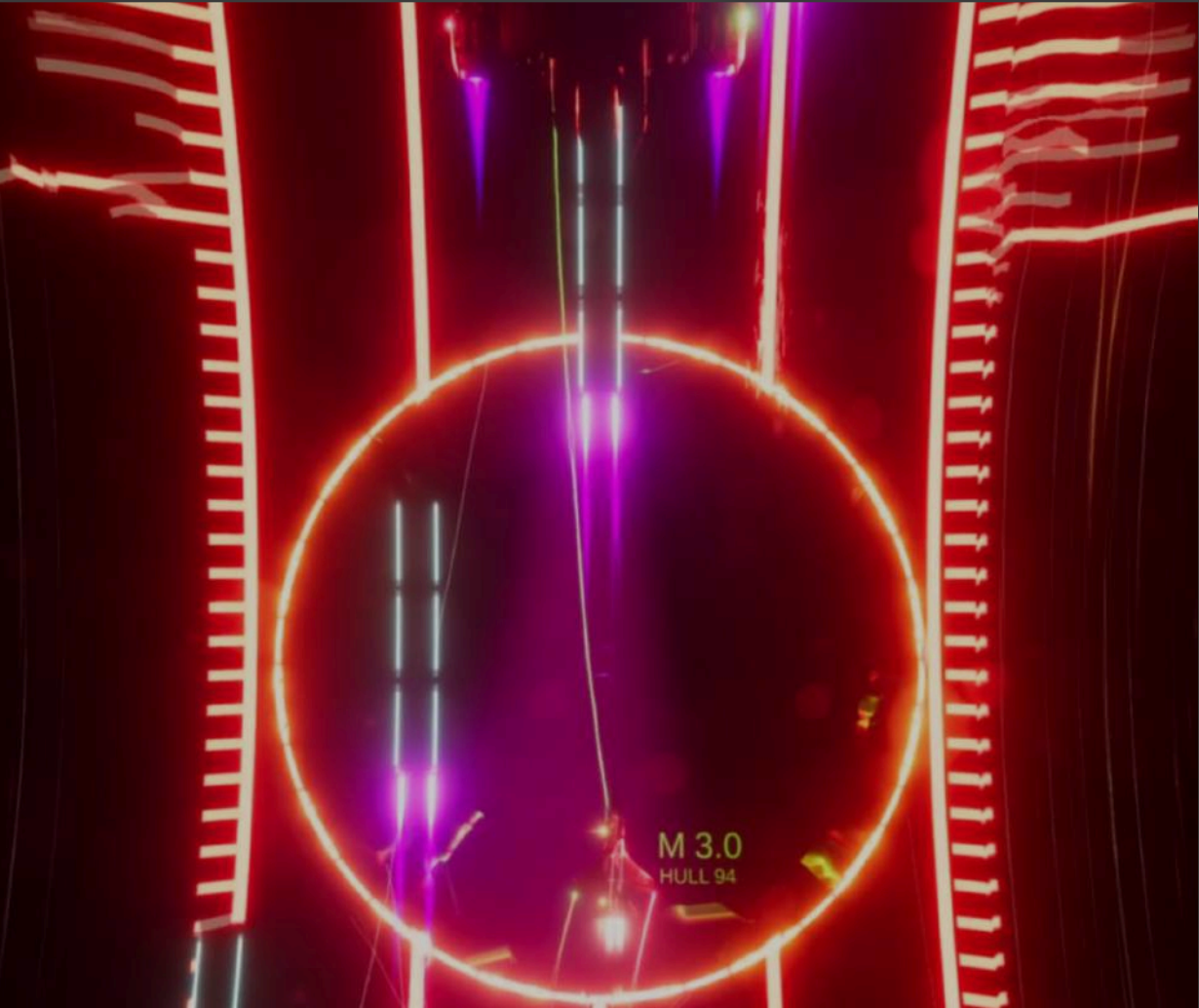
Handle boss laser attack according to given attack score

Handle boss movements

Handle the black hole objects behaviour, it attracts all attractable objects toward its center

The attractable class, all object with this class becomes an attractable object to the black hole

Production: Programming



Handle player movements, and invoke series of events when moving(eg. vfx)

AudioSpectrum is the engine of all audio sync elements, it retrieves the current active audio data from Unity

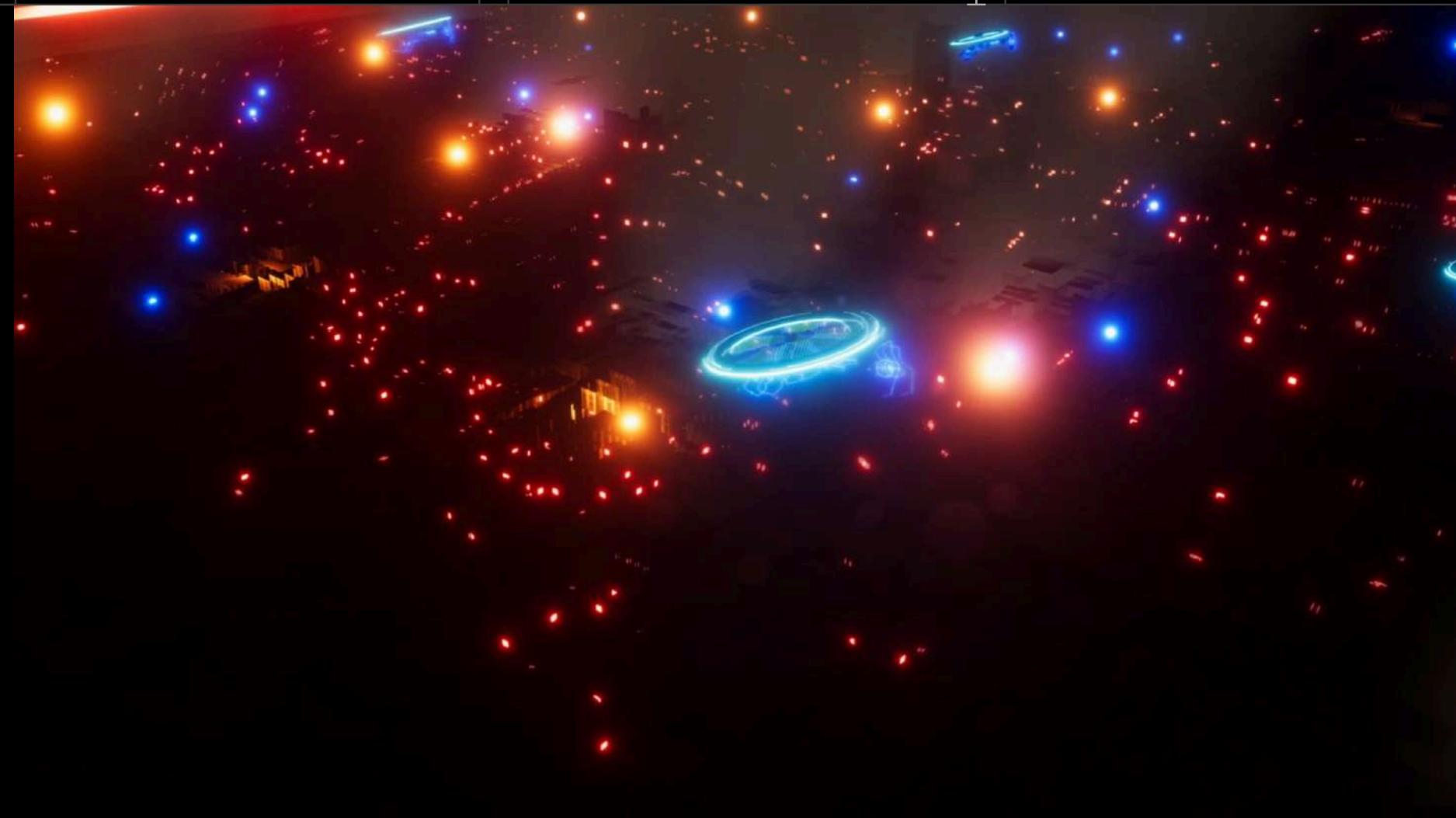
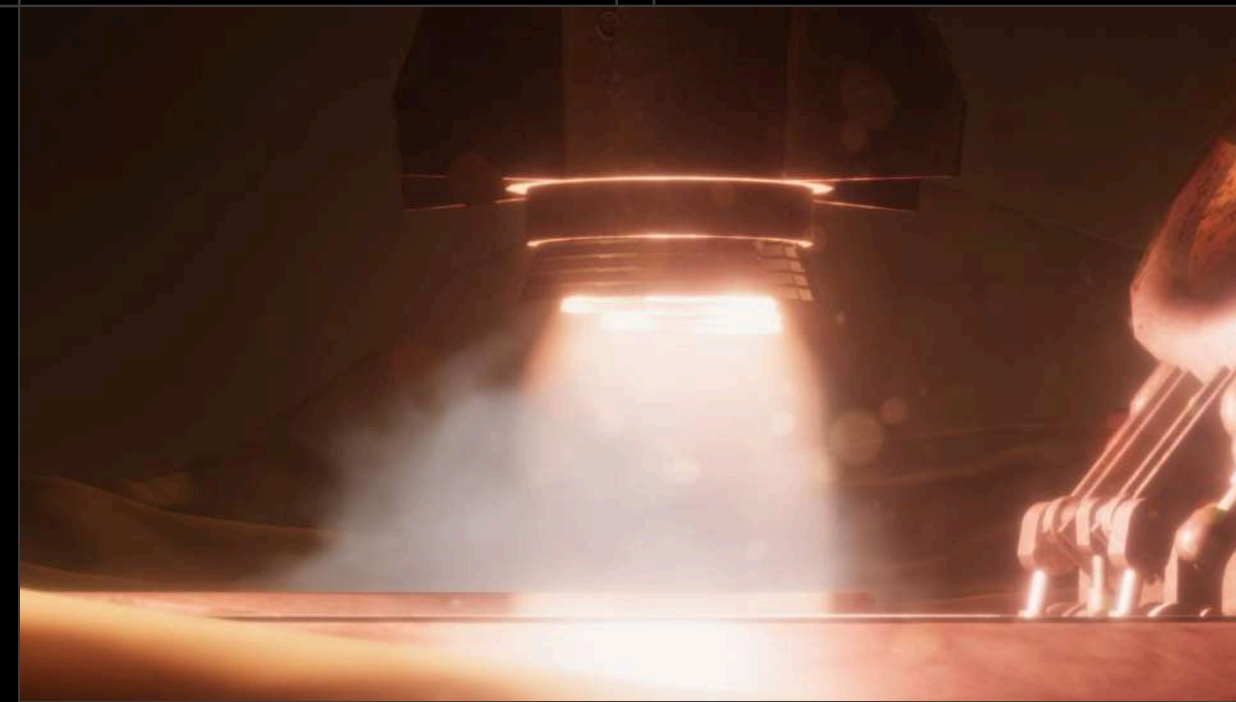
```

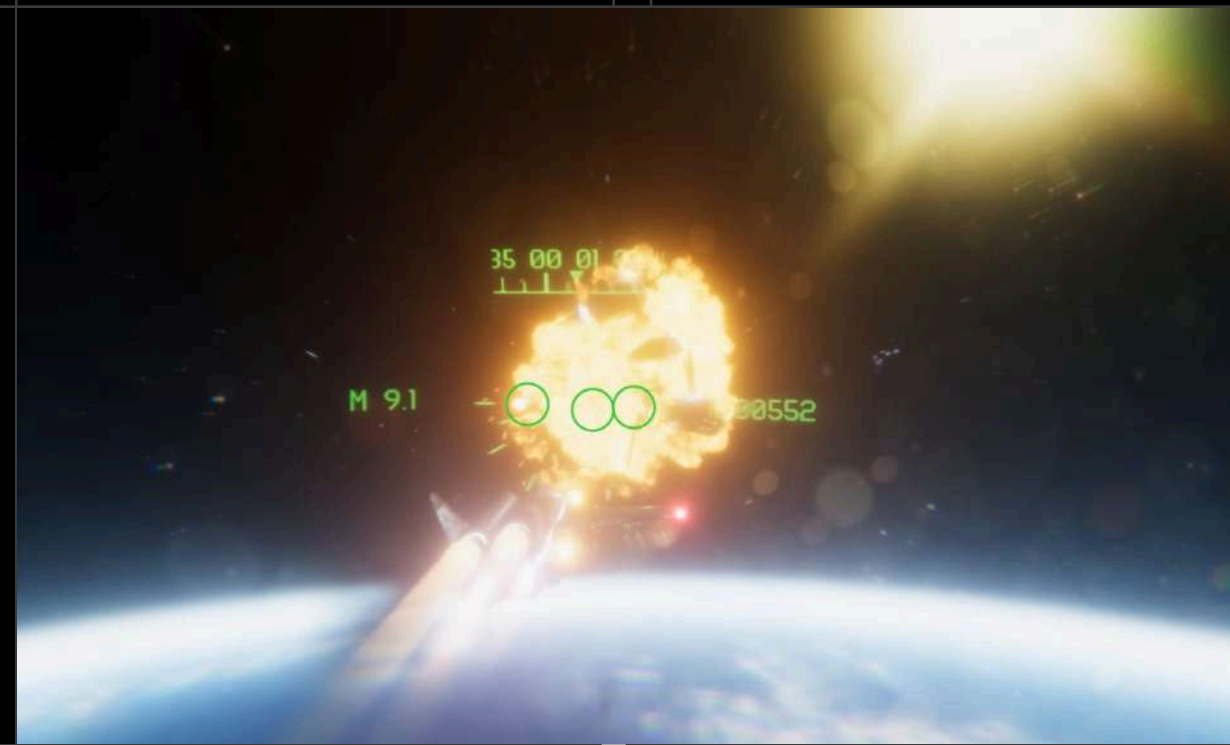
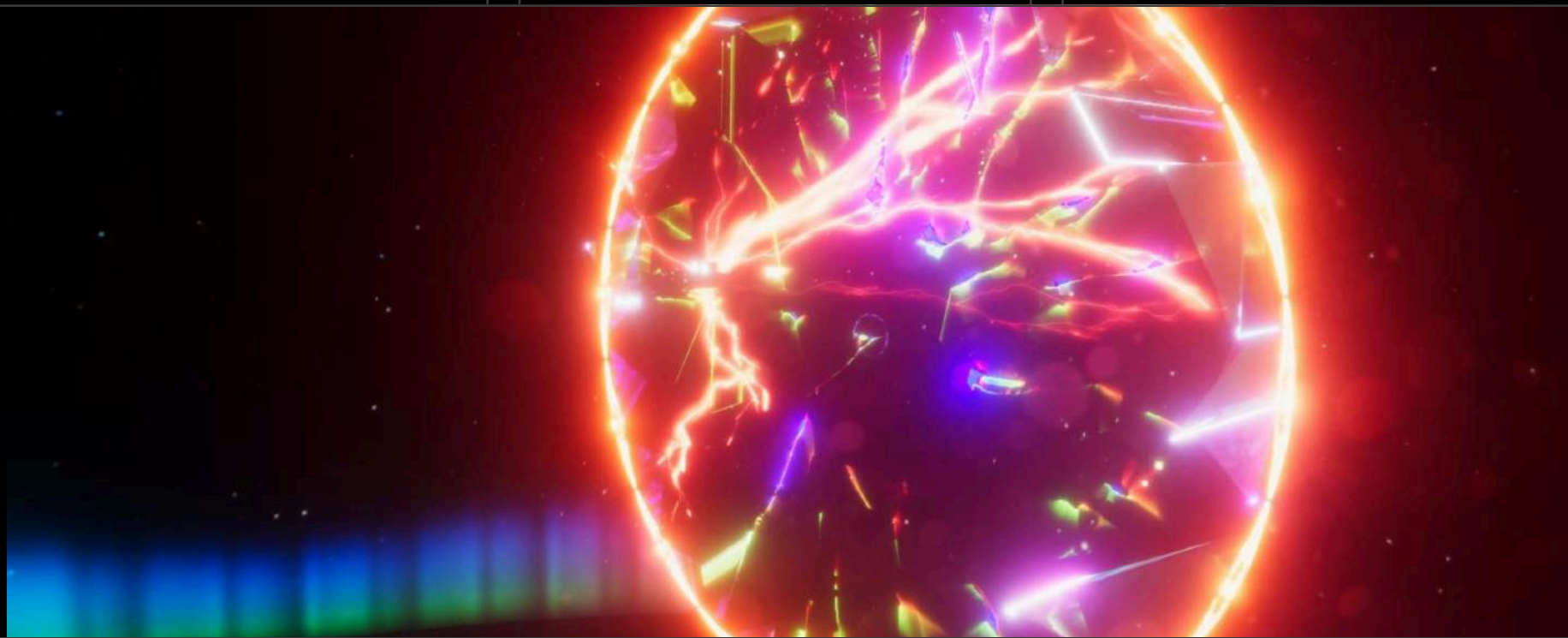
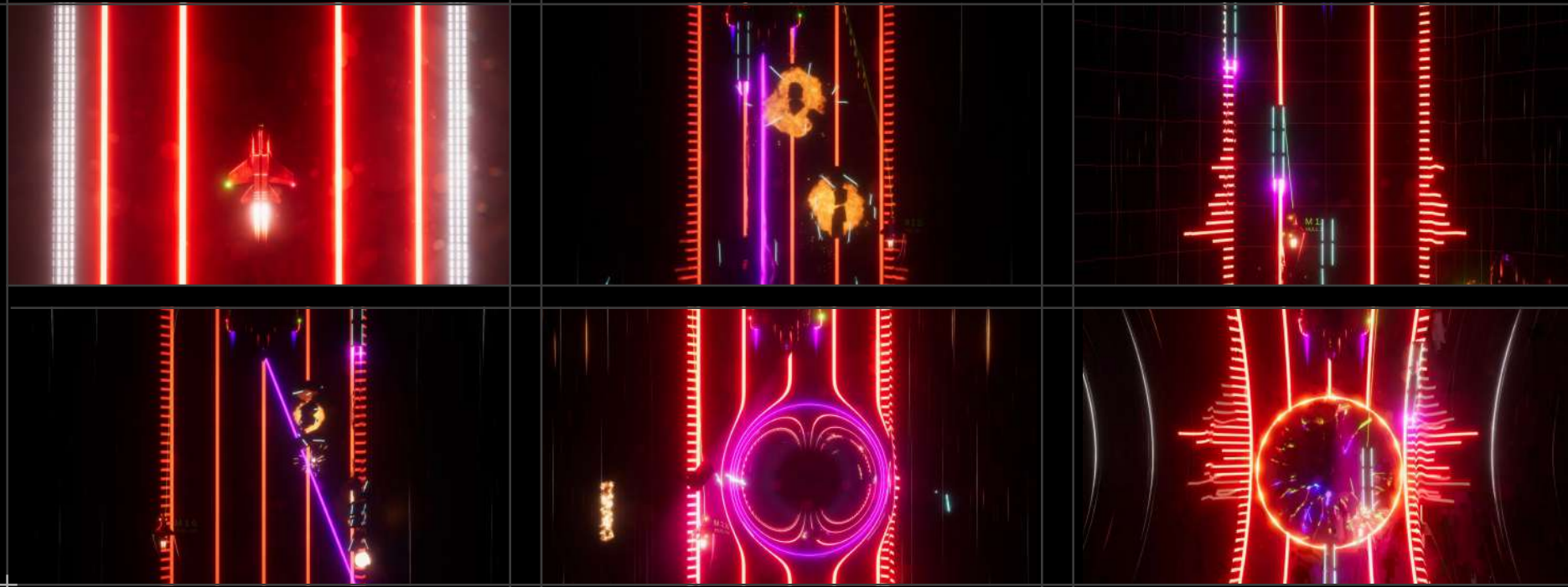
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 // Summary
6 // This "script" for analyzing spectrum data
7 // Few, free to get away in here for more accurate visualization
8 // Summary
9
10 // Unity 1.3 个引用
11 public class AudioSpectrum : MonoBehaviour {
12
13     // public AudioSource speaker;
14     // Unity 1.3 个引用
15     private void Update()
16     {
17         // get the data
18         // AudioListener.GetSpectrumData(audioSpectrum, 0, FFTWindow)
19         // speaker.GetSpectrumData(a_audioSpectrum, 0, FFTWindow)
20         // assign spectrum value
21         // this "require" format on the simplicity of other all
22         // , needing to compare one value (spectrumValue)
23         if (a_audioSpectrum != null && a_audioSpectrum.Length
24             {
25                 spectrumValue = a_audioSpectrum[0] * 100;
26                 specificSpectrumValue = a_audioSpectrum[0] * 100;
27             }
28     }
29
30     // Unity 1.3 个引用
31     private void Start()
32     {
33         // Initialize buffer
34         a_audioSpectrum = new float[128];
35
36         // This value served to AudioSource for best interaction
37         // 2 个引用
38         public static float spectrumValue { get; private set; }
39         // 2 个引用
40         public static float specificSpectrumValue { get; private set; }
41
42         // Unity fills this up for us
43         private float[] a_audioSpectrum;
44     }
45 }

```

To make the background lines show up and fades according to the drums in the music









Intro

PROTEST is a turn-based, competitive board game set in near-modern Eastern Europe, where 2 players take on the roles of protesters or police. The gameplay revolves around tactical decisions, with each side making strategic movements and actions. Protesters aim to storm the town hall with a large crowd, while police strive to prevent this by making arrests. Both sides have unique, progressively powerful actions as they advance. The game features intense strategy, evolving capabilities, and high-stakes confrontations, providing a thrilling and dynamic experience.



Contents

- Protest Leader Unit x1
- Crowd Unit x64
- Police Car Unit x2
- Police Station Token x1
- Town Hall Token x1
- D12 Die x1
- Protester Action Cards x20
- Police Action Cards x20
- Map x1
- Detention Cell x1
- Rulebook x1

SETTLEMENT AND WINNING CONDIT!ON

Protester

1 Crowd Absorption: If in a round, a crowd unit is an edge neighbour to the protest leader along the path that the protest leader had took within that round, that crowd unit will join the protesting crowd as the tail of the line. There can be up to 12 crowd units in the protesting crowd.

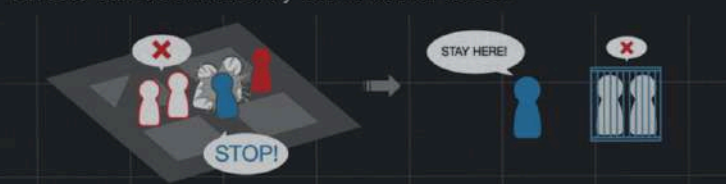


2 Winning Condition: If in a round, if the protesting crowd had 8 crowd units(excluding the protest leader), and the protest leader' s path steps through/on a non police occupied town hall, protester wins. Police can stay at most 1 turn at the townhall.



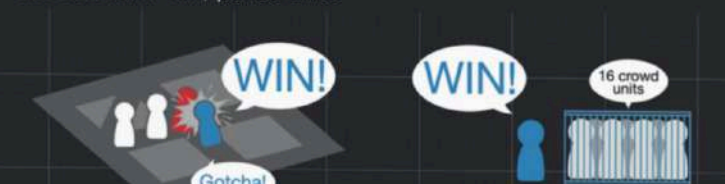
Police

1 Arrests: If in any moment, the police car collides with a protesting crowd unit, deduct 2 crowd units from the tail of the protesting crowd. The police adds 2 crowd units into detention cell. Arrests can be affected by active action cards.



2 Releases: If the police used an action card in a round, at the end of that round, release 1 crowd unit from detention cell and place it into the crowd bag.

3 Winning Condition: If in any moment, the police car collides with the protest leader and no action card is affecting arrests, police wins; or, in any moment, the police had 16 crowd units in the detention cell, police wins.



ACTION CARDS

Protester Card Cost

Protester' s can only play cards with a cost smaller than or equal to the amount of current crowd units in the protesting crowd(excluding the protest leader).

Police Card Cost

Police can only play cards with a cost smaller than or equal to the amount of crowd units in detention cell.

*Action cards are used on a depletion bases.

TURN STRUCTURE

1 At each turn, before making any movements, the player will choose to use a action card or not. If a card is played, the card effect will trigger according to its effect description.

2 Player will make a movement according to their movement rule.

3 Settlements

- The protester will check if the winning condition is met. If not, the protester player needs to calculate the amount of crowd units absorbed within this round.
- The police will check if the protest leader was arrested. If not, the police player needs to calculate if any arrests are made within this round.

4 If crowds are absorbed from the map, the protester needs to randomly draw the number of absorbed crowd units from the crowd bag and place them onto the map according to their assigned coordinates.

MOVEMENT

Protester

1 The protest leader must move to a connected road/cross-road/one-way trail block 6 units away. The movement speed can be affected by effects of active action cards.



2 Or the protester can U-turn; swap the protest leader with tail crowd unit of the protesting line, hence facing the opposite direction. Once a swap is made, deduct the new tail crowd unit from the protesting crowds and move to detention cell. No movements can be made if protester choose to swap.

TURN STRUCTURE

1 At each turn, before making any movements, the player will choose to use a action card or not. If a card is played, the card effect will trigger according to its effect description.

2 Player will make a movement according to their movement rule.

3 Settlements

- The protester will check if the winning condition is met. If not, the protester player needs to calculate the amount of crowd units absorbed within this round.
- The police will check if the protest leader was arrested. If not, the police player needs to calculate if any arrests are made within this round.

4 If crowds are absorbed from the map, the protester needs to randomly draw the number of absorbed crowd units from the crowd bag and place them onto the map according to their assigned coordinates.

MOVEMENT

Protester

1 The protest leader must move to a connected road/cross-road/one-way trail block 6 units away. The movement speed can be affected by effects of active action cards.



2 Or the protester can U-turn; swap the protest leader with tail crowd unit of the protesting line, hence facing the opposite direction. Once a swap is made, deduct the new tail crowd unit from the protesting crowds and move to detention cell. No movements can be made if protester choose to swap.

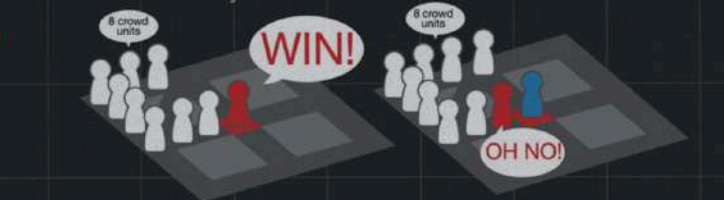
SETTLEMENT AND WINNING CONDIT!ON

Protester

1 Crowd Absorption: If in a round, a crowd unit is an edge neighbour to the protest leader along the path that the protest leader had took within that round, that crowd unit will join the protesting crowd as the tail of the line. There can be up to 12 crowd units in the protesting crowd.

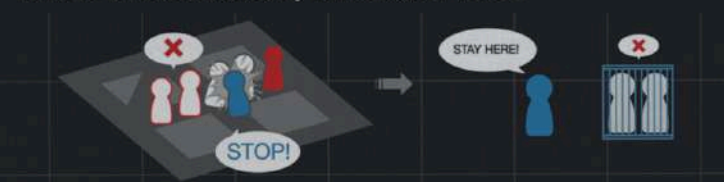


2 Winning Condition: If in a round, if the protesting crowd had 8 crowd units(excluding the protest leader), and the protest leader' s path steps through/on a non police occupied town hall, protester wins. Police can stay at most 1 turn at the townhall.



Police

1 Arrests: If in any moment, the police car collides with a protesting crowd unit, deduct 2 crowd units from the tail of the protesting crowd. The police adds 2 crowd units into detention cell. Arrests can be affected by active action cards.



2 Releases: If the police used an action card in a round, at the end of that round, release 1 crowd unit from detention cell and place it into the crowd bag.

3 Winning Condition: If in any moment, the police car collides with the protest leader and no action card is affecting arrests, police wins; or, in any moment, the police had 16 crowd units in the detention cell, police wins.



ACTION CARDS

Protester Card Cost

Protester' s can only play cards with a cost smaller than or equal to the amount of current crowd units in the protesting crowd(excluding the protest leader).

Police Card Cost

Police can only play cards with a cost smaller than or equal to the amount of crowd units in detention cell.

*Action cards are used on a depletion bases.

Inspiration

The inspiration for creating "REVOLT" came from my personal experiences with the protests that occur every Saturday below my apartment in Toronto. As a photography lover, I would often join the protest crowds to take photos, capturing the energy and emotion of the events.

Being amidst the protesters and observing their strategies and movements gave me the idea to turn these real-life experiences into a board game. I wanted to create a game that reflects the dynamic and strategic nature of protests, allowing players to experience the intensity and tactics involved in these powerful social movements.



Goal To Achieve

I aim to create a tactic-oriented board game featuring an interactive board that represents the city's streets and intricately detailed pawn figures.

Providing detailed representation for the properties and dynamics of the protesting event through movement tactics.

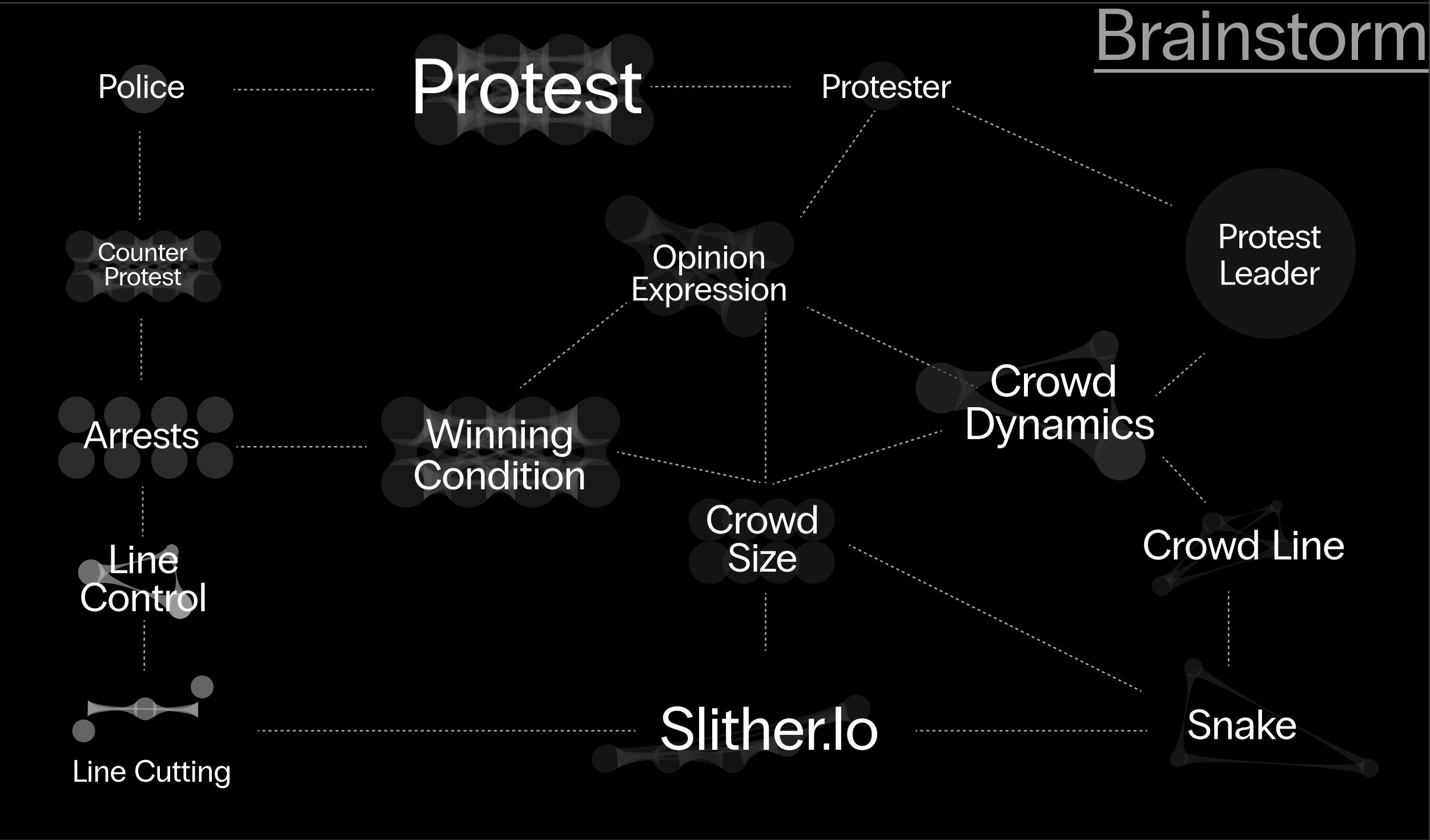
Tension to Recreate in through gameplay

I aim to recreate the complex contests' experiences through this game.

The protesting player should acknowledge the difficulties in the process of expressing their will.

The police should experience the challenges faced while stopping the tide.

Brainstorm



Prototype

Initial Map



Initial Movements

Movement:
Protesters move to connected road/
crossroad blocks.

Speed Formula:
Speed=
Clamp[2,7]
(Initial Crowd Units-Current Protester Units)

Key Effect:
More protester units → slower speed;
fewer protester units → faster speed.

Protester Movement (Slither.io style inspired):

The protest leader moves first, and all
other crowd units follow along the exact
path the leader took.

Police can move between connected
cross roads, the movement speed is
fixed at 1 crossroads.

Initial Map Settlements



Player Unit Settlements



Whenever the protest leader
became a neighbor of a crowd unit,
that crowd unit will join and became
the new tail of the protesting line.

Whenever the protester or the police
became a neighbor of a event block,
a event card will be drawn at random
and trigger its effects.

Isomorphic to the elimination
mechanism of Slither.io, when the
police collides with a non-protest
leader protesting unit, the protesting
line will be cut off from the location
of impact.

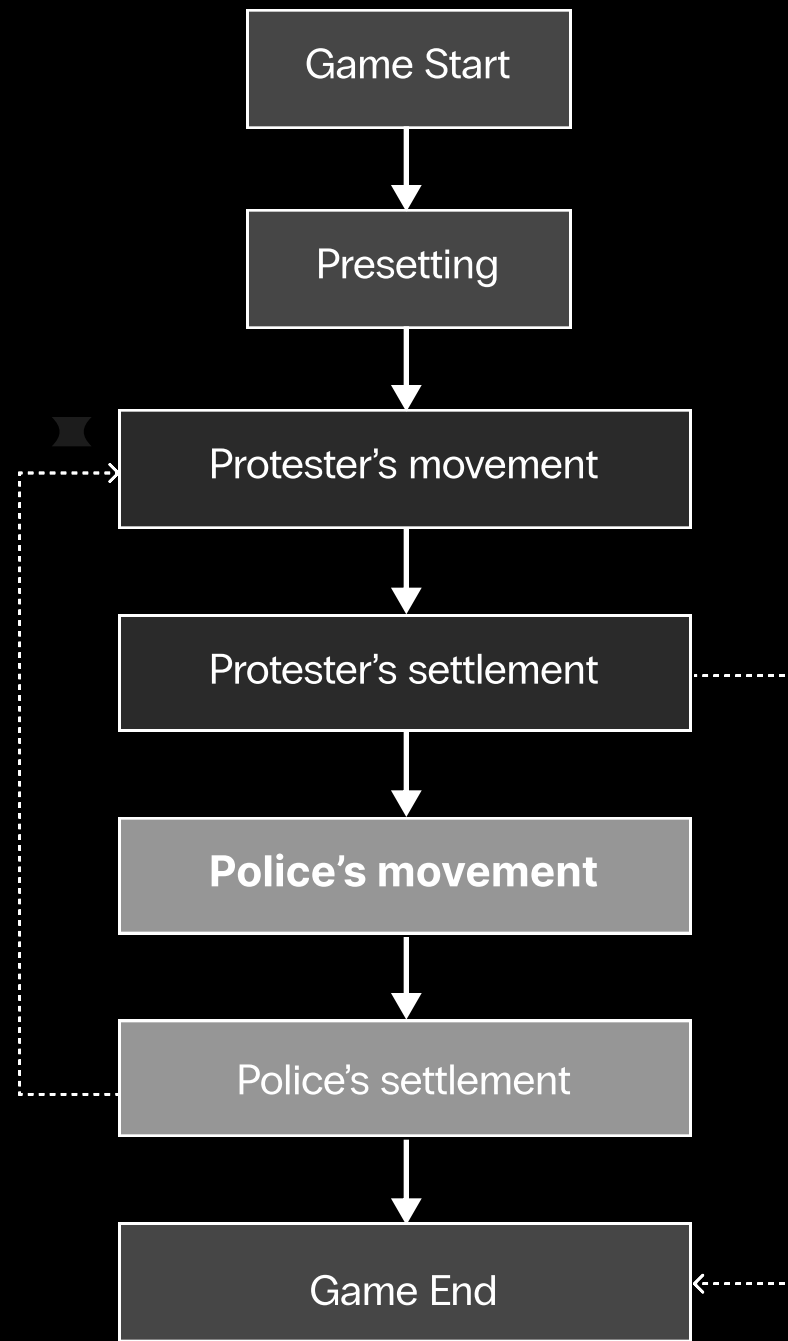
Whenever the police collides with
the protest leader, police wins and
the game is over.

Core Objectives

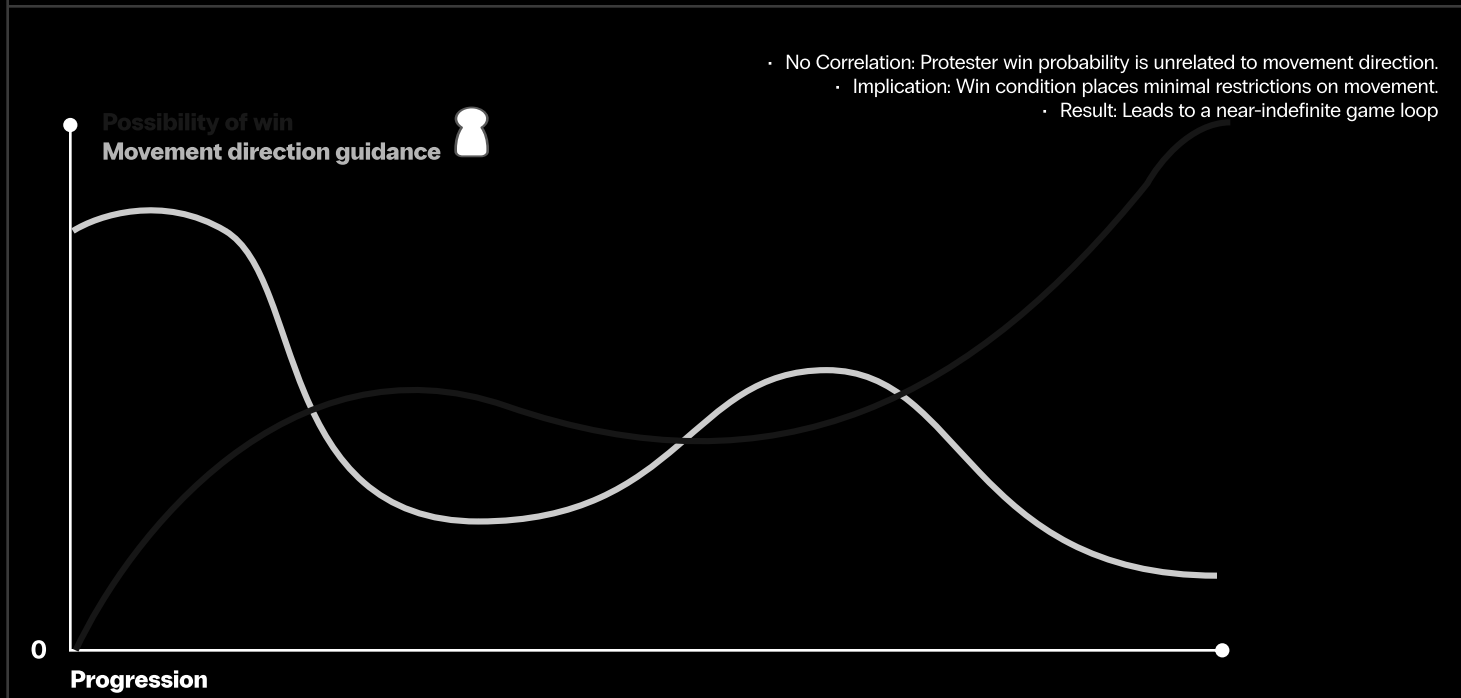
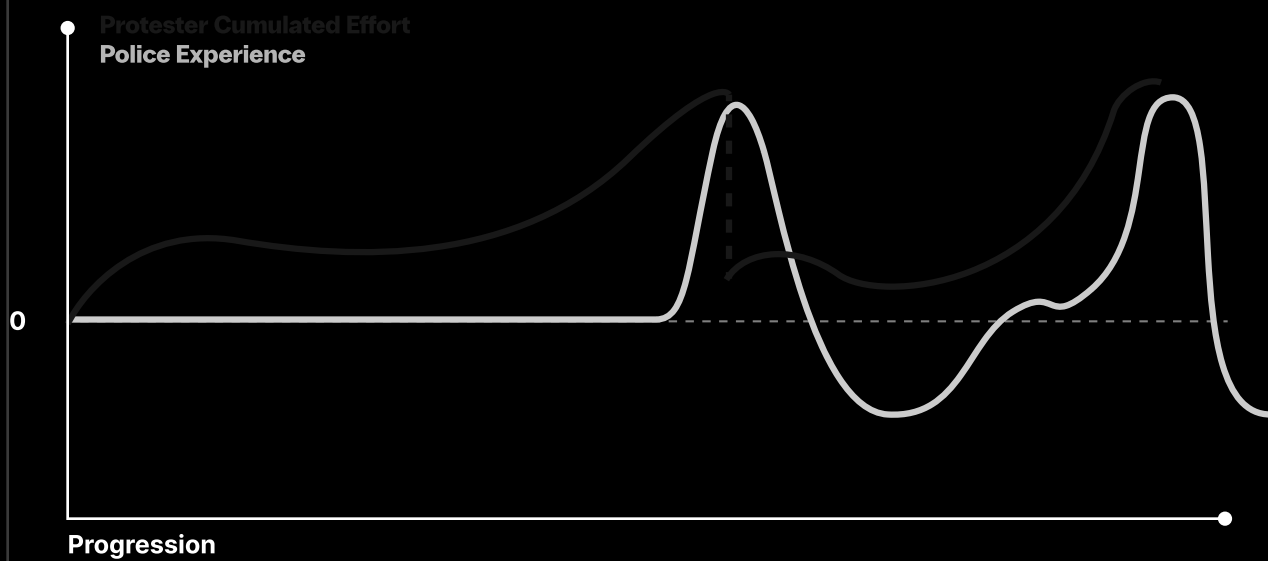
The protester's objective is to reach 10
crowd units in the protesting line at any
moment of the game, while he police's
objective is to arrest the protest leader
at any moment of the game.



Current Game flow



Test & Result

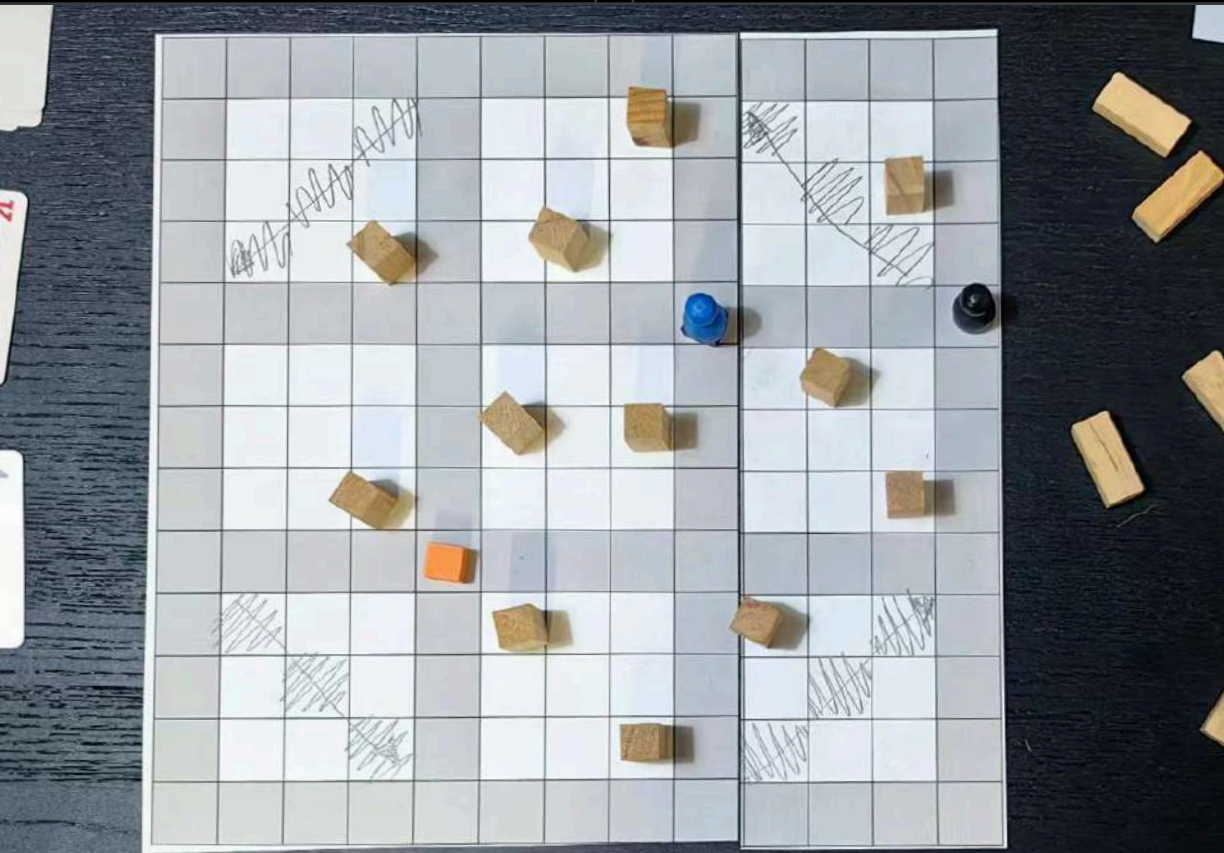


Prototype

Version 2.0 Updates

Map:

The final map is composed by 13 x 13 blocks, where each block gets assigned a type according to their neighbor compositions.



Empty Block:

Adjacent to only one road block or diagonally to a crossroad, not a one-way bridge.

Road:

Connected to blocks on one or opposite sides.

Crossroad:

Connected to multiple road blocks from different sides.

One-way Bridge:

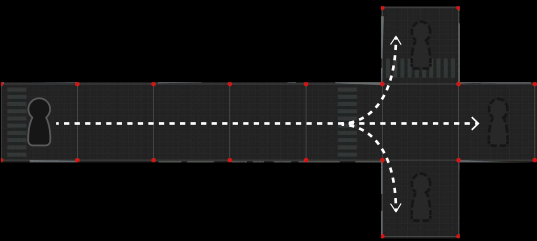
Fixed-location block.

Town Hall: Protester-selected crossroad.

Movements

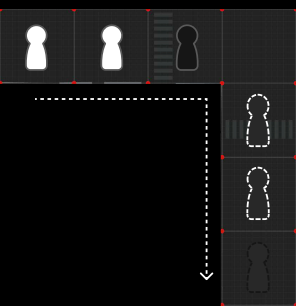
Protester Movement:

Can move up to 6 connected blocks (road, crossroad, or one-way bridge); speed may increase via action cards.



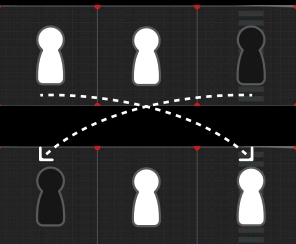
Movement Rule:

Protest leader moves first; crowd units follow the leader's path.



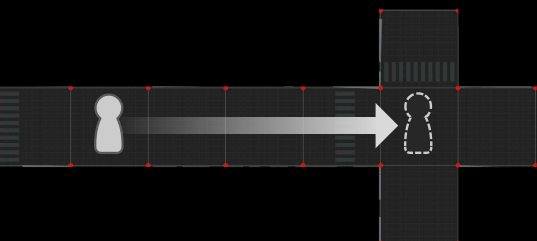
U-Turn Rule:

Protesters may swap head and tail instead of moving, skipping that round's movement.



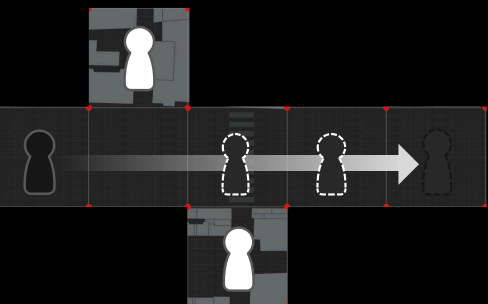
Police Movement:

Moves only between connected crossroads, avoiding protesters; normal speed: 1 crossroad per move.



Map Interactions

Whenever the protest leader became a neighbor of a crowd unit, that crowd unit will join and became the new tail of the protesting line.

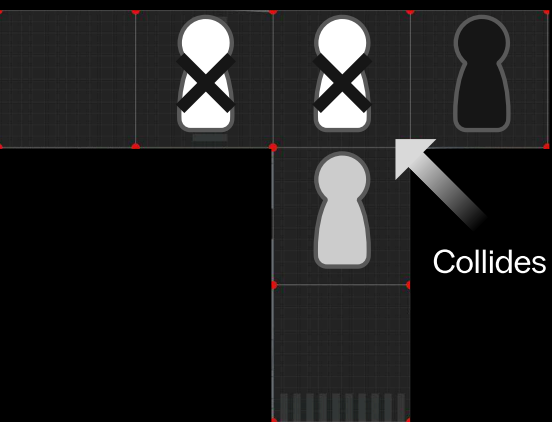


When any player steps on a one-way bridge, that player can only move towards the indicated direction of the bridge.



Player Unit Interactions

Hitting a non-leader removes 2 protester units to detention.



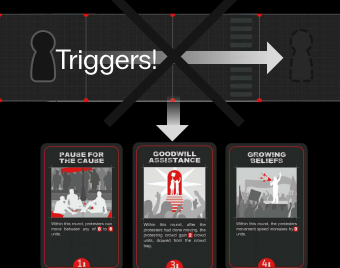
Turn Structure

Skill Phase:

Before moving, players may optionally play one affordable skill (consumed on use).



Action Cards



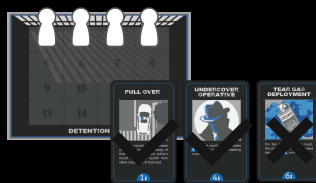
Events as Skills:

Events become playable skills with costs, calculated differently for protesters and police.



Protester Skill Limit:

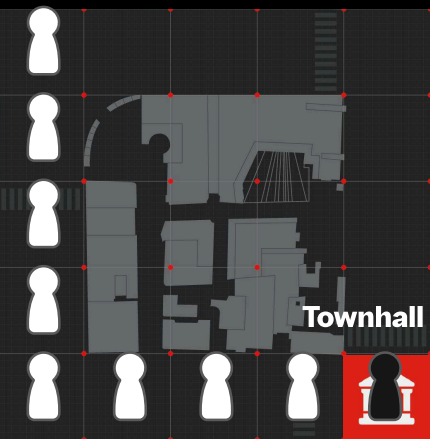
May use skills with cost ≤ current protester unit count.



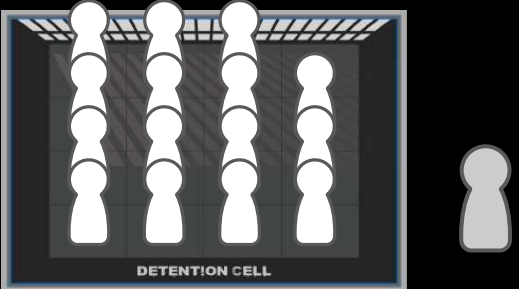
Protester Skill Limit:

May use skills with cost ≤ current protester unit count.

Core Objectives



The protester's objective is to cross-over with the town hall while maintaining at least 8 crowd units.

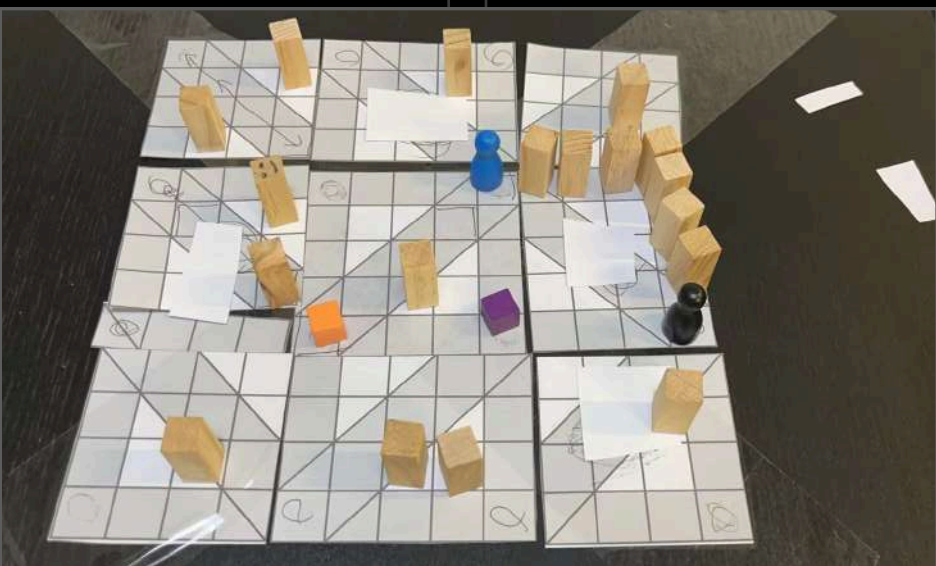


The police's objective is to arrest 16 crowd units or arrest the protest leader.

Test & Results



The U-turn diminishes the police's efforts of constructing a check-mate situation, while it effectively costs none to protesters.



Also, we noted that the protester would U-turn if and only if a check-mate situation if formed.

Further Test

Setup:
The protesters' speed is $s=6$. The protest leader's distance to the next crossroad is d , which can be 0, 1, 2, 3, 4, or 5.

Constraint:
 d cannot exceed 5; otherwise, the protesters would either reach or pass the crossroad after moving, placing them out of the police's **reachable range**.

Assumption:
The police can reach that crossroad in the upcoming round.

Key conclusion:
If the protesters' team has as many or more crowd units than the difference between their speed and the current distance ($s-d$), then at least one of those units will collide with the police.

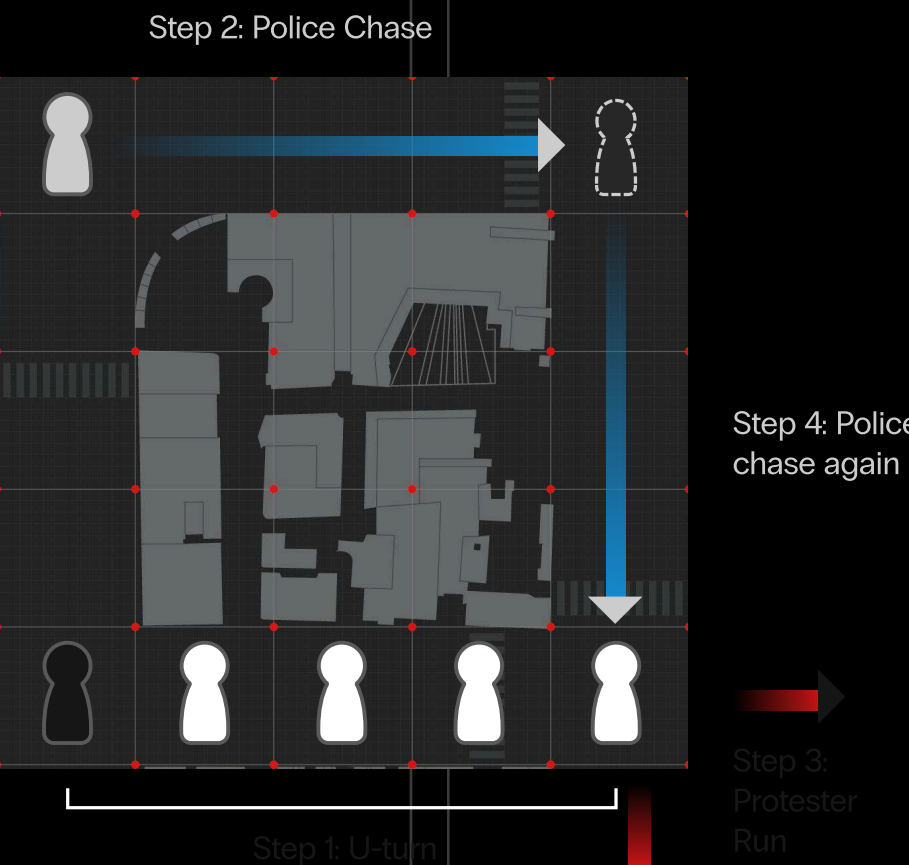
d	5	4	3	2	1	0
s-d	1	2	3	4	5	6

Notice that we omitted the $d = 6$ case here, since that would cause a collision between the police and the protest leader, we assumes that our player is generally making rational decisions here.

Let P_2 be the probability that the protesters having more or equal to 5 crowd units. We know that the protesting crowd can hold up to 12 units, and there are 7 cases achieving conditions of P_2 . Therefore, it is obvious that P_2 is at least less than or equal to $7/12$.

Analysis

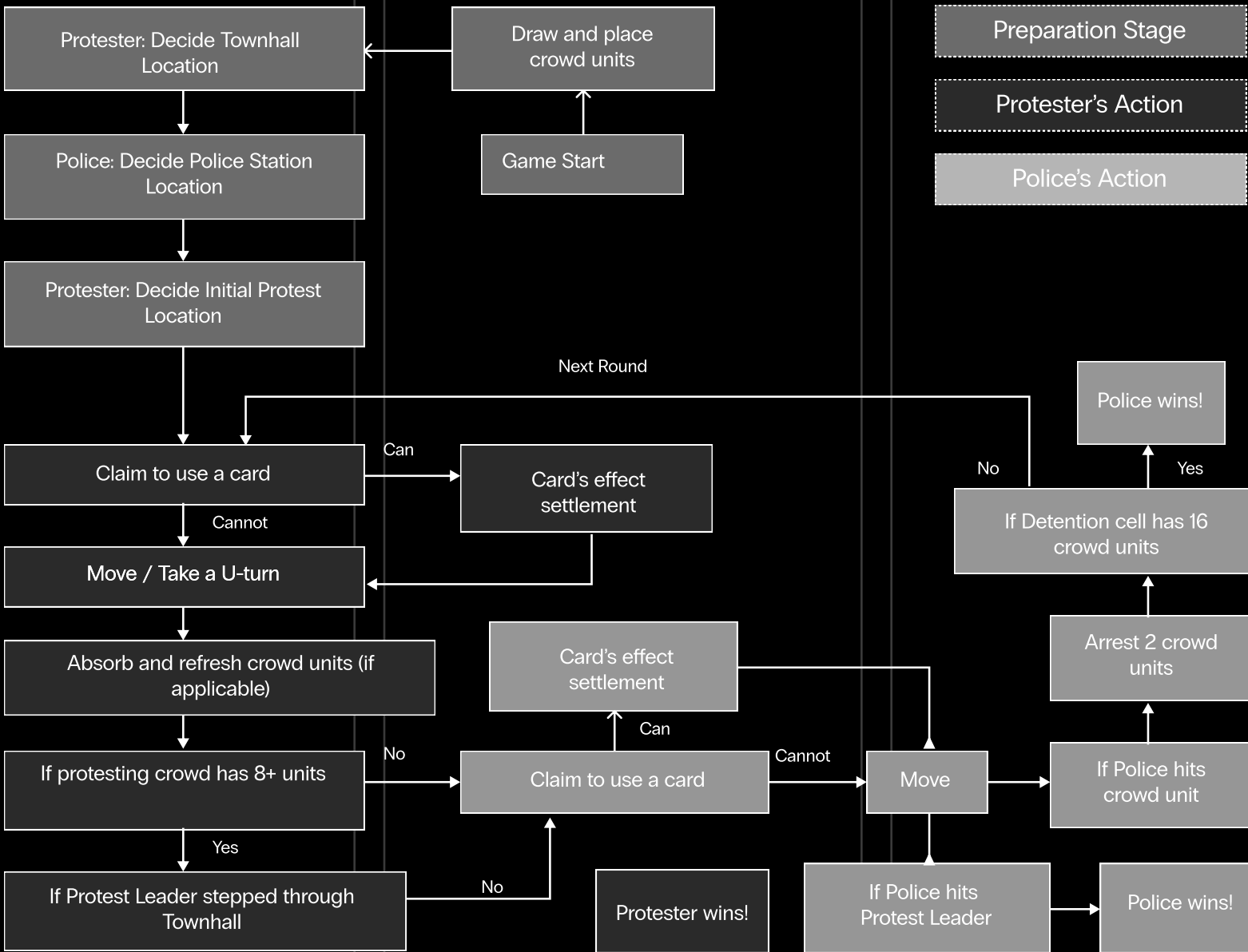
If protesters U-turn, collisions occur only with over five units or when cornered. Making a U-turn cost two units balances this but reduces police impact, as U-turns currently serve unintended roles.



Probability analysis of police colliding with protesters:

$P_1 = P(\text{Police forms a check-mate situation}) \leq 4/16 \leq 1/4$
 $P_2 = P(\text{Protesters had more than 5 units}) \leq 7/12$
 $\Rightarrow P(P_1 \text{ Intersects with } P_2) \leq 1/4 \times 7/12$
 $P(P_1 \text{ Intersects with } P_2) \leq 7/48 \leq 14.583\%$

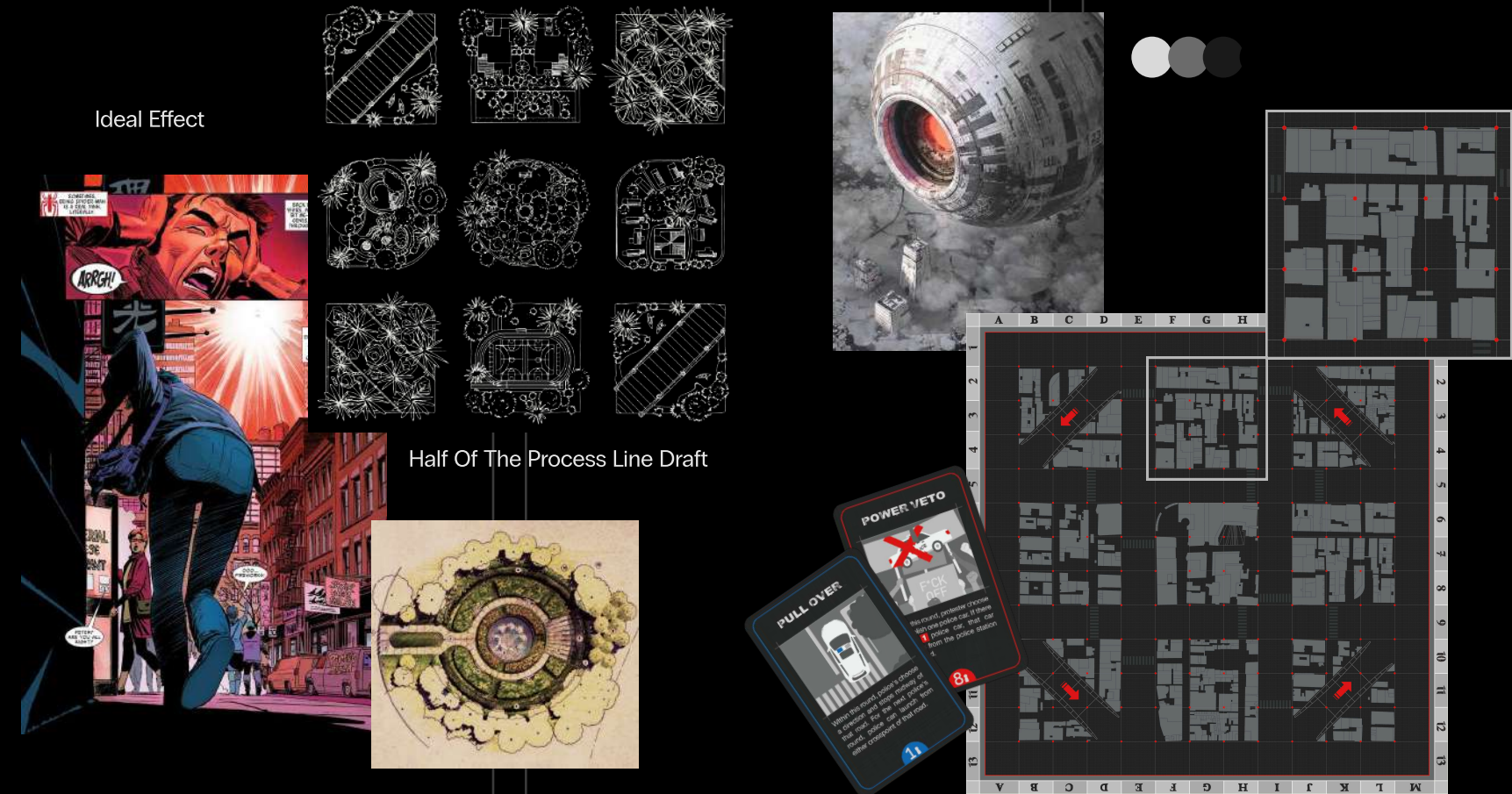
Game Flow



Visual Planning: The soviet-style

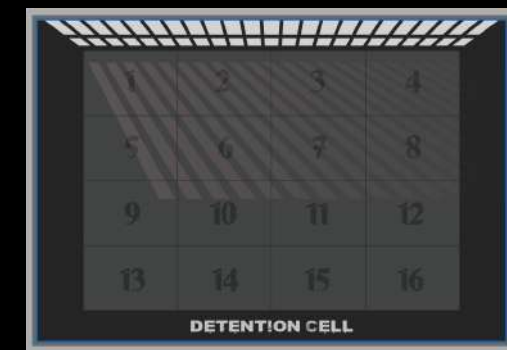
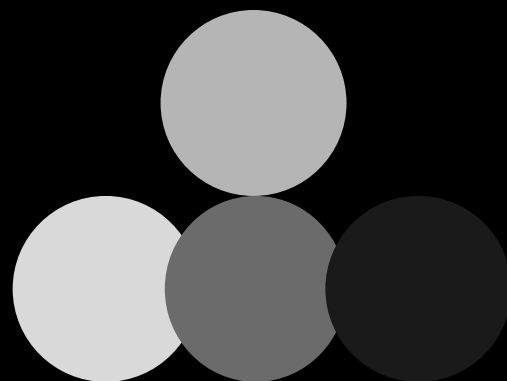
Original Idea:
For the "protest" theme, I want to show its harsh reality through irony — using a light, comic-style art to tell a heavy story, creating contrast and revealing the game's darker side.

Further Thoughts:
However, I later felt that such a serious story shouldn't be presented in a playful tone. Players should understand right away that this is a deeply serious board game. That's why I chose the Soviet-Style 'Hardcore' Aesthetics.



Aesthetic & Execution

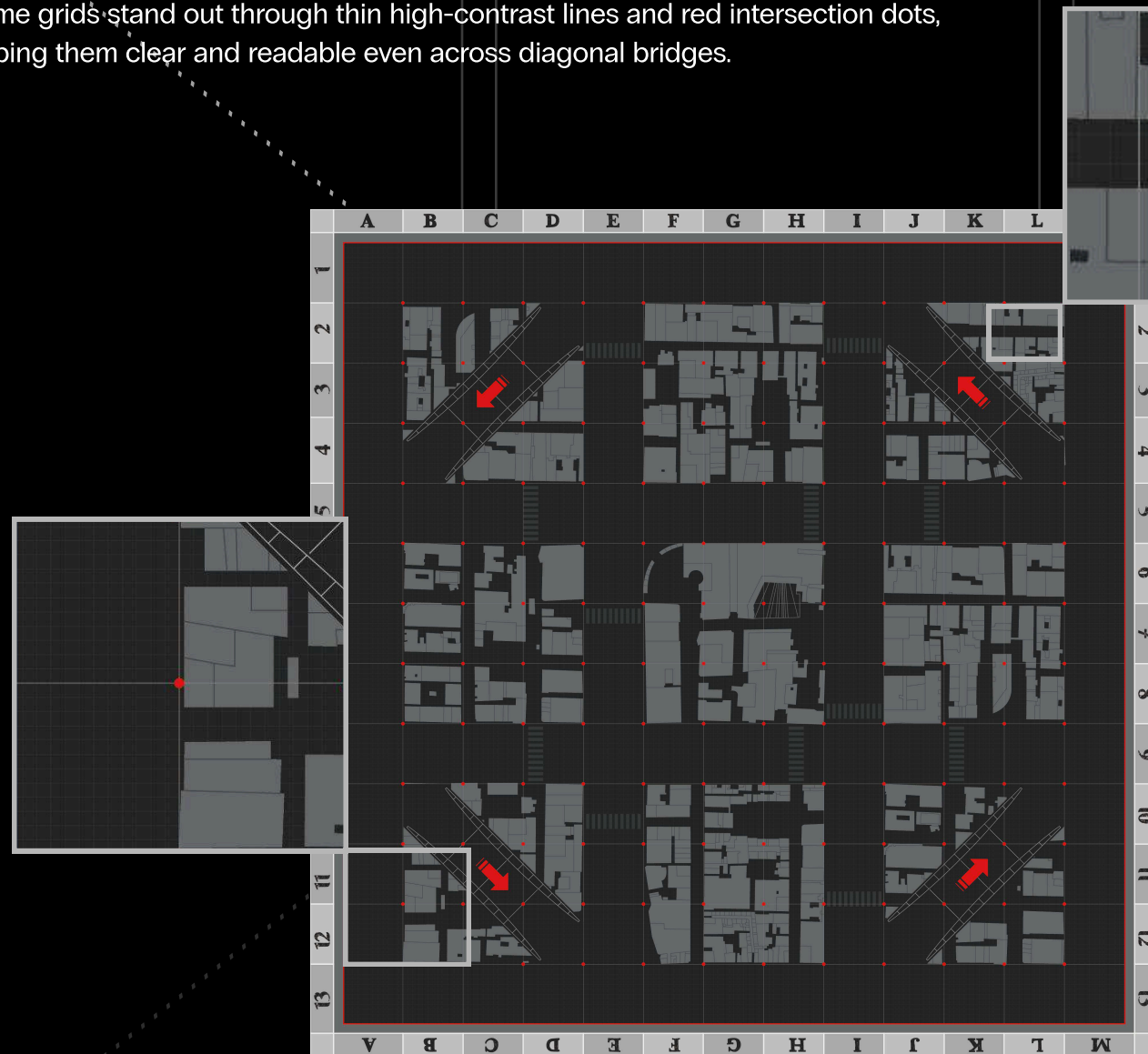
Blue: the police
Red: the protesters.



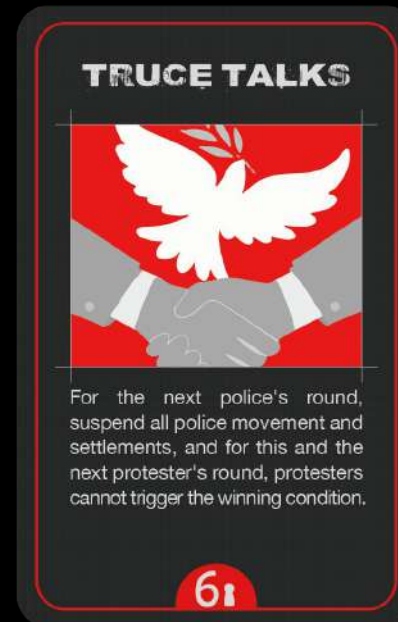
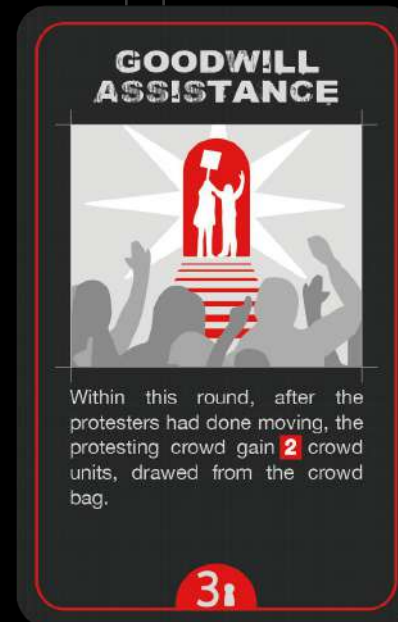
Accent Color 'Sniping'



In the map design, I used accent colors with dots, lines, and shapes.
Gray city clusters and dark roads form the background "shapes," serving only to show layout.
Game grids stand out through thin high-contrast lines and red intersection dots, keeping them clear and readable even across diagonal bridges.

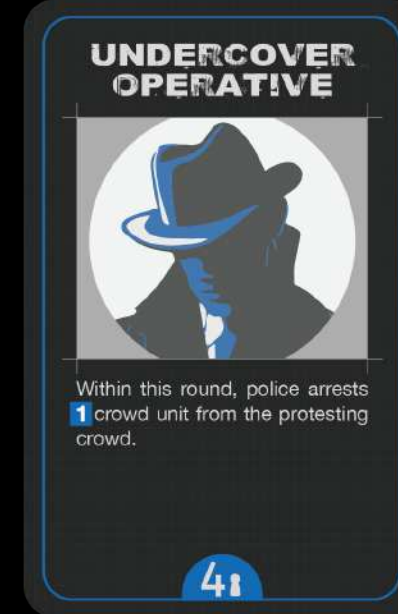


Dots, Lines, and Shapes

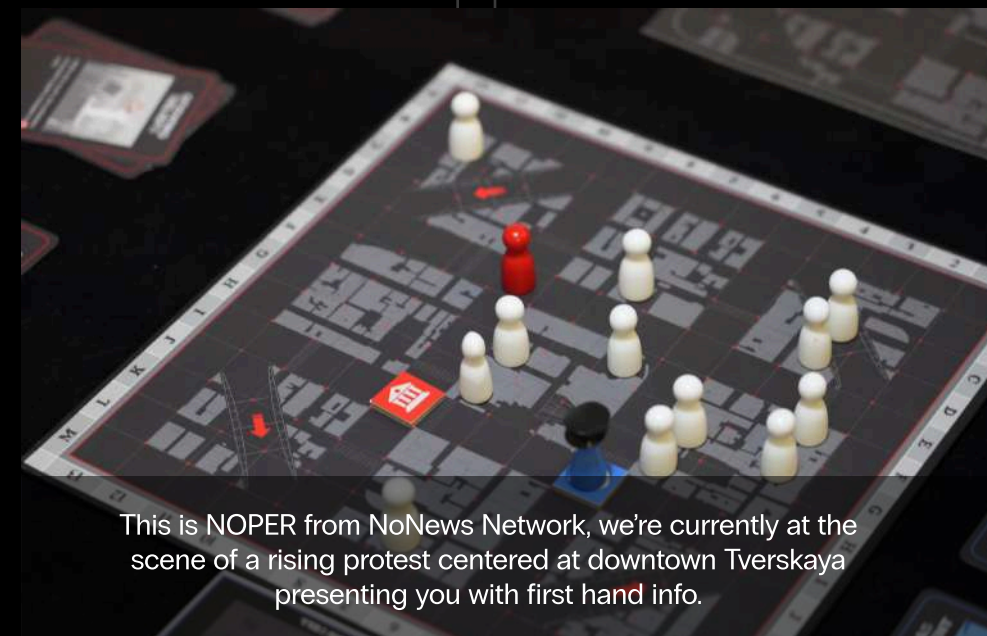


Protester Action Cards

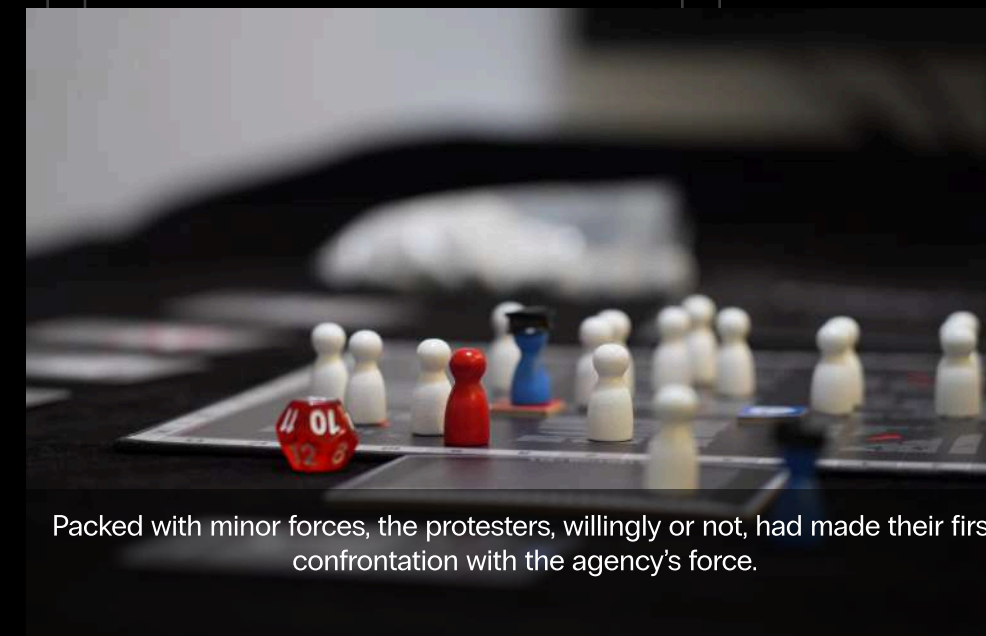
Police Action Cards



The Outcome&Gameplay Demonstration



This is NOPER from NoNews Network, we're currently at the scene of a rising protest centered at downtown Tverskaya presenting you with first hand info.



Packed with minor forces, the protesters, willingly or not, had made their first confrontation with the agency's force.



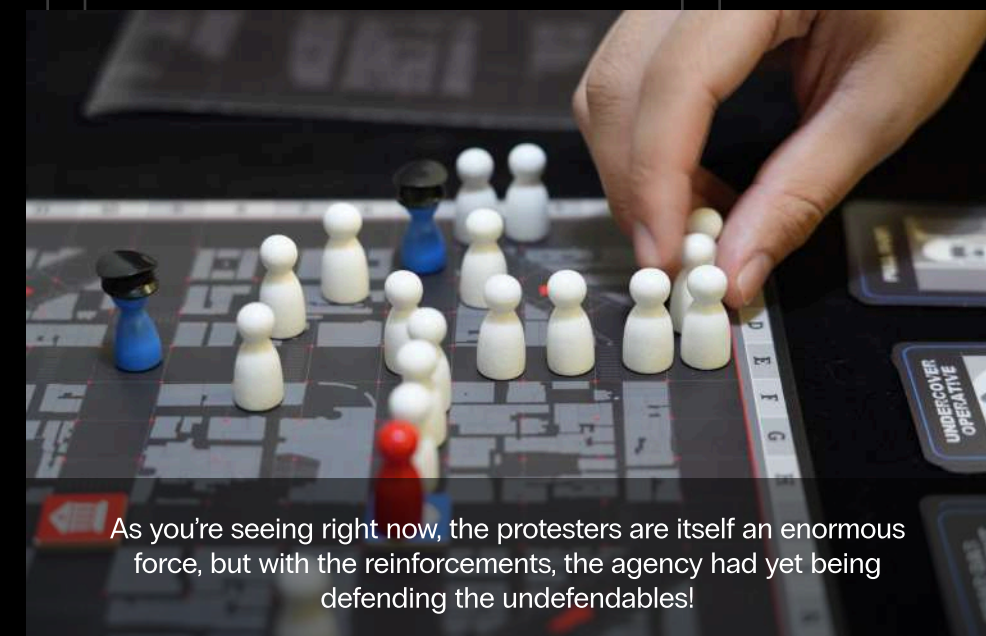
With the protesters hitting rocks with eggs, we have confirmed arrests by the police!



The situation is escalating rapidly, the protesters are drawing more following, where the police is making more and more arrests.



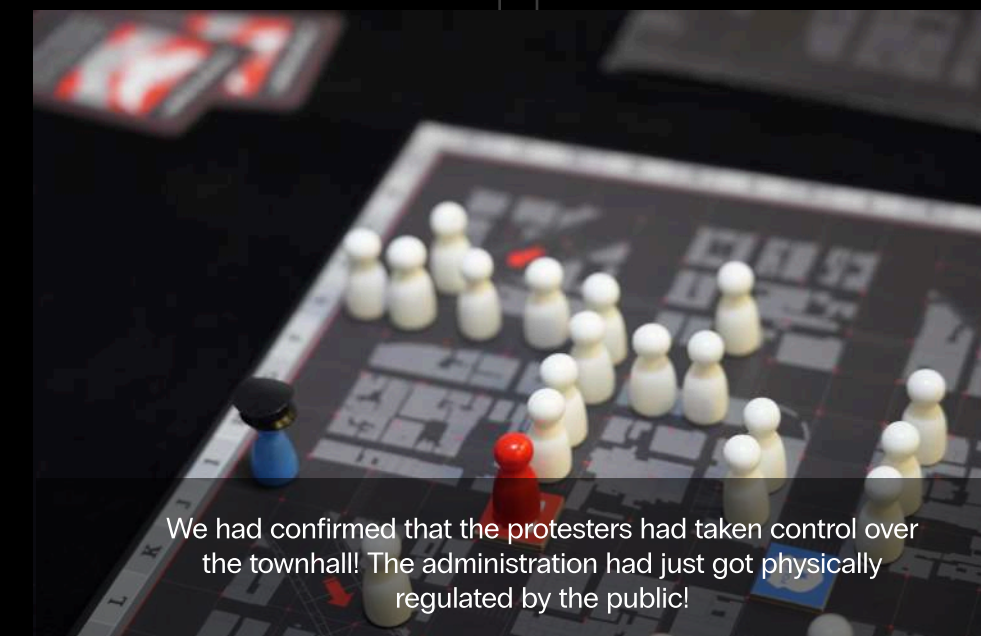
A informed source tells that in response to the massive number of protesters threatening charging towards the townhall, the agency had decided to reinforce the police with backup units!



As you're seeing right now, the protesters are itself an enormous force, but with the reinforcements, the agency had yet being defending the undefendables!



We just visually confirmed protesters bluntly pushing over barricades and burning road blocking police cars down! This protesting event had just became violent!



We had confirmed that the protesters had taken control over the townhall! The administration had just got physically regulated by the public!

WANTED YOU



WANTED

I WANTED YOU

The Conflict Within And Without,
The Fracture Between Nation And Self,
The Desire And The Ruin.

I WANTED YOU

I WANTED YOU

I WANTED YOU

Intro:

Between 1976 and 2021, the number of intrastate conflicts experienced a significant rise, reflecting increasing internal tensions within nations. Similarly, internationalized intrastate conflicts saw a notable increase, indicating a growing trend of external forces becoming involved in internal disputes.

In contrast, interstate conflicts, while fluctuating, did not exhibit the same upward trajectory, suggesting a relative stability in direct conflicts between nations. This period underscores the evolving nature of conflict, with internal and internationally influenced internal conflicts becoming more prevalent over time.

Project Info:

2024.07 | Individual Project | Unreal, Davinci Resolve, Figma

Thesis Proposal



A father holding up his healeson, which lost his life to bombardment

"I know not with what weapons World War III will be fought, but World War IV will be fought with sticks and stones."



When Images Speak the Truth:

During my time photographing protest events in Toronto against the humanitarian atrocities in Gaza,

I witnessed a profound and heart-wrenching display of human suffering. Among the most harrowing sights were the countless horrifying images depicting the casualties—innocent civilians and children, their lives abruptly and tragically cut short. These images, raw and unfiltered, brought the brutal reality of the conflict into sharp, unignorable focus.



Starting Points

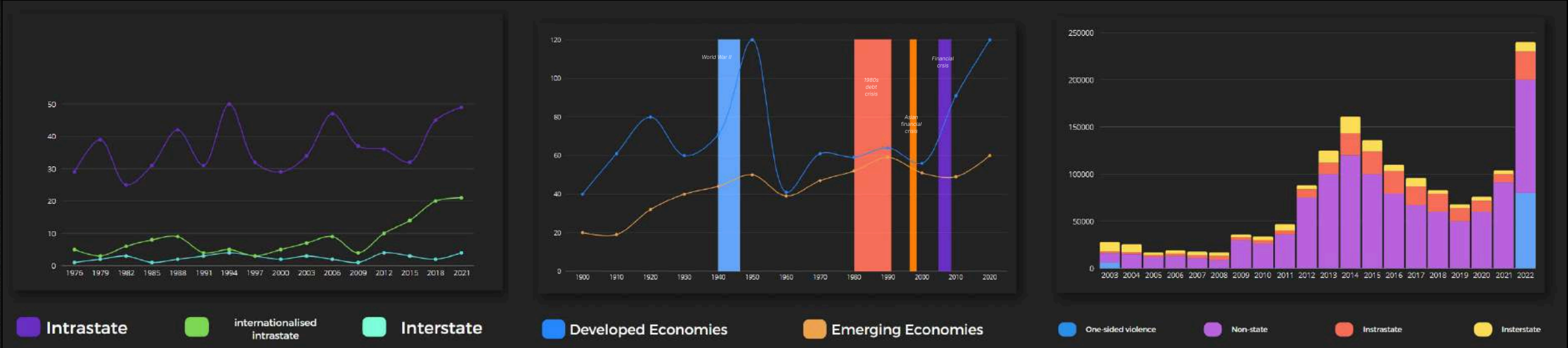
Albert Einstein Research

Global Conflict Trends, 1900–2022

From 1976 to 2021, intrastate and internationalized conflicts rose sharply, while interstate wars remained stable, showing a shift toward internalized warfare.

Across 1900–2020, global crises—from World War II to the 1980s debt and 1990s Asian financial crises—marked recurring waves of tension and instability.

Between 2003 and 2022, conflict deaths fluctuated, peaking in 2014–2015 and rising again in 2022, revealing the persistent volatility of global violence.



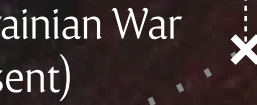
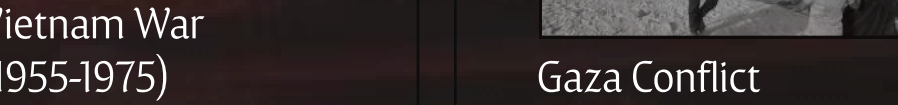
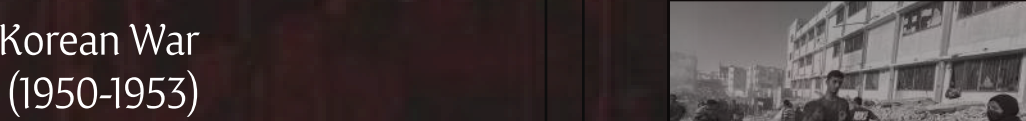
Timeline



Spanish Civil War (1936-1939)



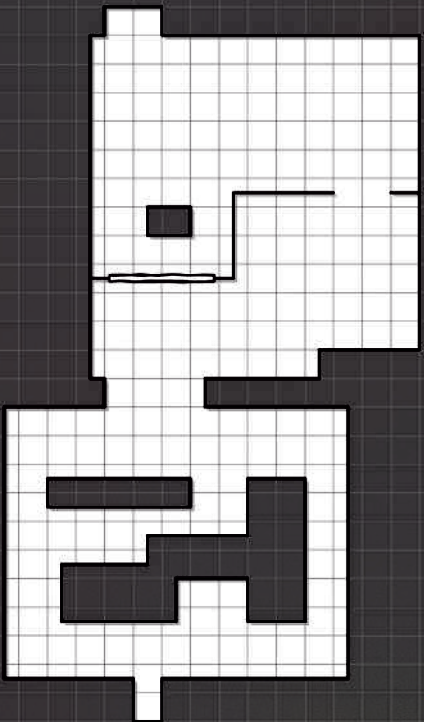
World War I (1914-1918)



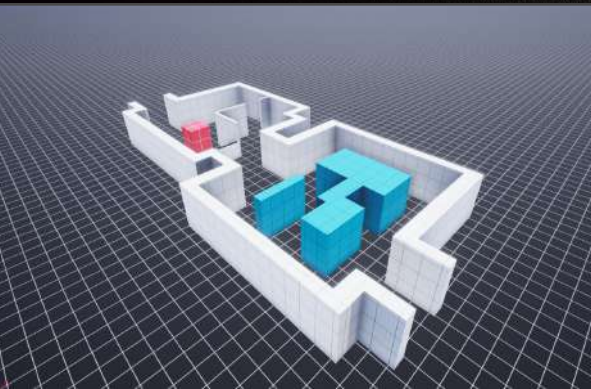
Global Conflict Indicator



Iteration



Map 1: Tutorial Version 1.0



Design Intention: A tutorial section, the player should learn the basics of their signal receiver

- Rules:**
- 1. Player can only see a small circular area when not using his signal receiver
 - 2. When the signal receiver is used, player can expand his vision and see the map and the hostiles nearby.
 - 3. Whenever the player see the hostile, the hostile also sees the player
 - 4. The hostile will linearly chase the last known player's location
 - 5. Anyone can see through but not go through glasses
 - 6. Hostile kills player when player can see hostile without signal receiver

Game flow:



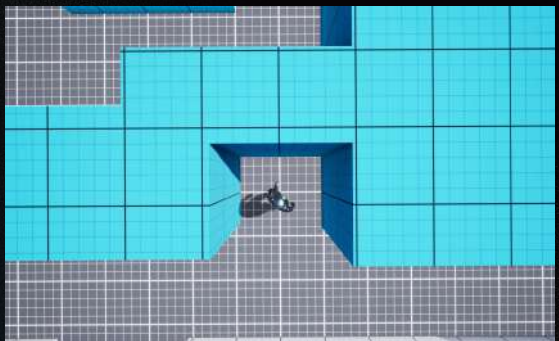
Hostile Behavior
Hostile tries to break glass, chases player's last known position.

Second Sector
Player exits maze, enters second sector, uses signal receiver to locate exit, sees hostile through glass.

Enter Maze
Player enters maze, uses signal receiver to locate path.



When the game start, the character will be at the entrance



Moving forward, player will encounter a maze

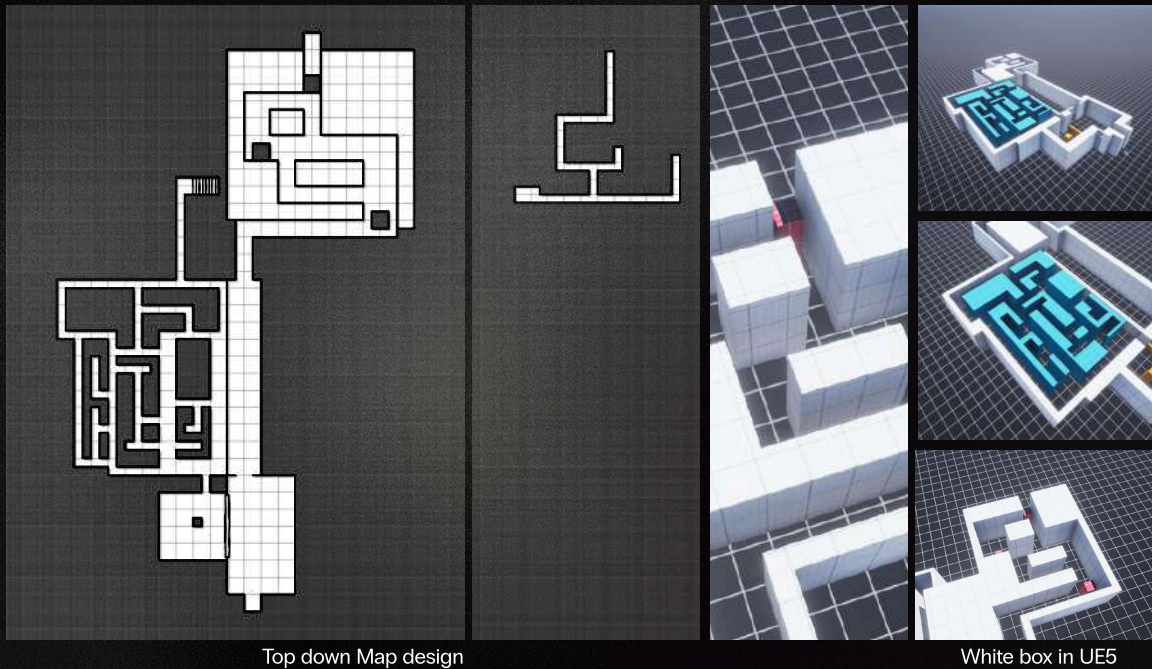


Exiting the maze, player will see the first enemy through glass



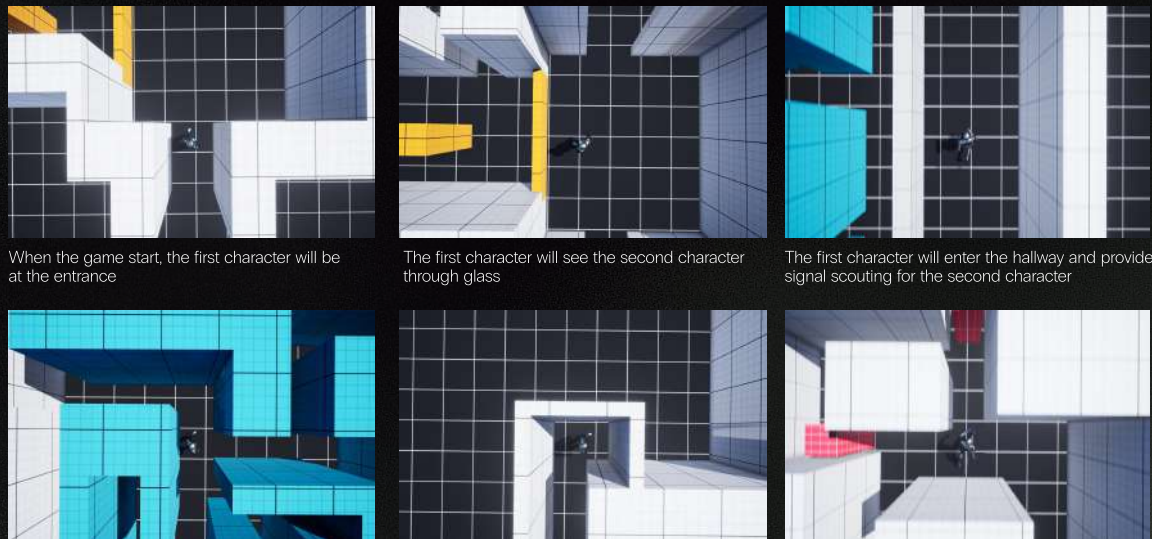
Distracting the enemy through their signal receiver, player will exit the level safely

Map 2: An Extra Hand (Version 1.1)



Top down Map design

White box in UE5



When the game start, the first character will be at the entrance

The first character will see the second character through glass

The first character will enter the hallway and provide signal scouting for the second character



Under guidance, the second character will move through the maze



Exiting the maze, the second character will go upstairs



The second character will then guide the first character through sectors filled with enemies

Rules:

- 1. Player can only see a small circular area when not using his signal receiver
- 2. When the signal receiver is used, player can expand his vision and see the map and the hostiles nearby.
- 3. Whenever the player see the hostile, the hostile also sees the player
- 4. The hostile will linearly chase the last known player's location
- 5. Anyone can see through but not go through glasses
- 6. Hostile kills player when player can see hostile without signal receiver
- 7. The second character can travel through passages that the first character could not
- 8. The second character's signal emitter cannot receive any information
- 9. The second character's signal emitter can still draw enemies to the character

Design Intention:

The section where the player should learn the properties and abilities of the second operate-able character

Game Flow

1

Encounter Second Character
Player meets a character behind glass.

4

Scout Maze
Player switches to scout character to map maze.

8

Exit Map **EXIT**
Player guides characters to exit the map.

2

Operate Second Character
Player learns to control the character.

3

Enter Maze
Player uses character to enter the maze.

5

Exit Maze
Player guides character out of the maze.

7

Distract Hostiles
Player emits signal to draw hostiles away.

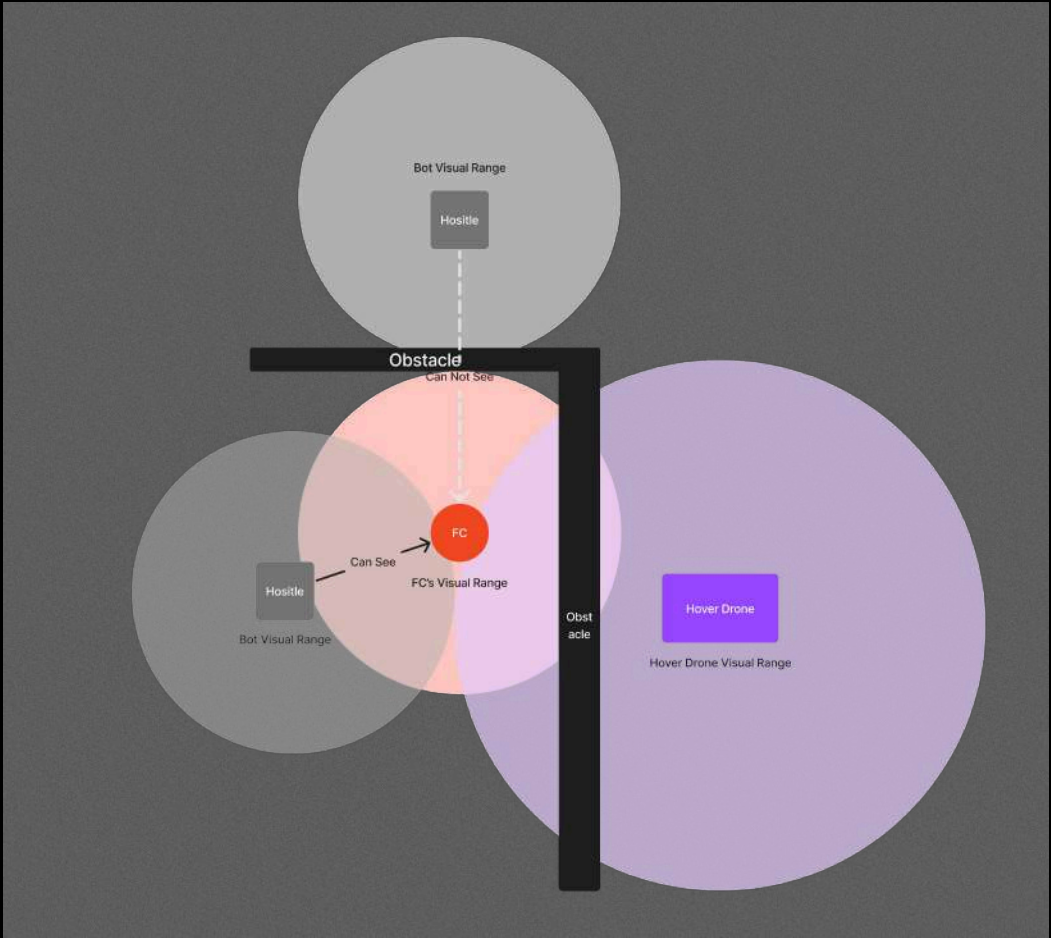
6

Use Emitter
Player uses emitter to see hostiles.

Core Mechanics

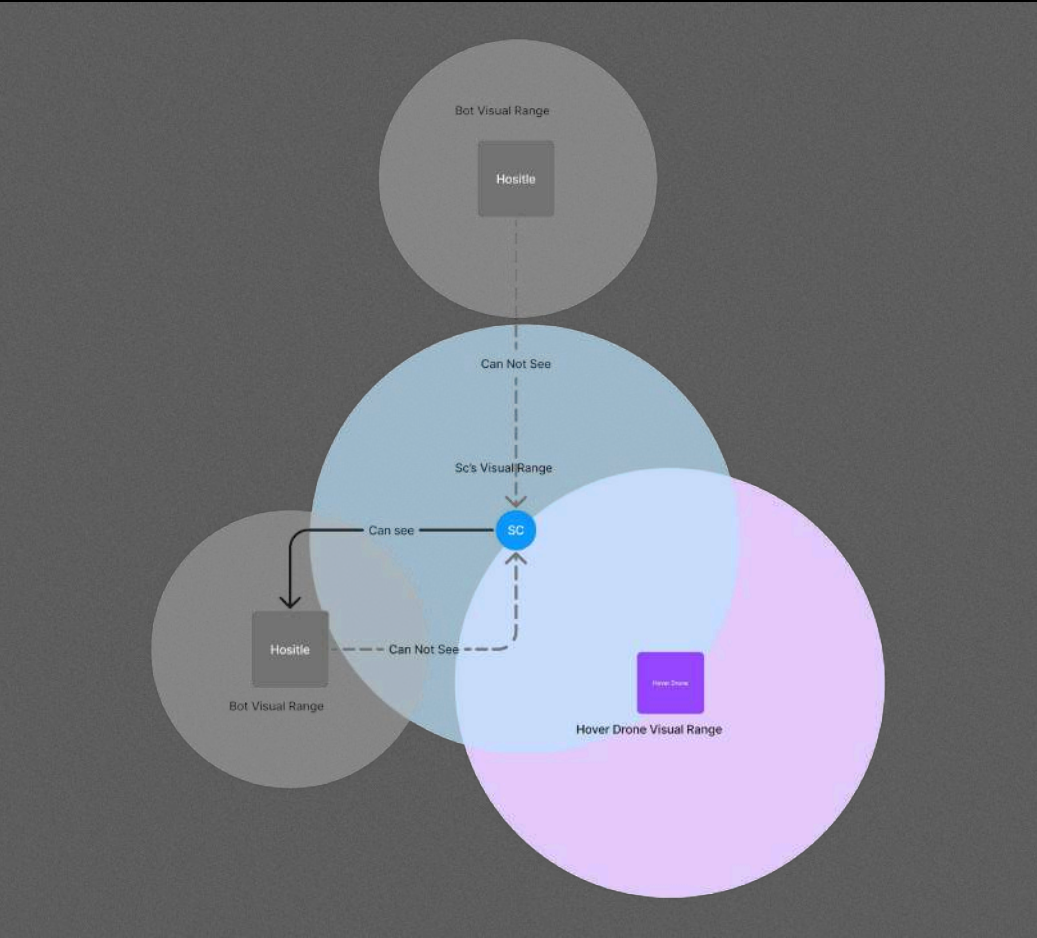
Level Design

FC's Visual Mechanics



Under such situation, only one hostile can see the first character, the other hostiles' vision is blocked by walls

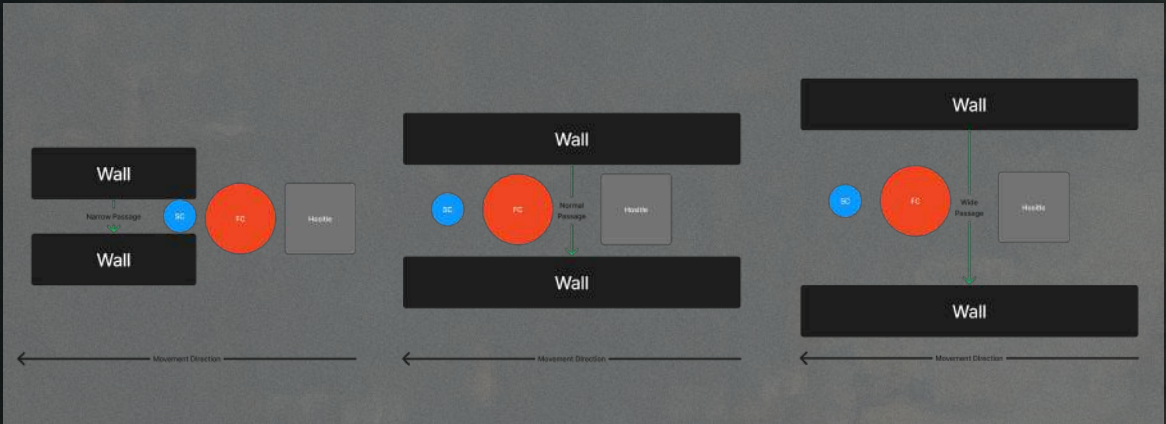
SC's Visual Mechanics



Under such situation, because the second character has a smaller body size, only hover drones can see the second character

Character's Body Size Mechanics

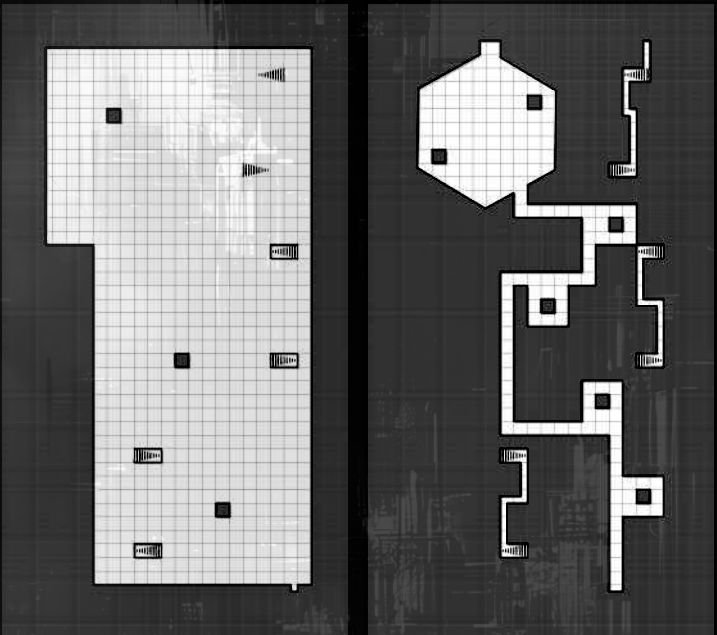
The illustrations shows different character's passability against different types of passages



Both Character's Signal Mechanics

The two illustration demonstrates the first character and the second character's signal emission/receive mechanics, interacting with different types of enemies

Map 3:

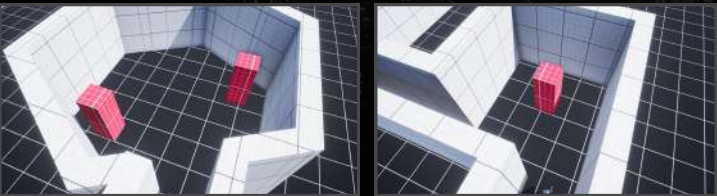
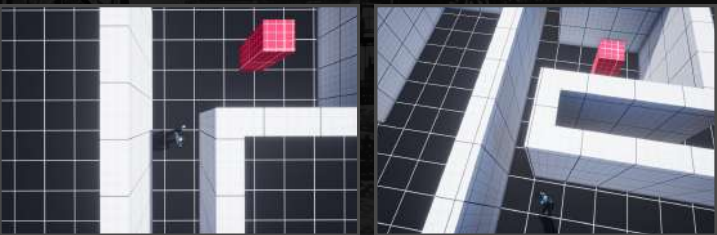
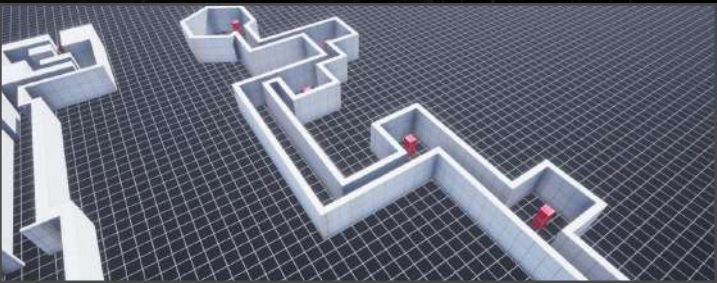


Rules:

- 1. All existing rules
- 2. All characters will also see everything inside hover drone's visual range.
- 3. Hover Drone can see through glass

Design Intention:

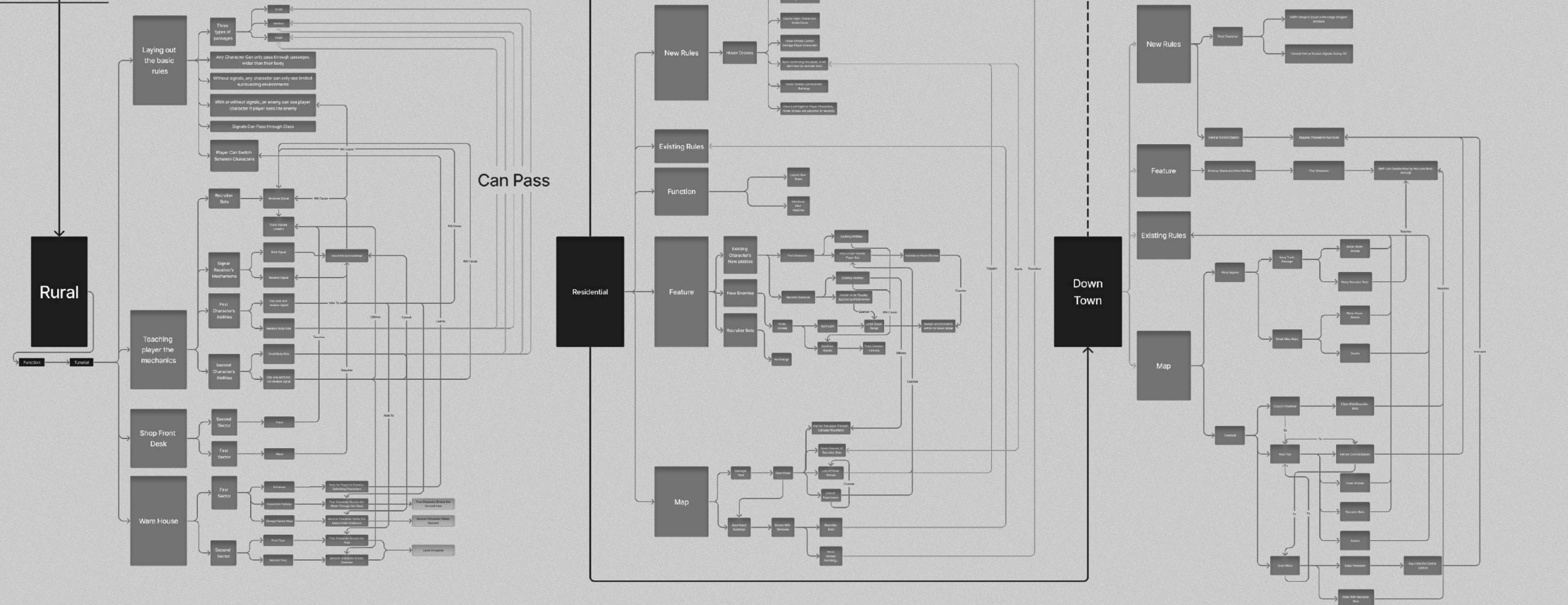
The final all-in-one map, contains all types of enemies and requires player to use all the character's skills



Game Flow



Flow chart



Future Goals

Desired Atmosphere



I aim to immerse players in a deeply unsettling post-apocalyptic world—tense, bleak, and haunting. Inspired by *Darkwood*, my goal is to evoke a constant sense of dread and consequence, drawing players into an experience that lingers long after the screen fades to black.

Desired Level Experience



To heighten the game's dread, I aim to instill a constant sense of stress and urgency. Like Resident Evil, progression will rely on solving intricate puzzles rather than combat, forcing players to think critically and strategically. This approach deepens tension and immersion, emphasizing psychological intensity over brute force.

Desired Gameplay Experience



The core reason behind creating a dreadful atmosphere and demanding levels is to deliver a challenge-driven experience. We want players to feel immense satisfaction and reward after overcoming each obstacle. Inspired by the Dark Souls series, known for its punishing difficulty and deeply gratifying victories, we aim to evoke that same sense of hard-earned triumph and fulfillment.

Potential Problems

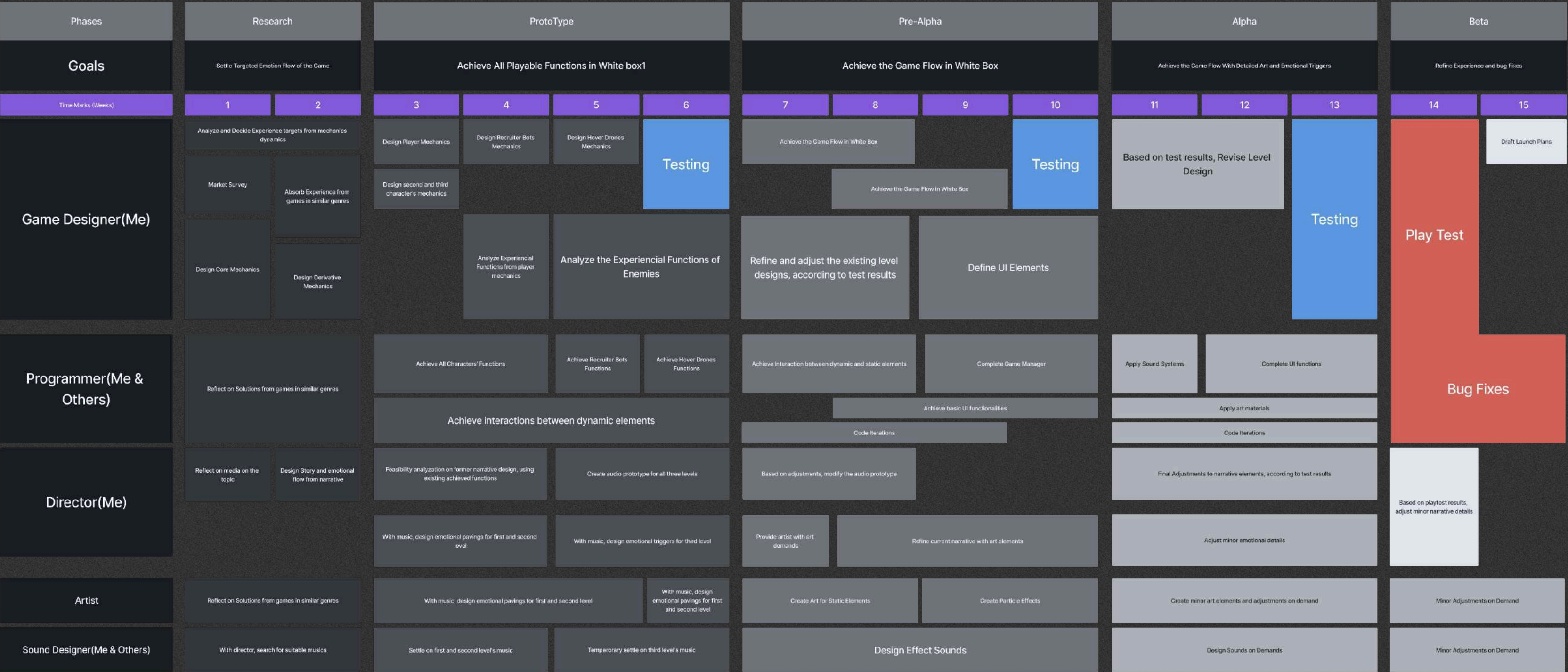
Intensive Art Demands:

Creating a daunting atmosphere poses a major challenge for our artist, requiring both skill and a deep passion for dark, immersive aesthetics. Beyond technical proficiency, genuine enthusiasm for this style is essential to convey the intense mood we envision, ensuring every visual element enhances the game's pervasive sense of dread and immersion.

Play-Testing Mayhem:

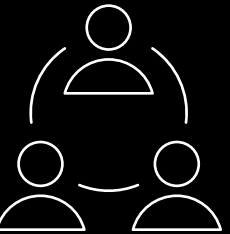
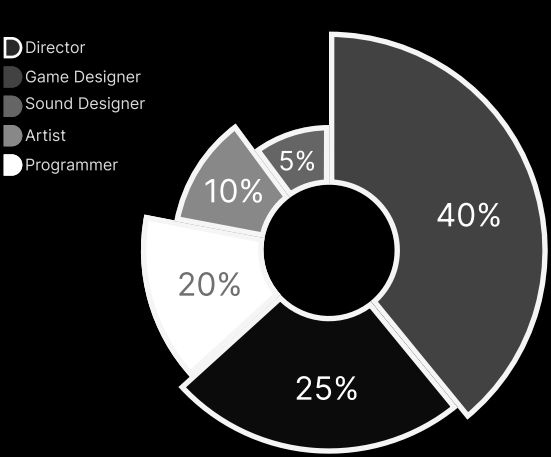
Designing puzzle-driven levels requires meticulous planning and thorough play-testing. However, repeated testing by the same players can lead to biased results, as prior knowledge influences their feedback. To ensure accurate and unbiased insights, a key challenge is recruiting new play-testers for each iteration, providing fresh perspectives with every evaluation.

Future Plan



Production Management

Research Phase:



Research Phase:

The game is conceptual; designers and the director define the vision.

Prototype & Pre-Alpha Phase:

Core mechanics are built. Programmers lead; artists and sound designers support.

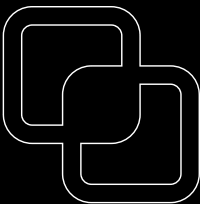
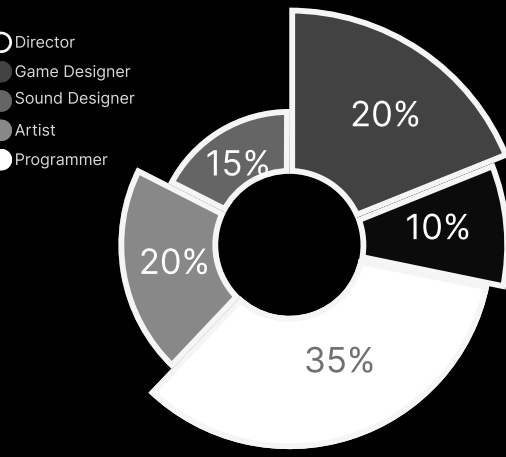
Alpha & Beta Phase:

Focus on polish and final integration. The team refines assets and gameplay.

Different Backgrounds:

As this game focuses on possible post apocalyptic events, one would adore team members from different regions of the world, bringing different cultural insights during the designing process.

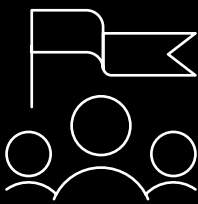
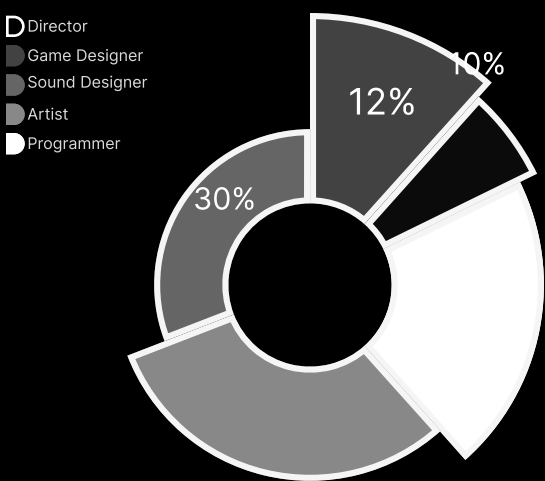
Prototype & Pre-Alpha Phase:



Different Perspectives:

One's experience and ideas relates significantly with ones wealth class, hence there should be members from all of upper, middle and the working class families.

Alpha & Beta Phase:



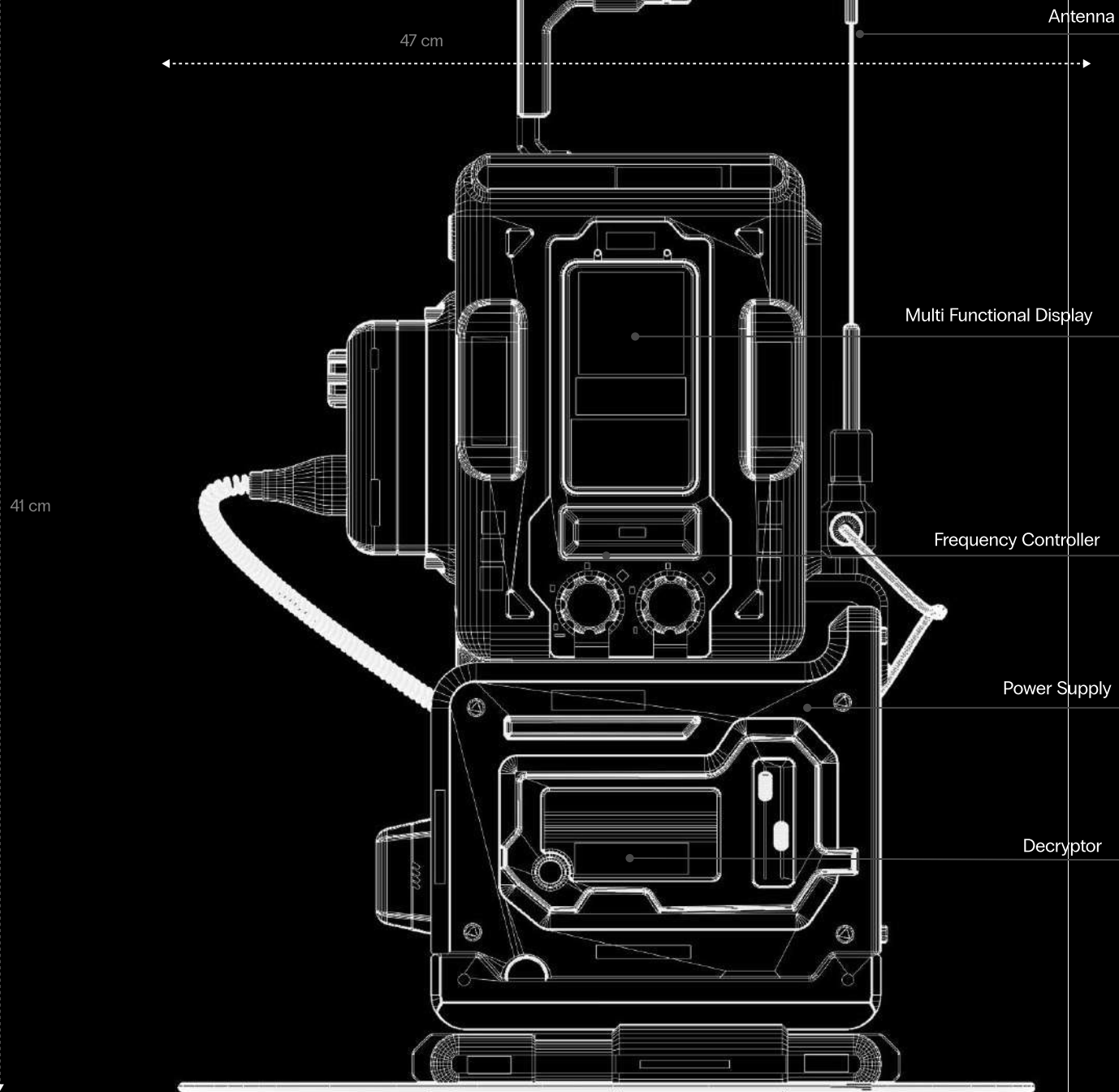
Different Passions:

With the broader passion on the project, each member should also be passionate on their roles and responsibilities.

Visual Design



Signal Receive / Emitter



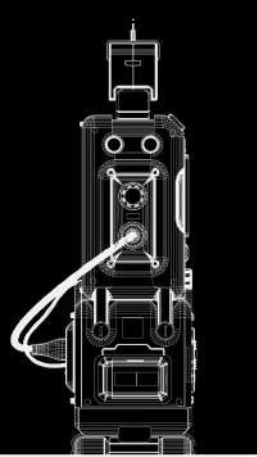
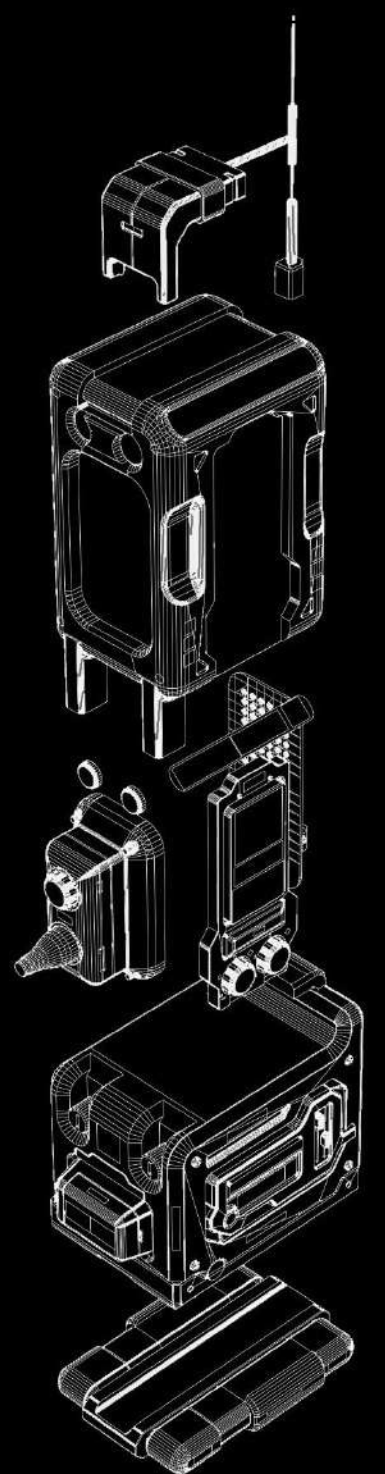
Antenna: Determines the range of possible signal transmission. On this particular model, the antenna length is fixed.

Casing: Contains all the processing units and the wires and connectors. Provides mounting for control panels

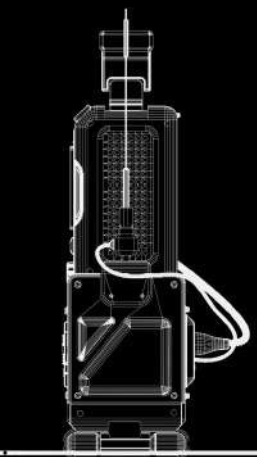
Frequency Selector: Panel that controls the frequency of emitted and received signal frequencies.

Lower Body: Container for decryptor and power supply

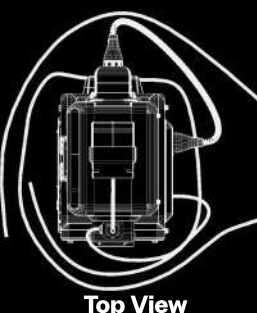
Base: The carrying panel that all devices mounts to



Front View



Left View



Top View

Development process

// Fill out your copyright notice in the Description page of Project Settings.

#pragma once

#include "GameFramework/Actor.h"
#include "ProFow.generated.h"

UCLASS()
class TTTT_API AProFow : public AActor
{
GENERATED_BODY()

public:
// Sets default values for this actor's properties
AProFow();

virtual void PostInitializeComponents() override;

// Set the fog size
void setSize(float sz);

// Reveal a portion of the fog
void revealSmoothCircle(const FVector2D &pos, float radius);

protected:
// Called when the game starts or when spawned
virtual void BeginPlay() override;

public:
// Called every frame
virtual void Tick(float DeltaTime) override;

void UpdateTextureRegions(UTexture2D* Texture, int32 MipIndex, uint32 NumRegions, FUpdateTextureRegion2D* Regions, uint32 SrcPitch, uint32 SrcBpp, uint8* SrcData, bool bFreeData);

// Fow texture size
static const int m_textureSize = 512;

UPROPERTY()
UStaticMeshComponent* m_squarePlane;
//TSubobjectPtr<UStaticMeshComponent> m_squarePlane;
UPROPERTY(EditAnywhere, Category = "ProFow")
UTexture2D* m_dynamicTexture;
UPROPERTY()
UMaterialInterface* m_dynamicMaterial;
UPROPERTY()
UMaterialInstanceDynamic* m_dynamicMaterialInstance;
uint8 m_pixelArray[m_textureSize * m_textureSize];
FUpdateTextureRegion2D m_wholeTextureRegion;
float m_coverSize;
UPROPERTY(EditAnywhere, Category = "ProFow")
APawn* myPawn;
UPROPERTY(EditAnywhere, Category = "ProFow")
bool bIsMMOBA = false;
};

// Fill out your copyright notice in the Description page of Project Settings.

#include "TTTT.h"
#include "ProFow.h"

// Sets default values
AProFow::AProFow() : m_wholeTextureRegion(0, 0, 0, m_textureSize, m_textureSize)

// Set this actor to call Tick() every frame. You can turn this off to improve performance if you don't need it.
PrimaryActorTick.bCanEverTick = true;

m_coverSize = 5000;

// Set this actor to call Tick() every frame. You can turn this off to improve performance if you don't need it.
PrimaryActorTick.bCanEverTick = true;

// Create a planar mesh from engine's planar static mesh
m_squarePlane = CreateDefaultSubobject<UStaticMeshComponent>(TEXT("Fow Plane Static Mesh"));
RootComponent = m_squarePlane;
m_squarePlane->SetCollisionProfileName(UCollisionProfile::NoCollision_ProfileName);

static ConstructorHelpers::FObjectFinder<UStaticMesh> asset(TEXT("StaticMesh/Engine/ArtTools/RenderToTexture/Meshes/S_1_Unit_Plane_S_1_Unit_Plane"));
m_squarePlane->SetStaticMesh(asset.Object);

m_squarePlane->TranslucencySortPriority = 100;
m_squarePlane->SetRelativeScale3D(FVector(m_coverSize, m_coverSize, 1));

// Load the base material from your created material
static ConstructorHelpers::FObjectFinder<UMaterial> asset(TEXT("Material/Game/Fow/FowMat.FowMat"));
m_dynamicMaterial = asset.Object;

// Create the runtime FOW texture
if (!m_dynamicTexture)

{
m_dynamicTexture = UTexture2D::CreateTransient(m_textureSize, m_textureSize, PF_EBGR);
m_dynamicTexture->CompressionSettings = TextureCompressionSettings::TC_GrayScale;
m_dynamicTexture->SRGB = 0;
m_dynamicTexture->UpdateResource();
m_dynamicTexture->MapGenSettings = TMGS_NoMipmaps;
}

// Initialise array to all black
for (int x = 0; x < m_textureSize; ++x)
for (int y = 0; y < m_textureSize; ++y)
m_pixelArray[y * m_textureSize + x] = 255;

// Propagate memory's array to texture
if (m_dynamicTexture)
UpdateTextureRegions(m_dynamicTexture, 0, 1, &m_wholeTextureRegion, m_textureSize, 1, m_pixelArray, false);

void AProFow::PostInitializeComponents()

Super::PostInitializeComponents();

// Create a dynamic material instance to swap in the FOW texture
if (!m_dynamicMaterialInstance)
{
m_dynamicMaterialInstance = UMaterialInstanceDynamic::Create(m_dynamicMaterial, this);
m_dynamicMaterialInstance->SetTextureParameterValue("FowTexture", m_dynamicTexture);
}

// Set the dynamic material to the mesh
if (m_dynamicMaterialInstance)
m_squarePlane->SetMaterial(0, m_dynamicMaterialInstance);

void AProFow::setSize(float sz)

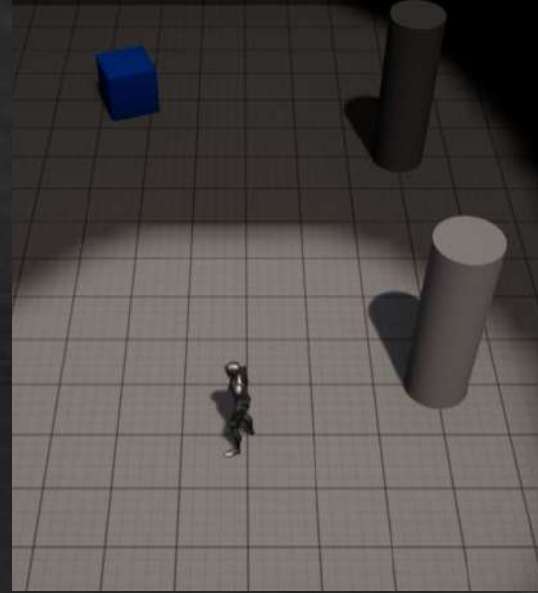
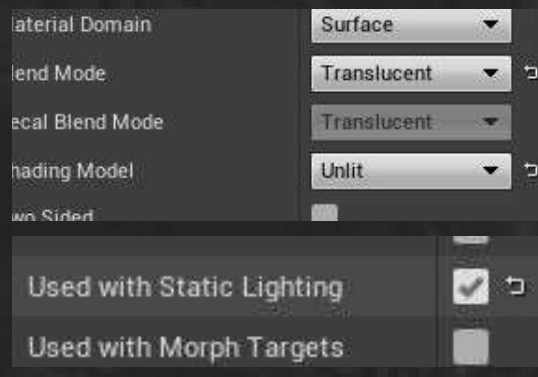
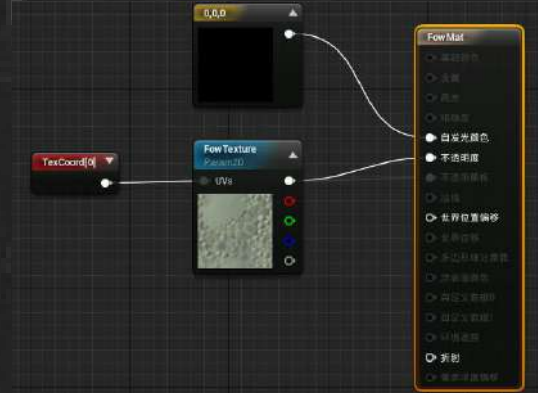
m_coverSize = sz;
m_squarePlane->SetRelativeScale3D(FVector(m_coverSize, m_coverSize, 1));

void AProFow::revealSmoothCircle(const FVector2D &pos, float radius)

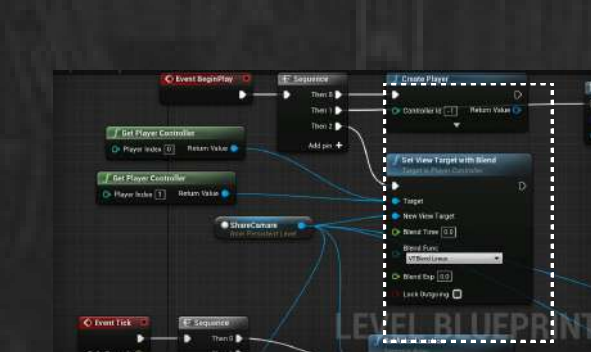
// Calculate the where circle center is inside texture space
FVector2D level = pos - FVector2D(GetActorLocation().X, GetActorLocation().Y);
level = level * m_textureSize / m_coverSize;
level += FVector2D(m_textureSize / 2, m_textureSize / 2);
// Calculate radius in texel unit (1 is 1 pixel)
float texelRadius = radius * m_textureSize / m_coverSize;

// The square area to update
int minX = FMath::ClampToInt(level.X - texelRadius, 0, m_textureSize - 1);
int minX2 = FMath::ClampToInt(level.X - texelRadius, 0, m_textureSize - 1);
int maxX = FMath::ClampToInt(level.X + texelRadius, 0, m_textureSize - 1);
int maxX2 = FMath::ClampToInt(level.X + texelRadius, 0, m_textureSize - 1);
int minY = FMath::ClampToInt(level.Y - texelRadius, 0, m_textureSize - 1);
int minY2 = FMath::ClampToInt(level.Y - texelRadius, 0, m_textureSize - 1);
int maxY = FMath::ClampToInt(level.Y + texelRadius, 0, m_textureSize - 1);
int maxY2 = FMath::ClampToInt(level.Y + texelRadius, 0, m_textureSize - 1);
uint8 theVal = 0;

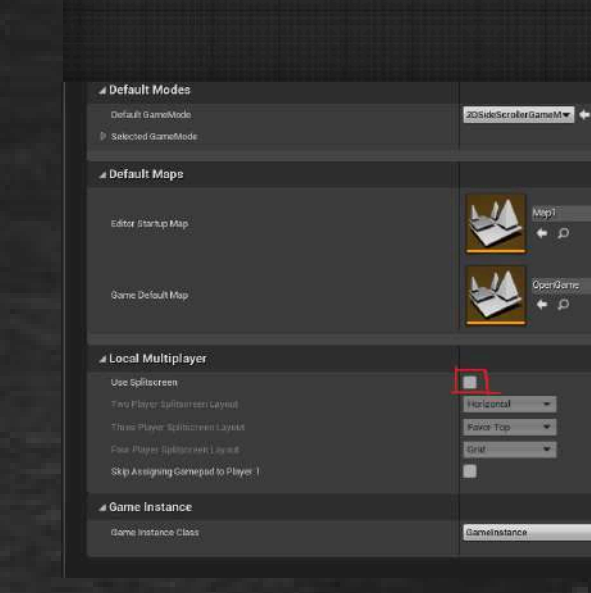
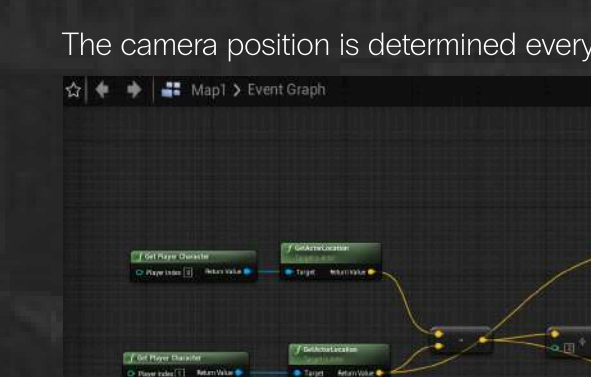
How to build fog of war



Set the camera position in the level blueprint



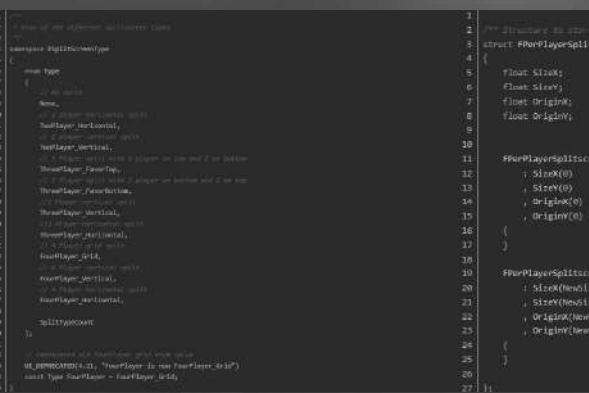
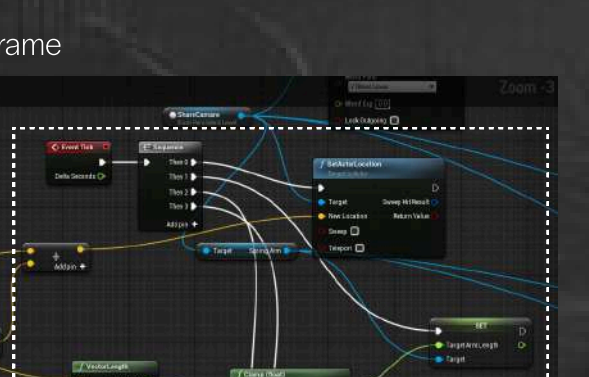
Start by creating an actor class called share camera that shares cameras



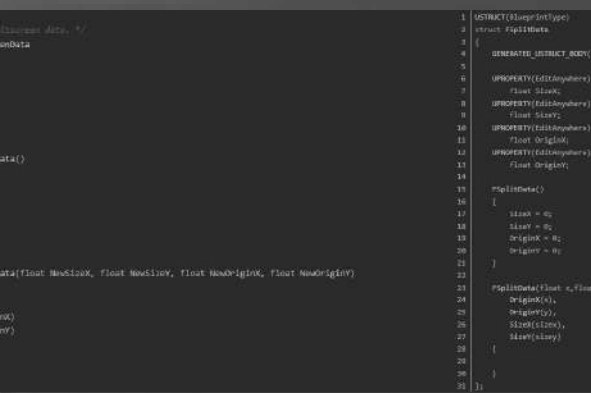
Set the camera position in the level blueprint



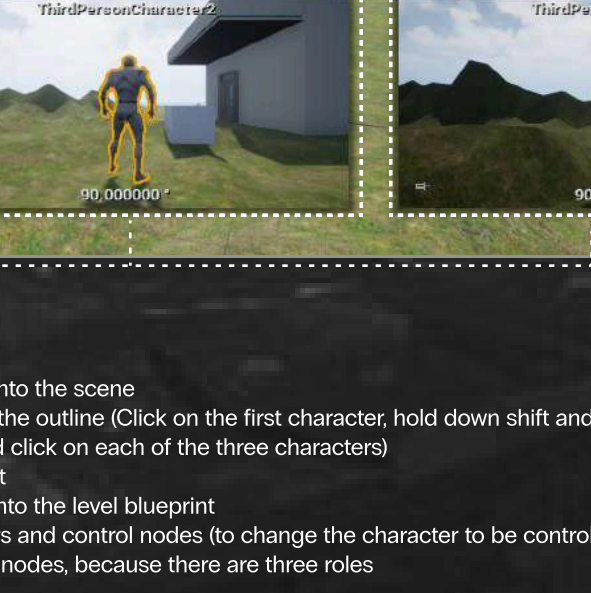
Start by creating an actor class called share camera that shares cameras



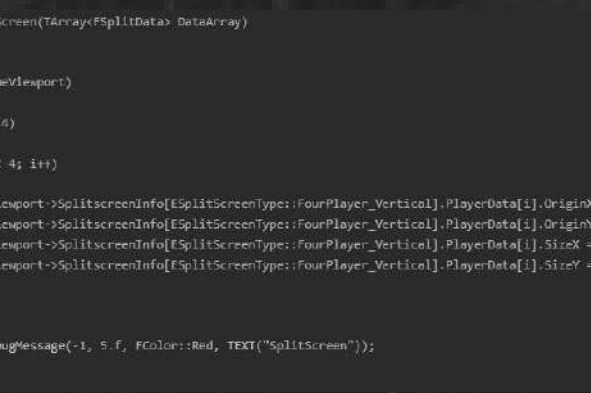
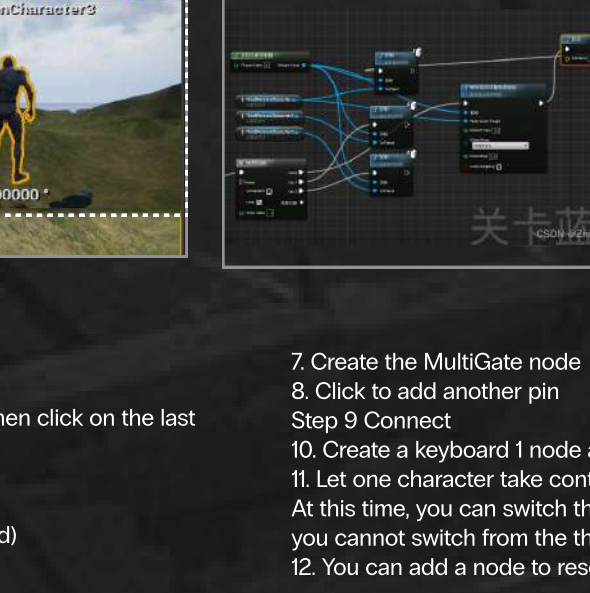
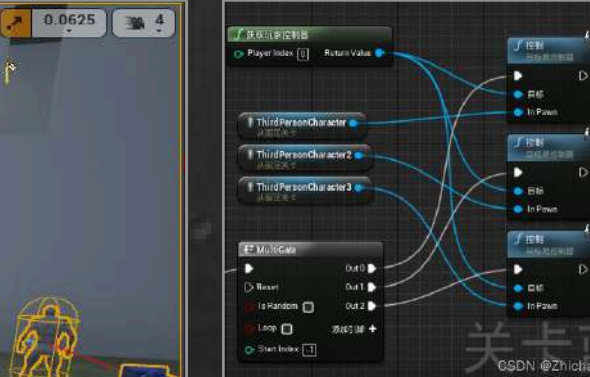
Set the camera position in the level blueprint



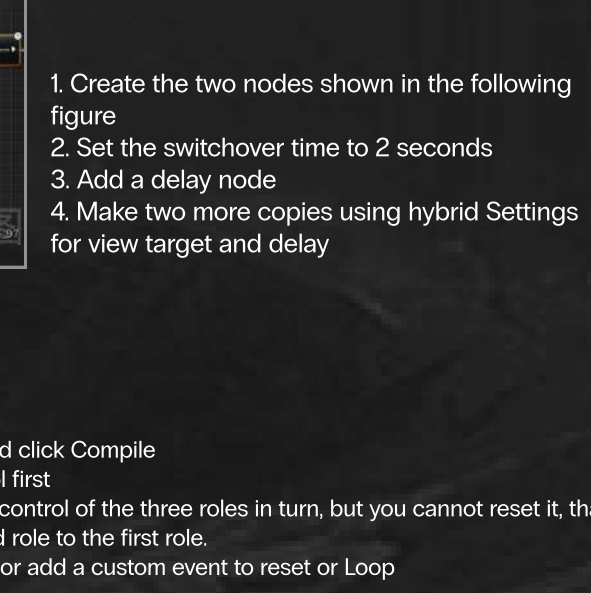
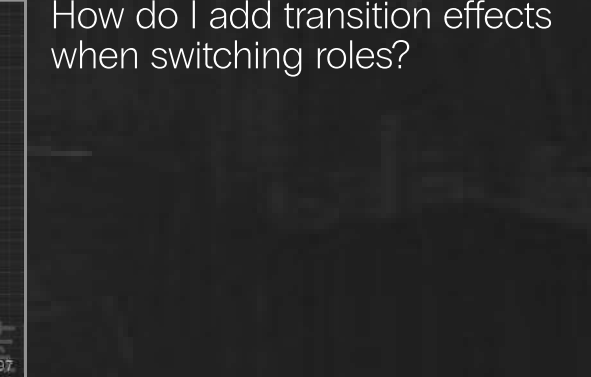
Set the camera position in the level blueprint



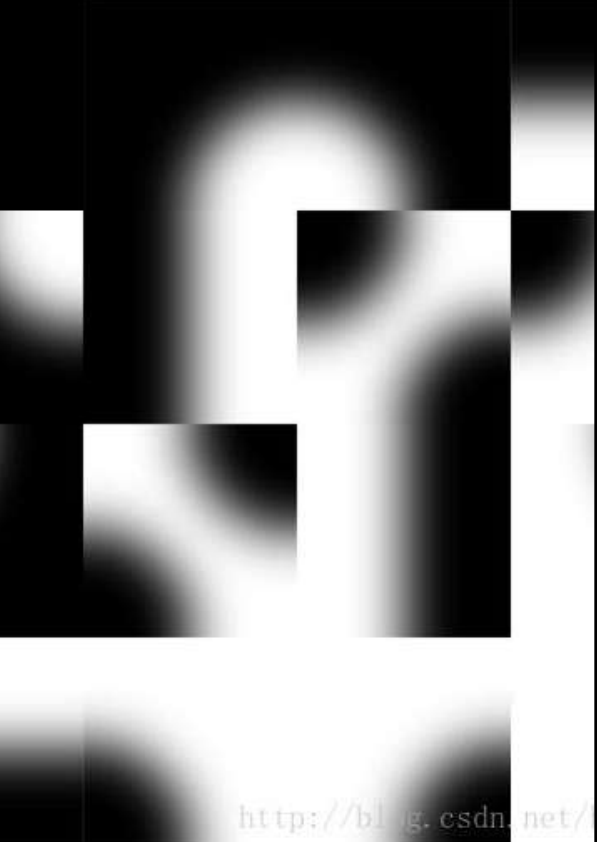
Set the camera position in the level blueprint



Set the camera position in the level blueprint



Add in build.cs
Add an actor called ProFow



How do I add transition effects when switching roles?

1. Create the two nodes shown in the following figure
2. Set the switchover time to 2 seconds
3. Add a delay node
4. Make two more copies using hybrid Settings for view target and delay

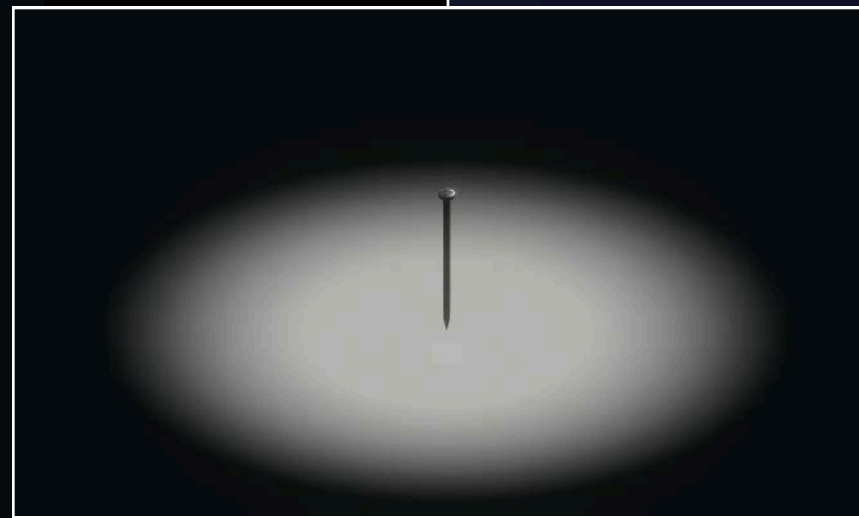
7. Create the MultiGate node
8. Click to add another pin
- Step 9 Connect
10. Create a keyboard 1 node and click Compile
11. Let one character take control first
- At this time, you can switch the control of the three roles in turn, but you cannot reset it, that is, you cannot switch from the third role to the first role.
12. You can add a node to reset or add a custom event to reset or Loop

Fourth of July

A VR experiential game dedicated to recreate the symptoms and experiences of veterans with PTSD.

PREFACE

In the stories of war, we often remember those actions of soldiers on the battlefield, but we forget that their fight doesn't end when they come home. Veterans face many challenges, from physical injuries to mental health struggles, and they need our support just as much after their service as they did during it. Ignoring their care is like leaving a job half done.



It's important that we remember their sacrifices and make sure they get the help they need to live healthy, fulfilling lives after their service. Taking care of veterans is just as crucial as any part of the war itself.

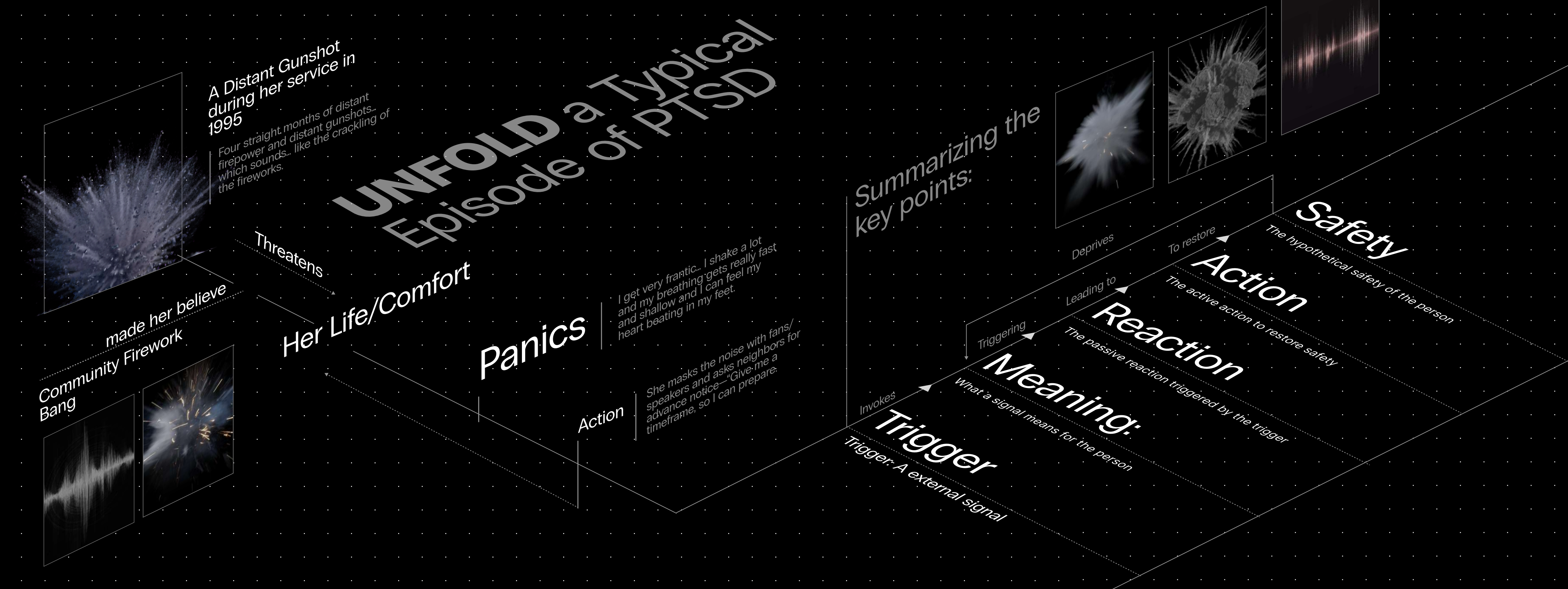
Analysis of PTSD Episode

What triggers the intense emotional reactions?
How does the trigger invokes such intense emotions?
To answer that, I had to verify these elements and study the underlying reality.

Case Study: Air Force veteran KJ Becke



Air Force veteran KJ Becker of Fowler, Colorado, links fireworks to past service trauma—sonic booms in 1995 and months of gunfire in Saudi Arabia. Unpredictable neighborhood fireworks trigger panic, rapid breathing, and shaking. To cope, she masks noise with fans and speakers and asks for advance notice to prepare.

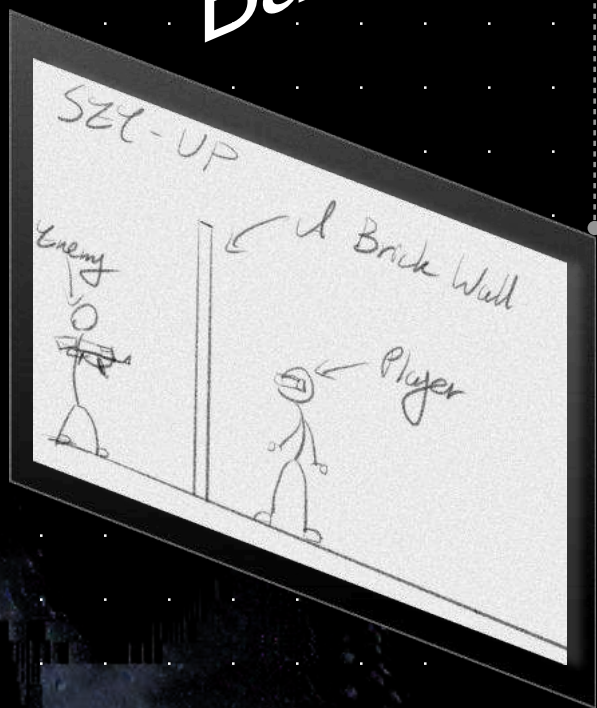


Reconstruction

We need to construct an environment that conveys the structure

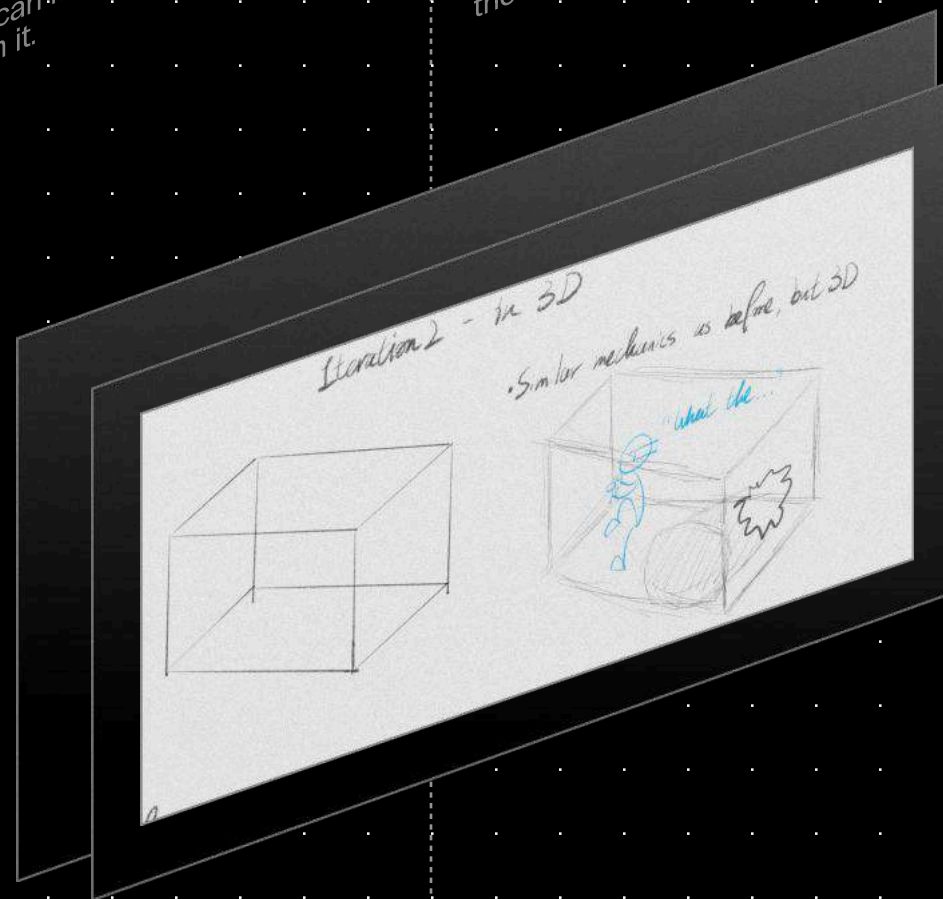
Baseline

Player is constrained within a room, where they can maneuver within it.



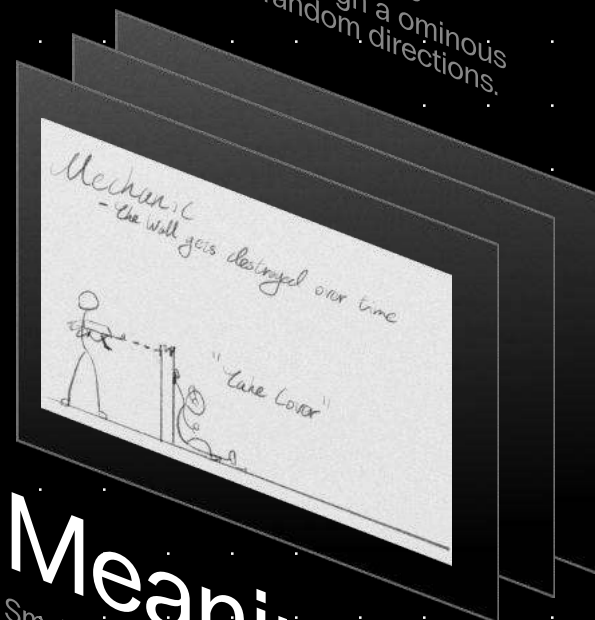
Safety

When wall breached, red volumetric light beams shines into the room, when the player touches the beams, they take damage.



Trigger

Similar to a fireworks Bang, we signals the trigger through a ominous hum coming from random directions.

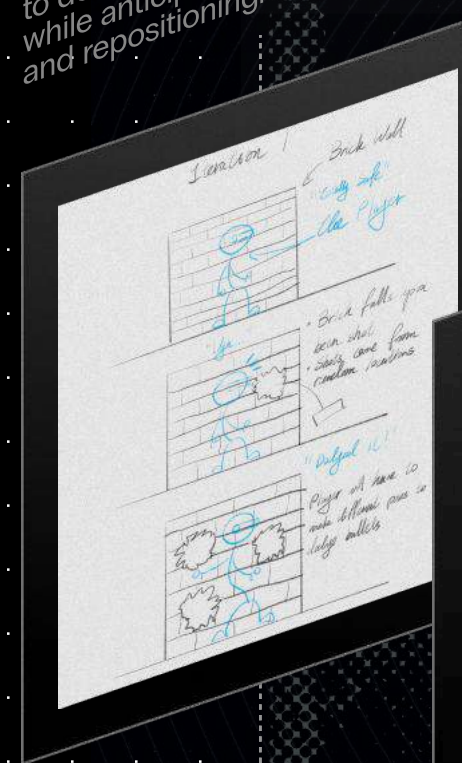


Meaning

Small period of time after the hum, a cannon shot from the direction of the hum hits the wall and cause certain damage (breach) to it. The breached wall falls and becomes debris.

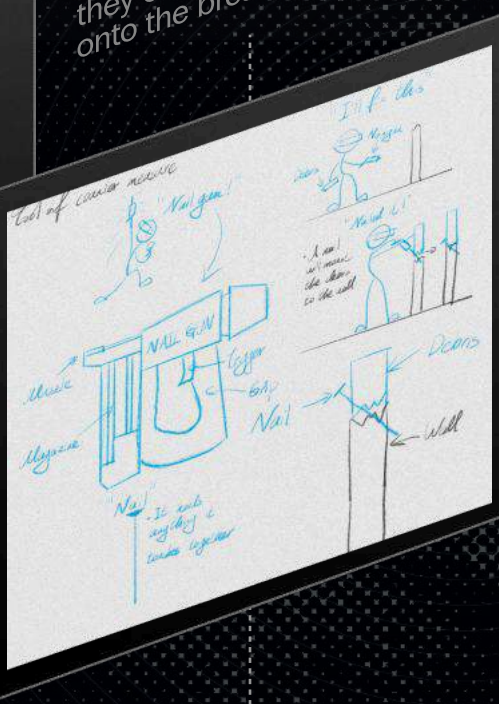
Reaction

Given the mechanics, we believe that the player will take all sorts of stands to dodge the existing red lights, while anticipating the next attack and repositioning.

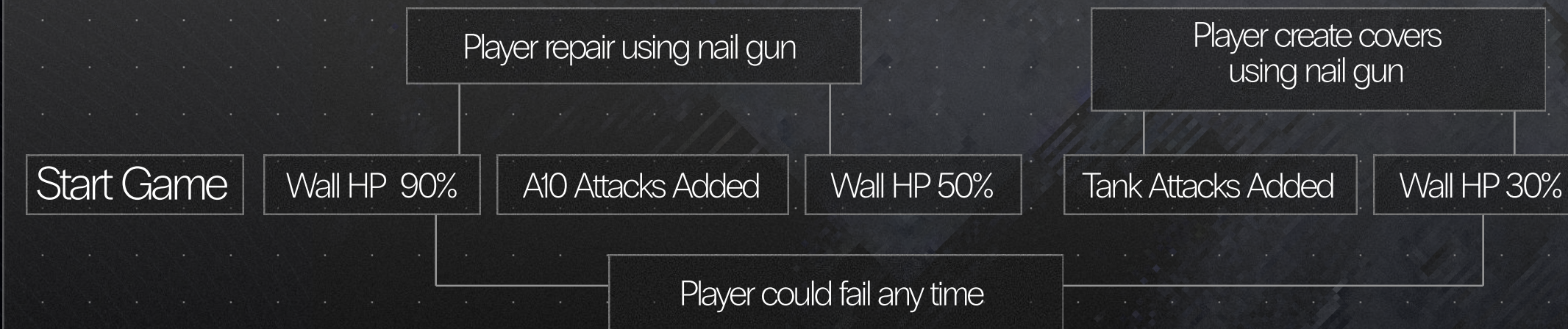


Action

the next attack and repositioning. Player will be given a nail gun, which they can take a debris, and nail it onto the breached walls.



Game Flow



Controller

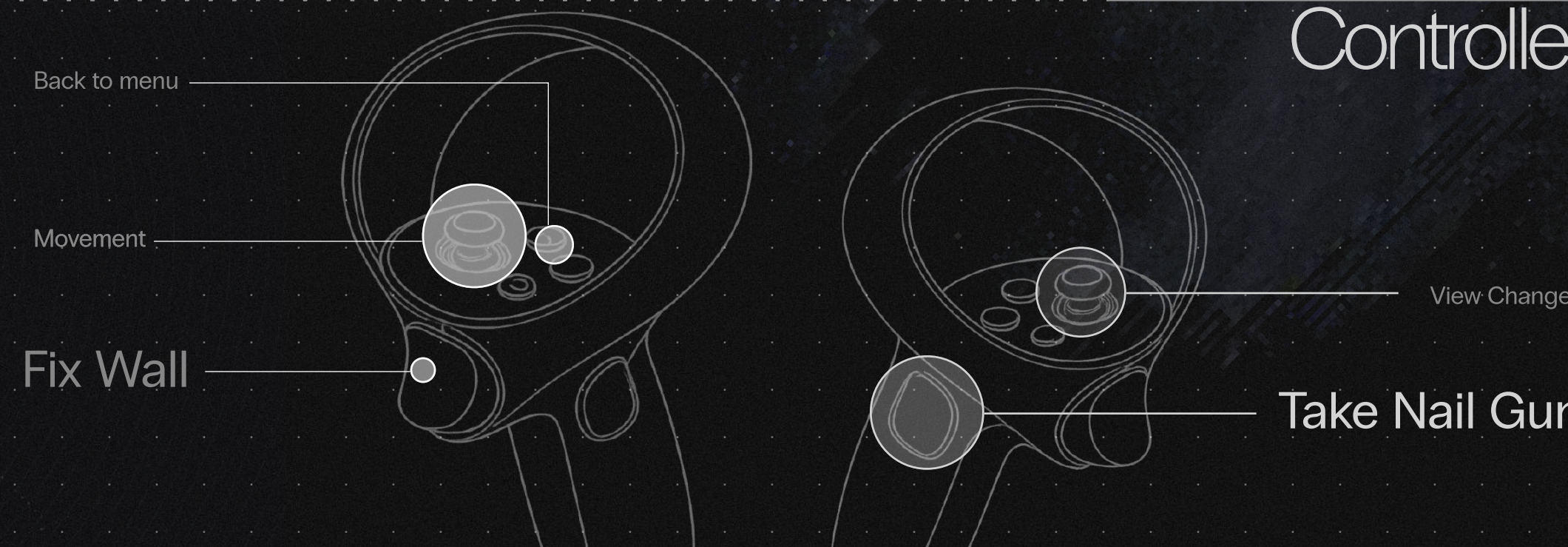
Back to menu

Movement

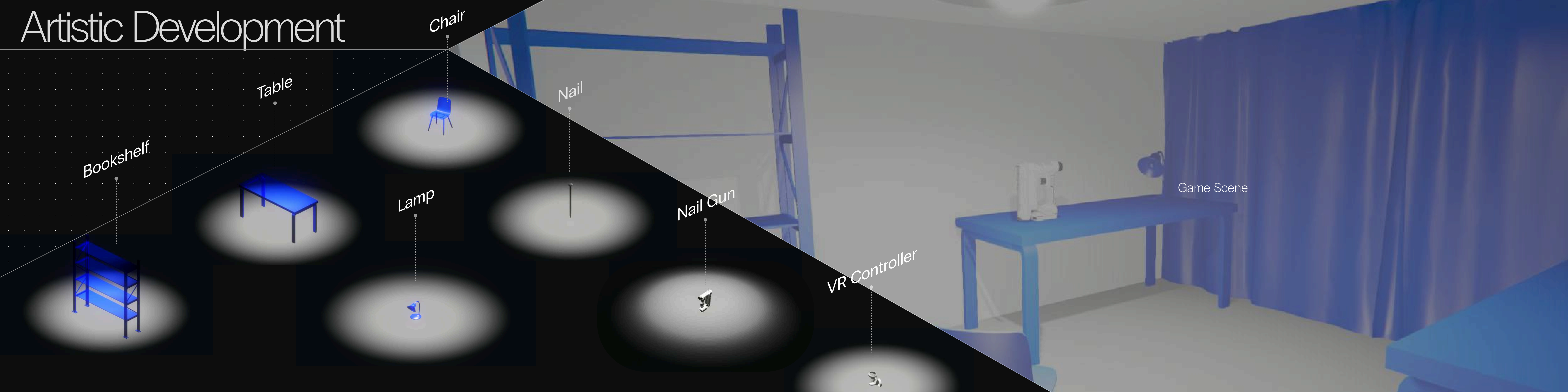
Fix Wall

View Change

Take Nail Gun



Artistic Development

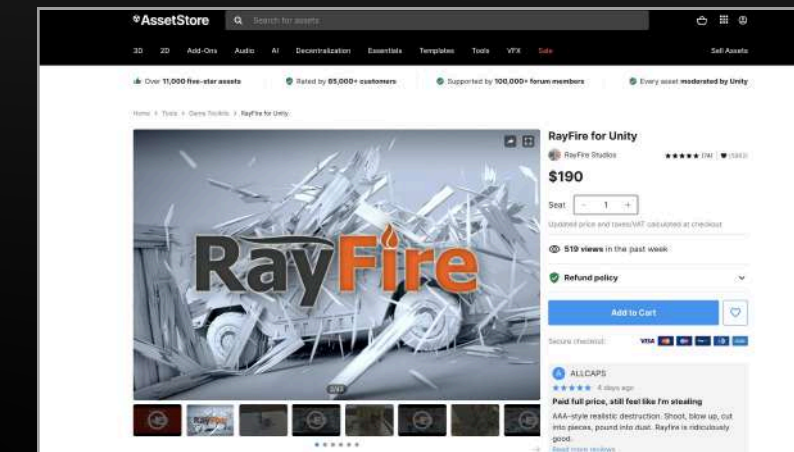


Programming | code sample

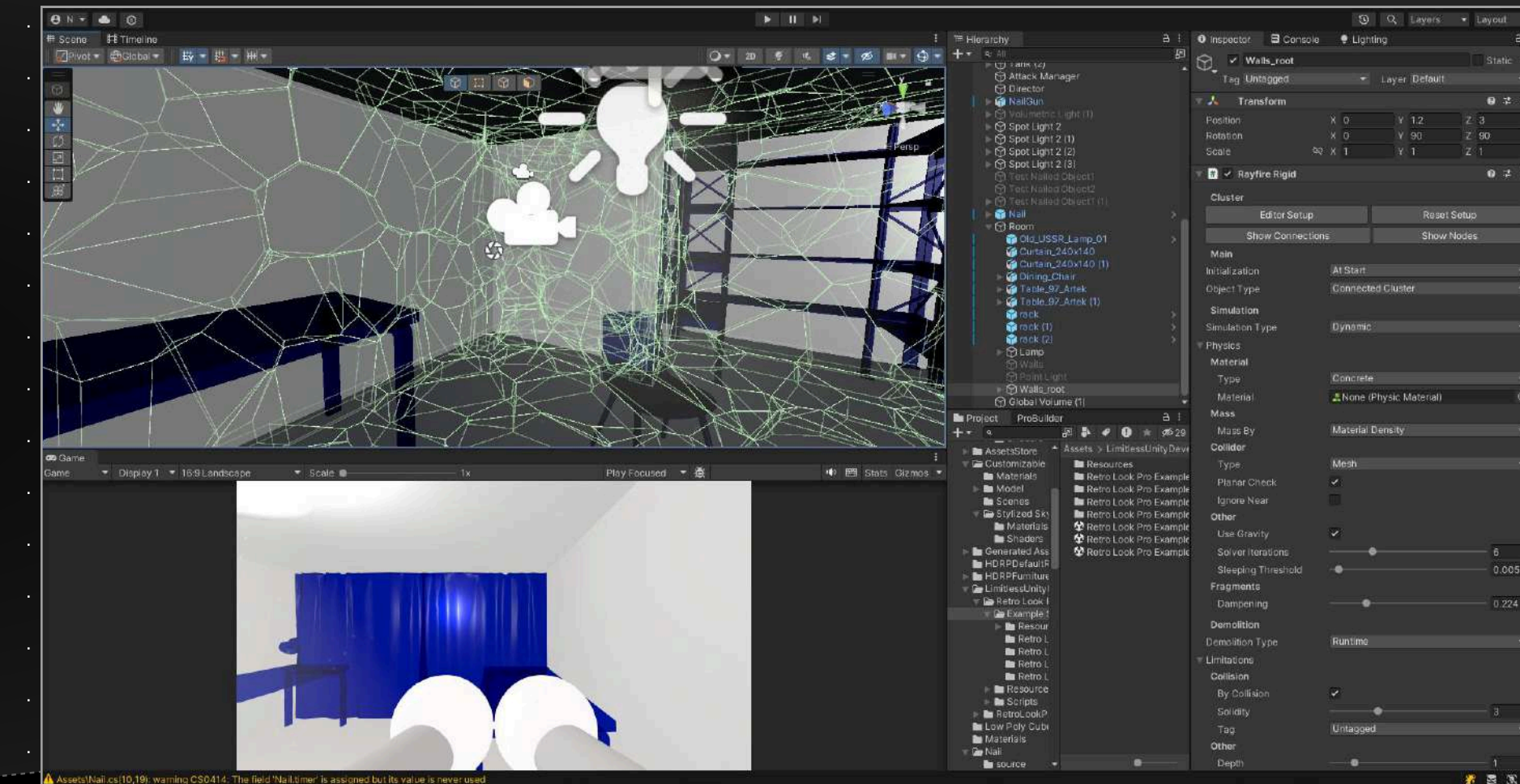
Implementation of Custom Bullet Launching Logic with Specified Position, Rate, and Force

Custom Bullet Launching Logic Implementation.

Nail Collision and Parenting Logic Implementation



The wall demolition effect is achieved through the rayfire plugin. Specifically, it is achieved through first fragmenting the room into small fragments. Through rayfire rigidbody component, we then mark these fragments as a connected cluster to merge those fragments into a solid demolitable wall.



In-game Screenshots

