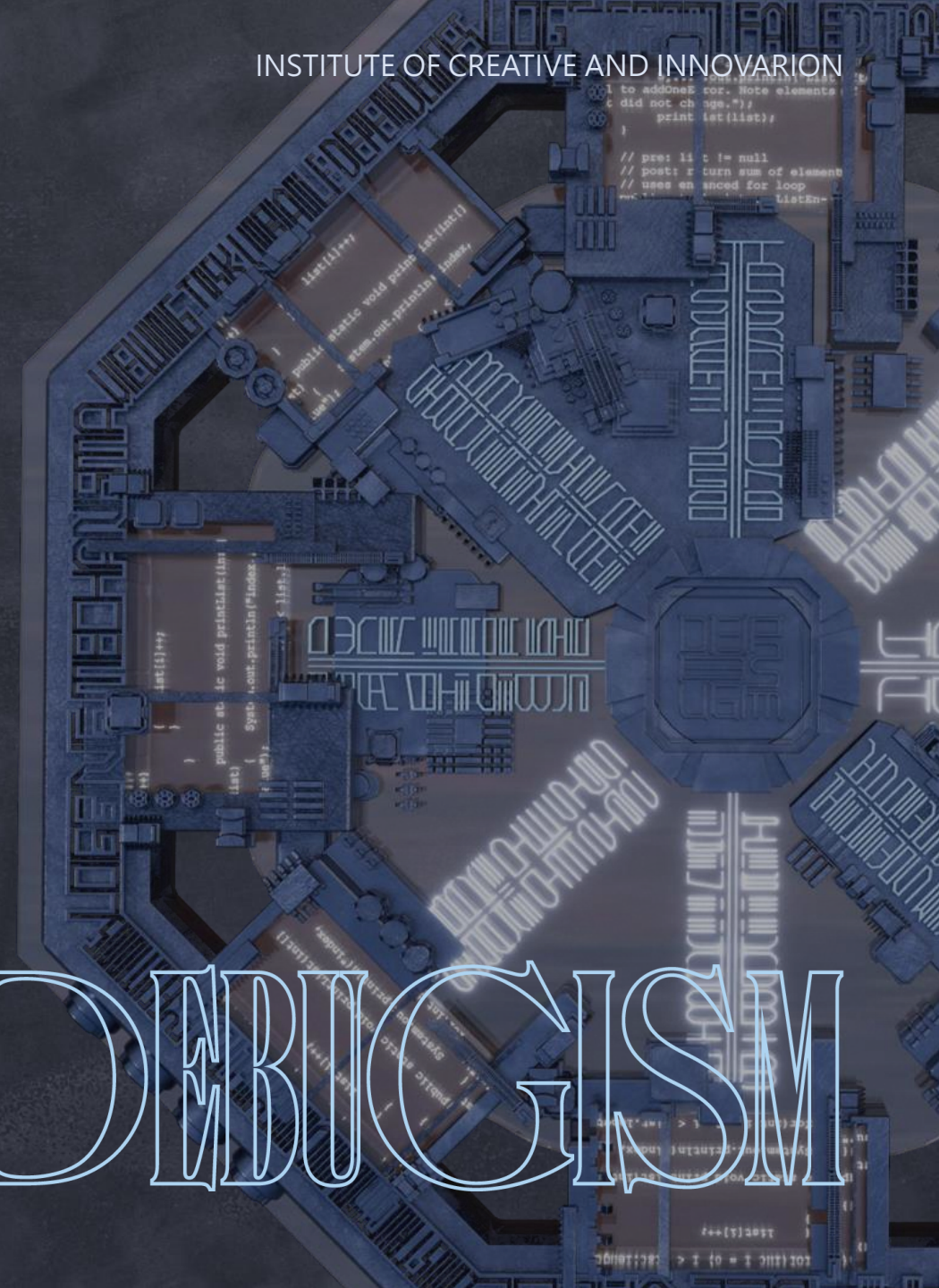
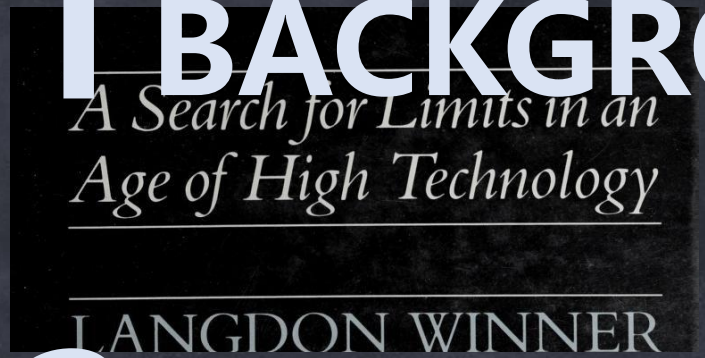
What If Bugs In Technological Products Promote A Religion In The Future?

Revealing The Impact Of Societal Dependence On Technology Through Speculative Design



CHEN CHUNXI

CONTENT



BACK GROUND

"WELCOME TO THE WORLD OF HIGH-RISK TECHNOLOGIES."

—— Charles Perrow, Normal Accident

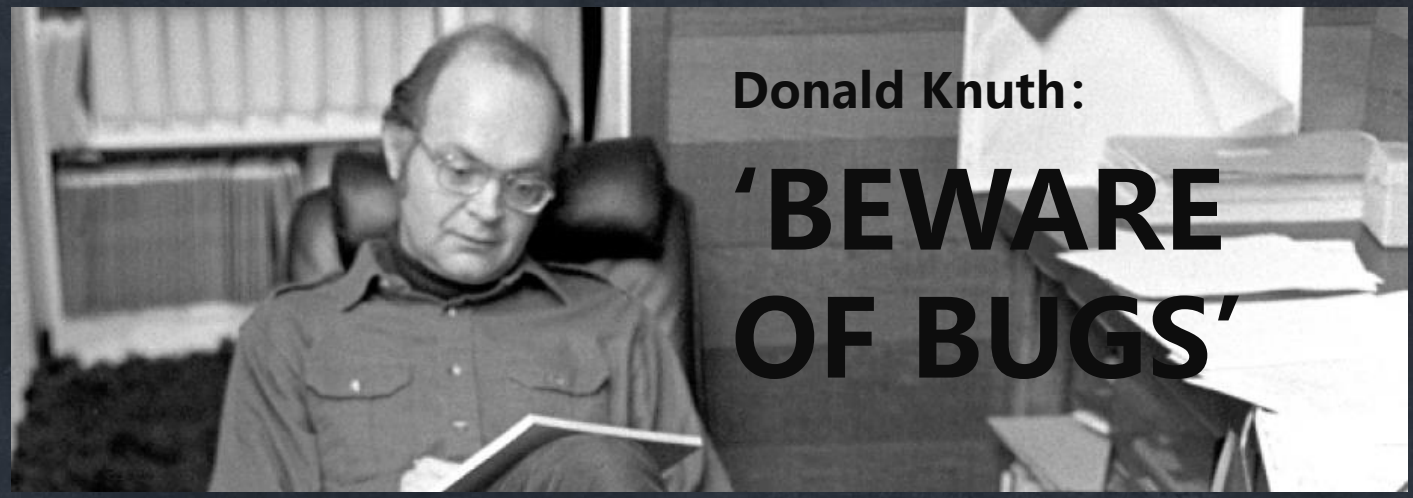
- The cultural phenomenon of technology fetishism and technology dependence is rampant in the digital age.
- Technological instability has increased over the past decade.
- The contradiction between the imperfection of technology and society's blind trust in it has become a defining feature of the contemporary world.

DESIGN STATEMEN

The unreliability and fragility of programs are becoming the most fatal problems of contemporary technology. However, the cultural phenomenon of technology worship and technology dependence still exists. Reflecting on our relationship with technology is particularly important at a time when the duality between technology and humanity is breaking down.DEBUGRISM presents a future scenario through critical design, which aims to reflect and promote discussion of the impact of contemporary technology by exploring the forms of religious belief that technology worship, technology dependence, and programming errors may lead to in the future through design language.

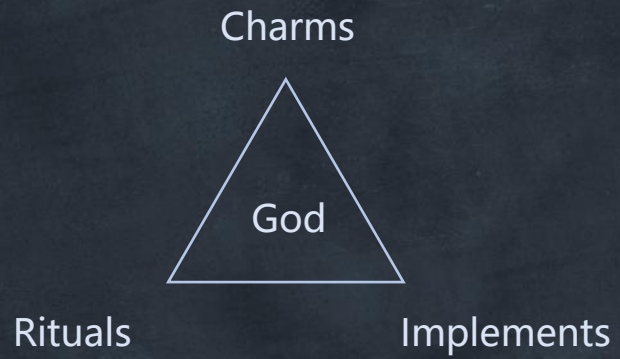
DESIGN IDEA

START POINT

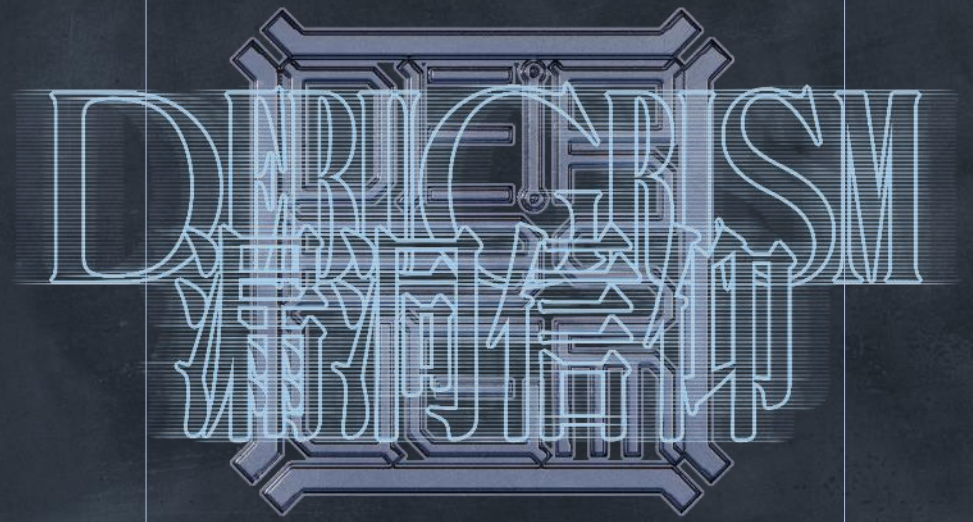


- The introduction of machine algorithms leads to more and more complex and difficult to solve bugs.
- Over the past decade, the number of major incidents due to bugs has increased year by year.

RELIGIOUS PRACTICE



A form of religion in which the deity is empowered through spells, rituals, and rituals.



As more complex Bugs escape human control, people begin to believe that there is a supernatural force in the digital environment that can control Bugs, and gradually form a new religion – DEBUGRISM.

VISUAL FORMS

Poster

Charms and holistic religious visual concept

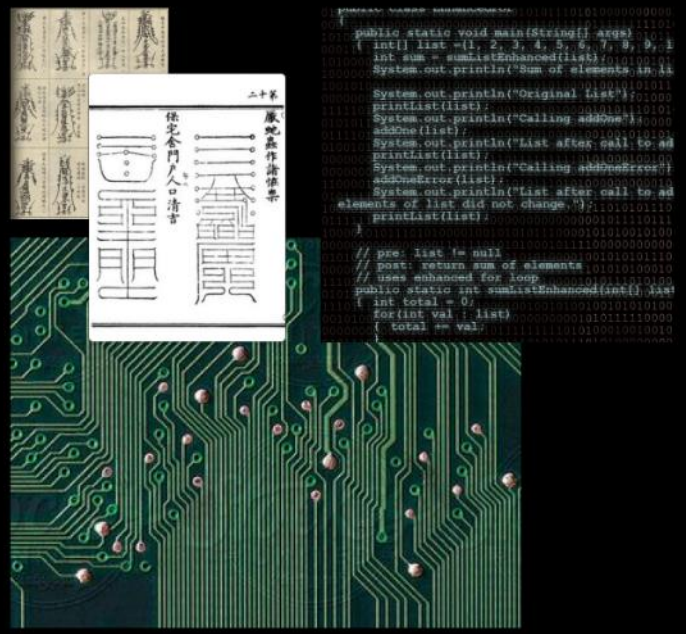
Book2

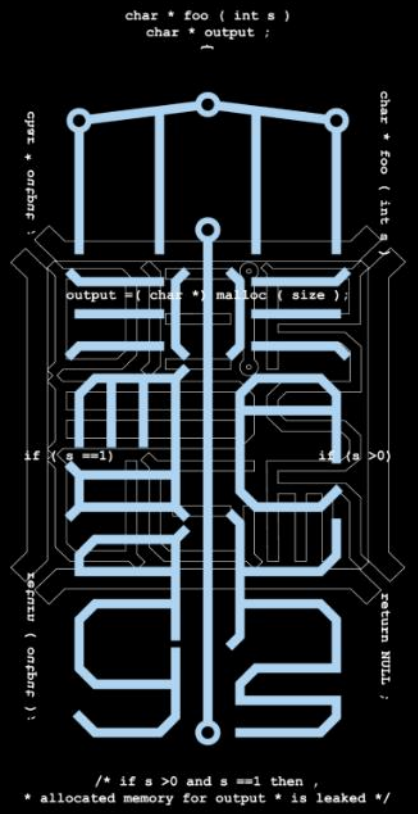
16 sets of charms and corresponding program bugs

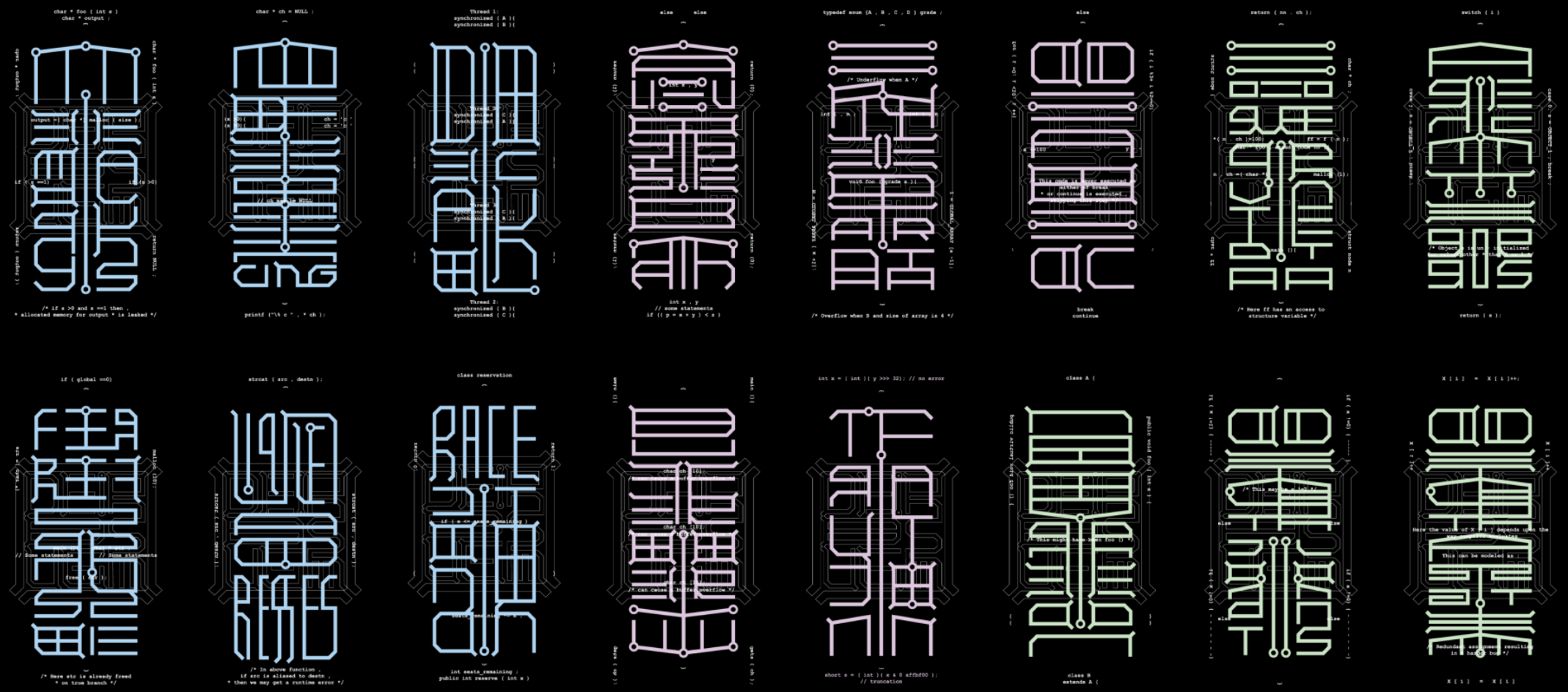
Book1

Religious stories, religious rituals

CHARMS VISUAL TRANSFORMATION



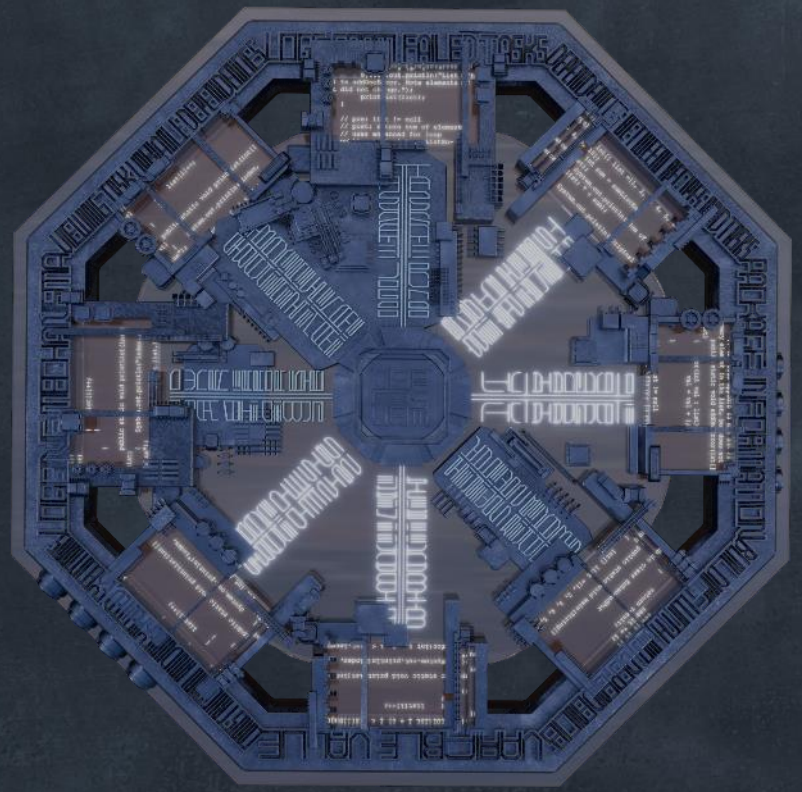


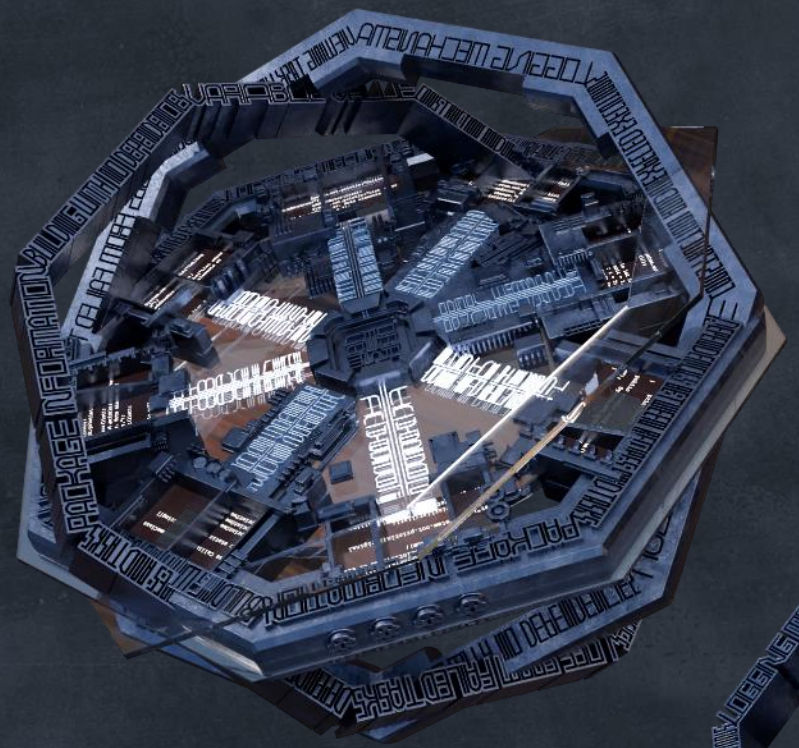


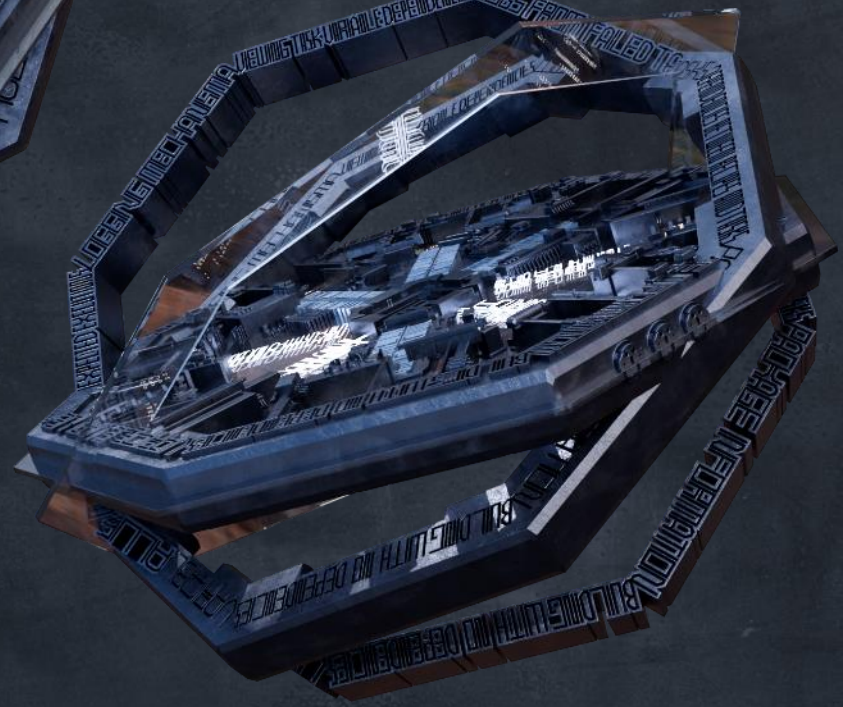
IMPLEMENTS VISUAL TRANSFORMATION

Extract visual elements from traditional religions

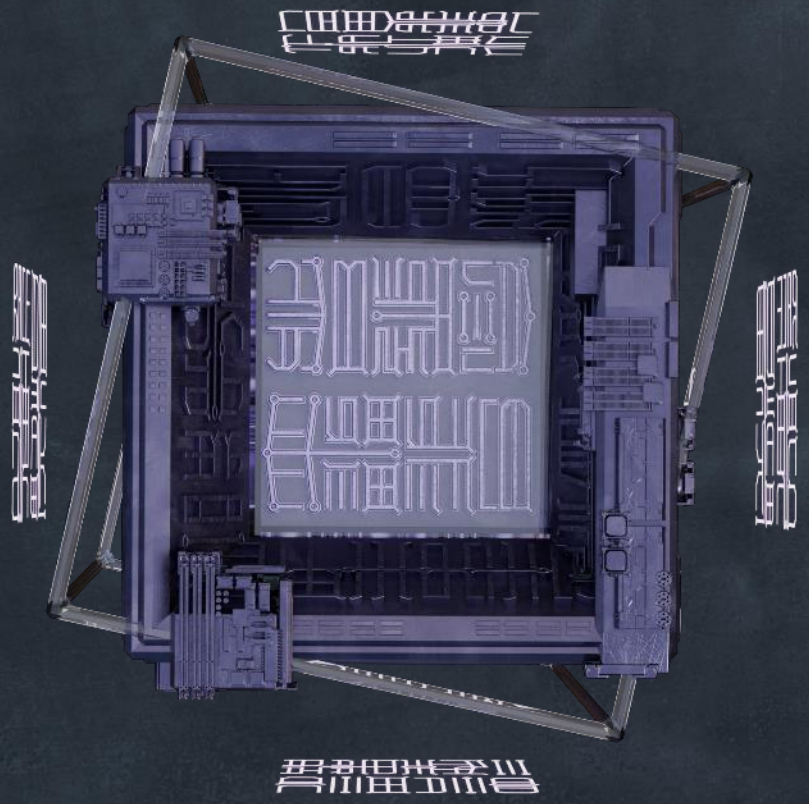
Compass

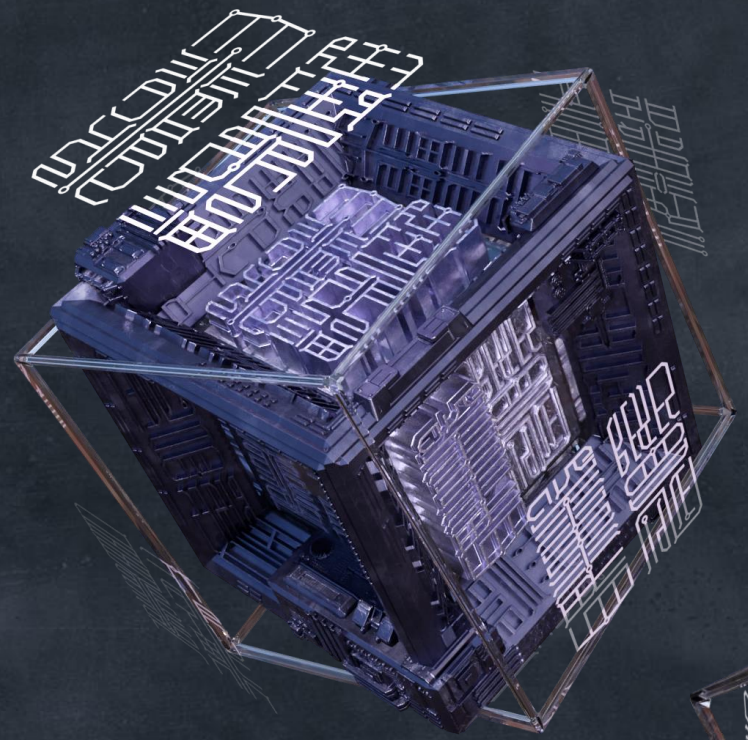


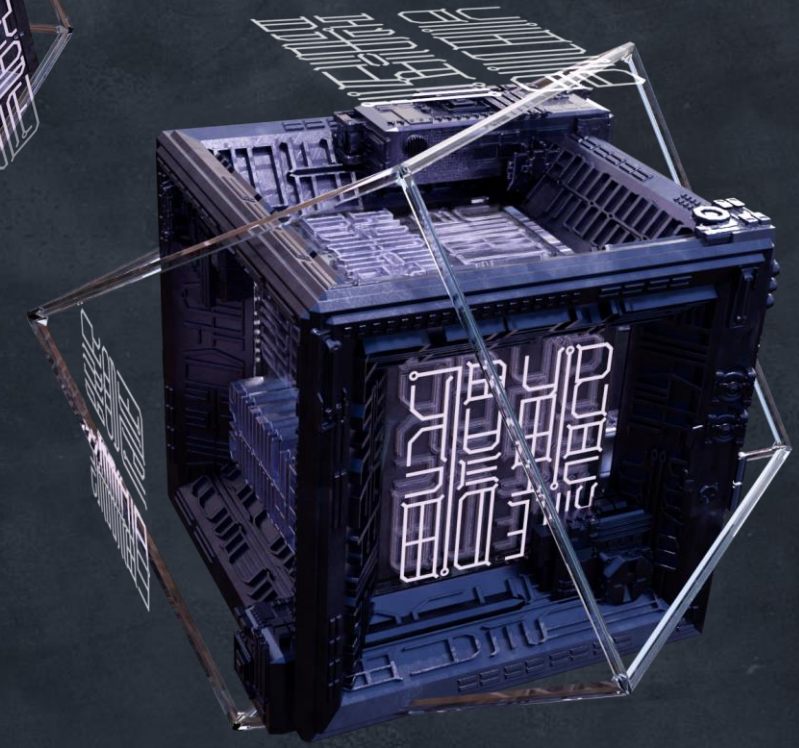




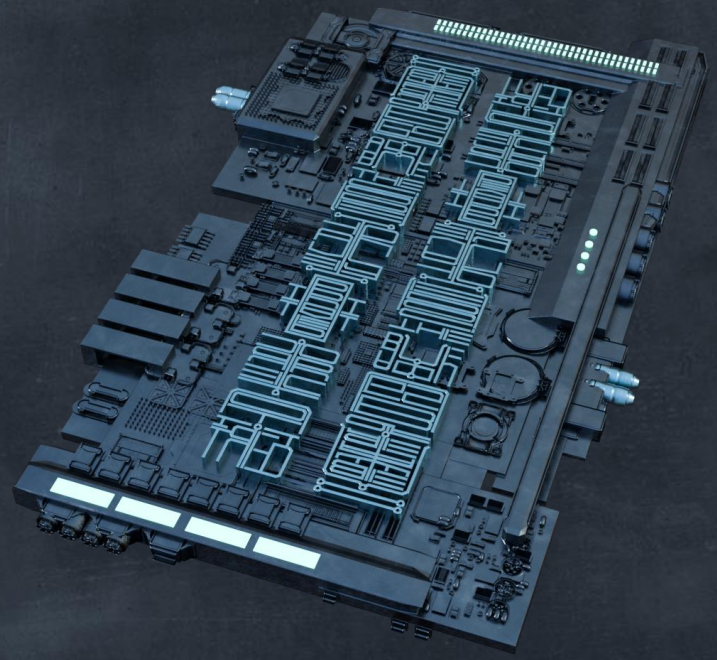
Sign

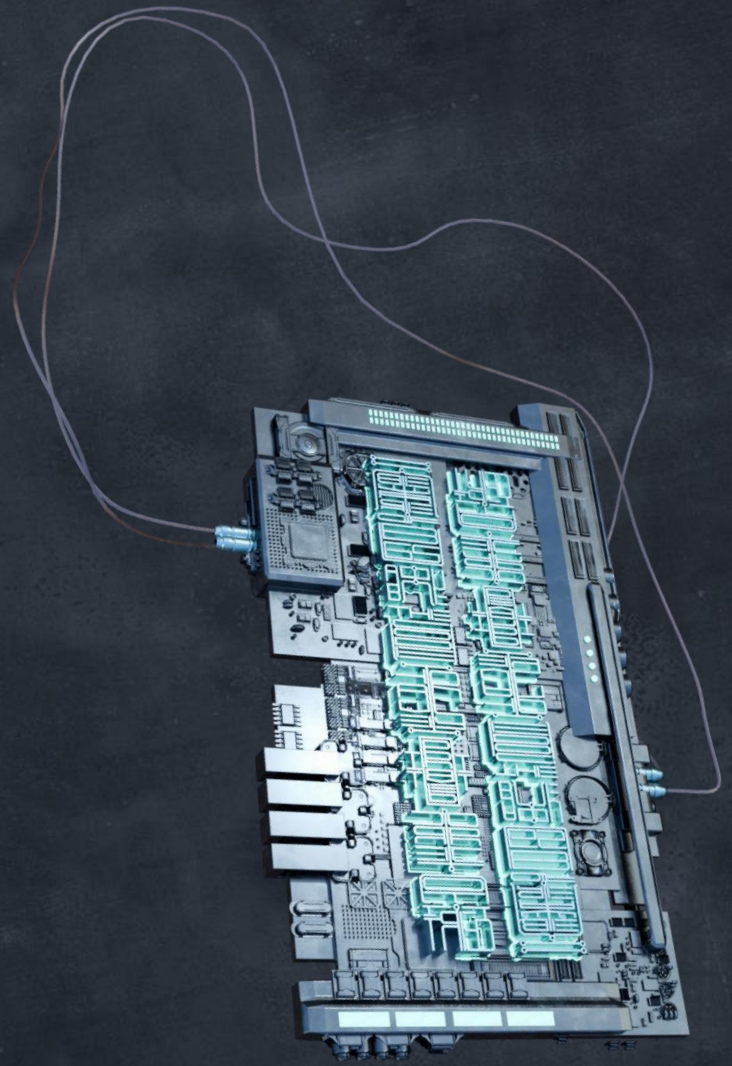






Token

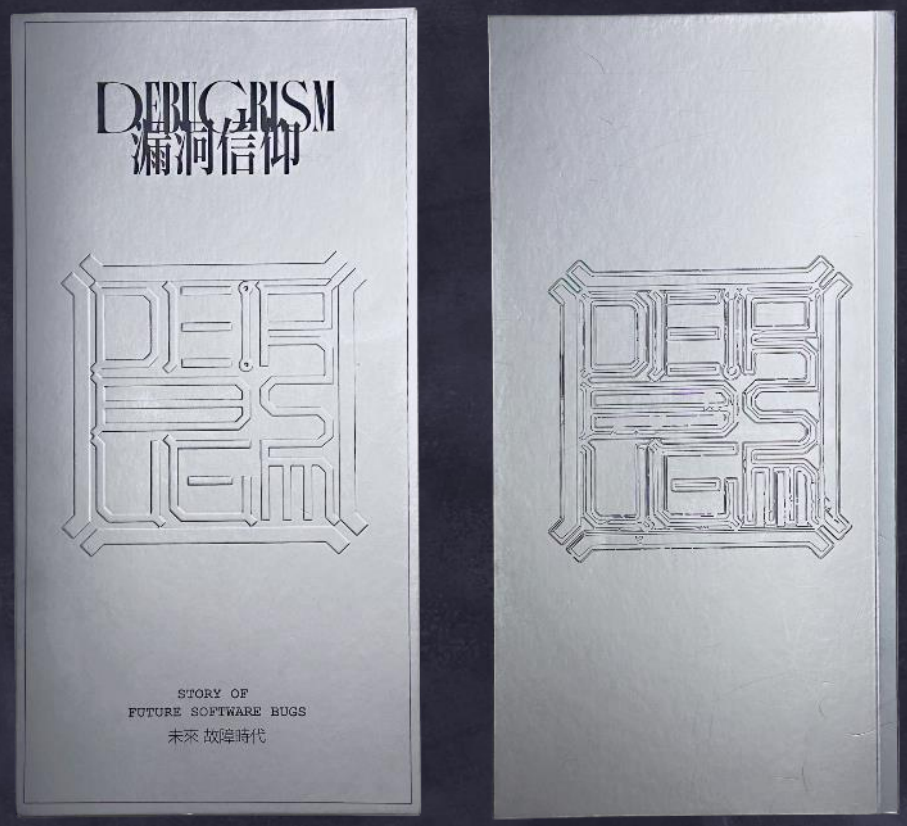


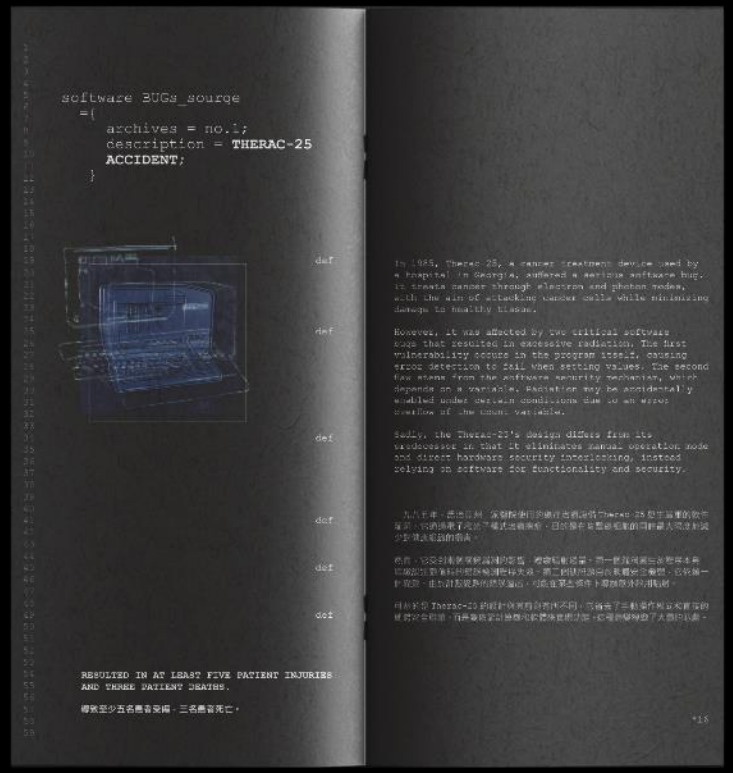


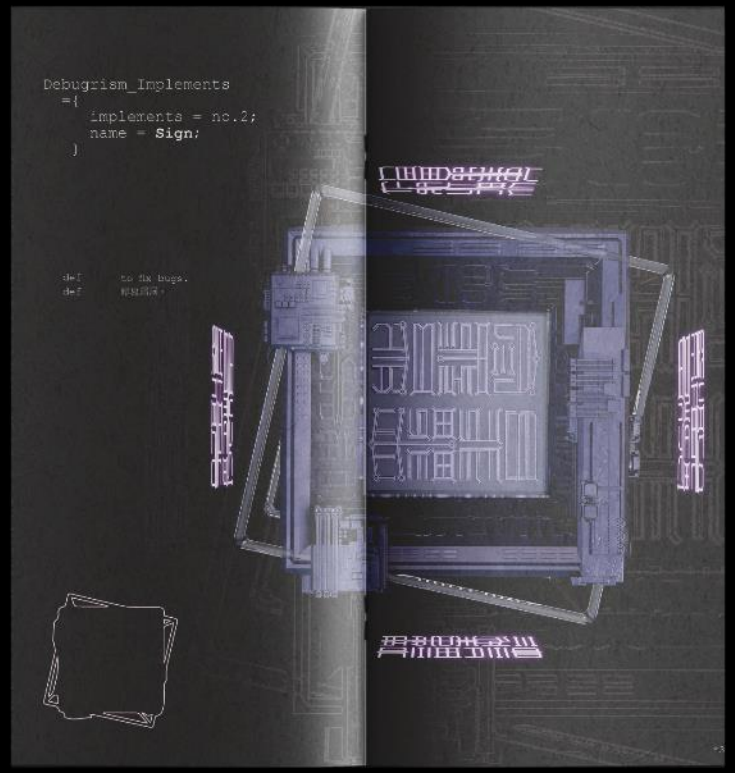
DESIGN OUTCOME

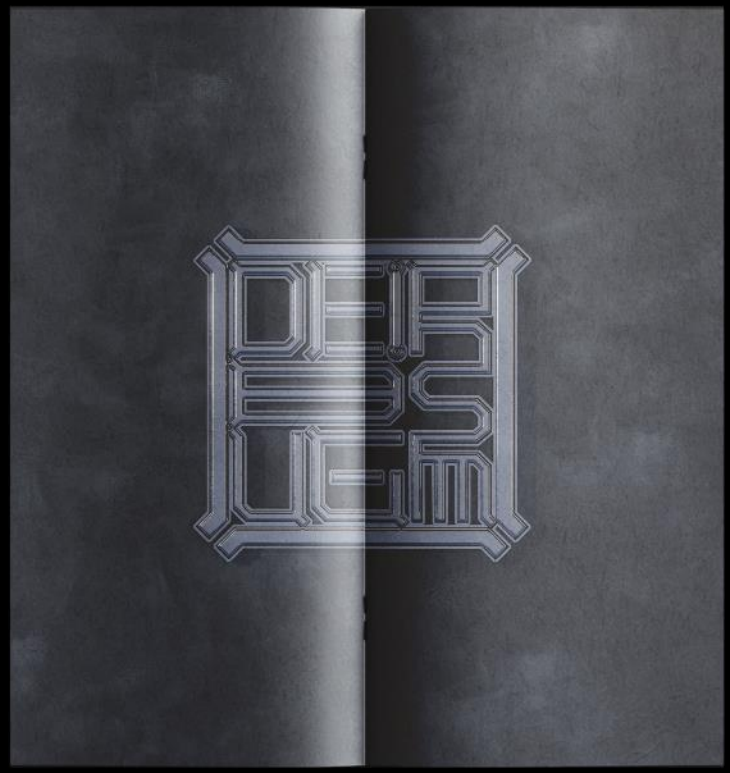
- BOOK-DEBUGRISM Secret Book
- FOLD-Charms Guide
- Poster-DEBUGRISM Implement
- Other Visual Items

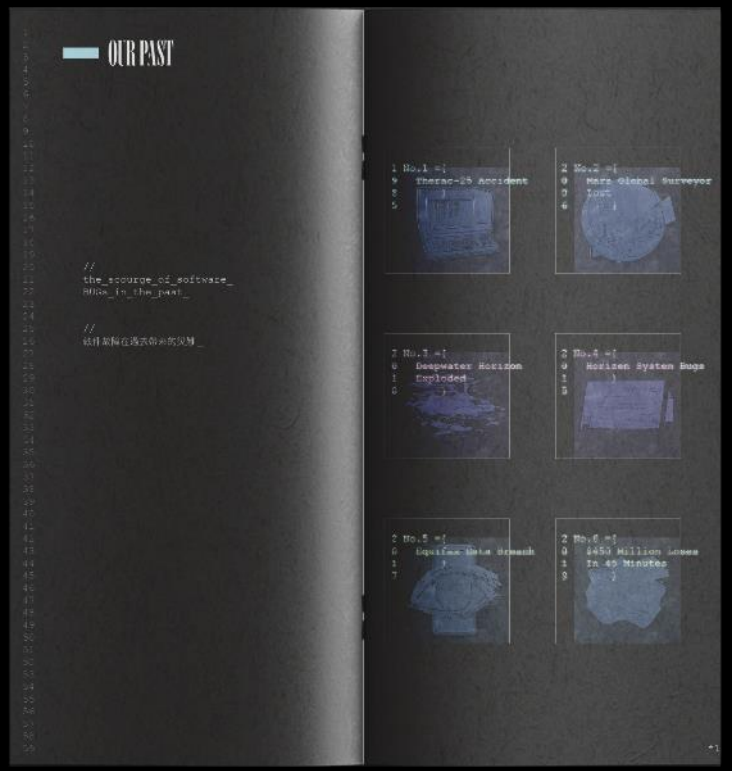
BOOK-DEBUGRISM
Secret Book

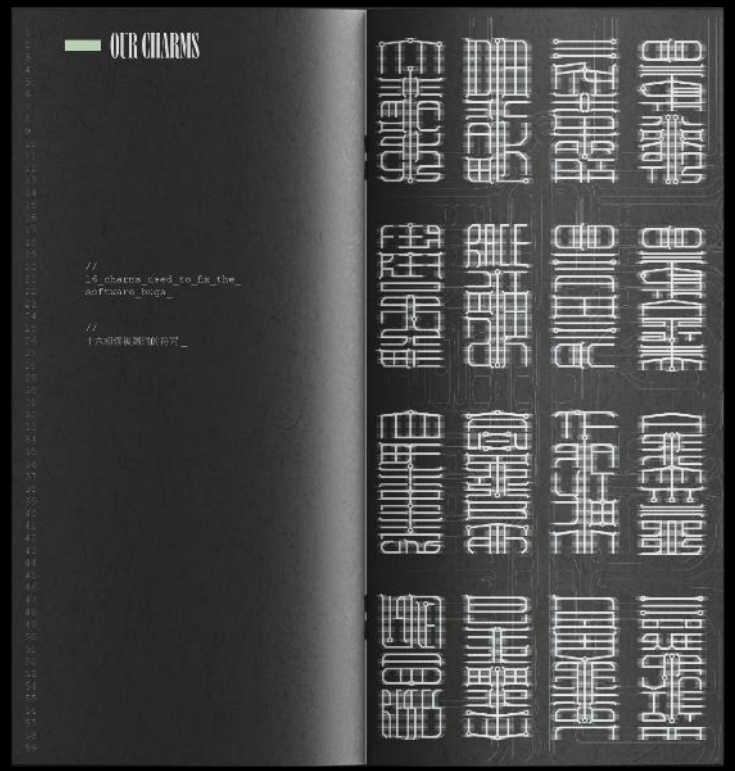



software BUGs_source
={
archives = no.1;
description = THERAC-25
ACCIDENT;
}
def
def
def
def
def
def
RESULTED IN AT LEAST FIVE PATIENT INJURIES
AND THREE PATIENT DEATHS.
导致至少五名患者受伤，三名患者死亡。


Debugrism_Implements
={
implements = no.2;
name = Sign;
}
def to fix bugs.
def



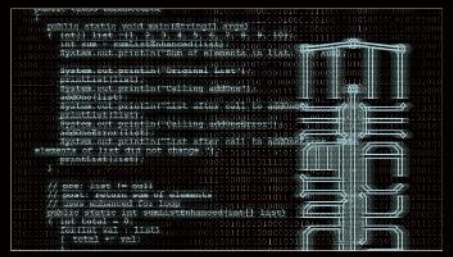

OUR PAST


OUR CHARMS

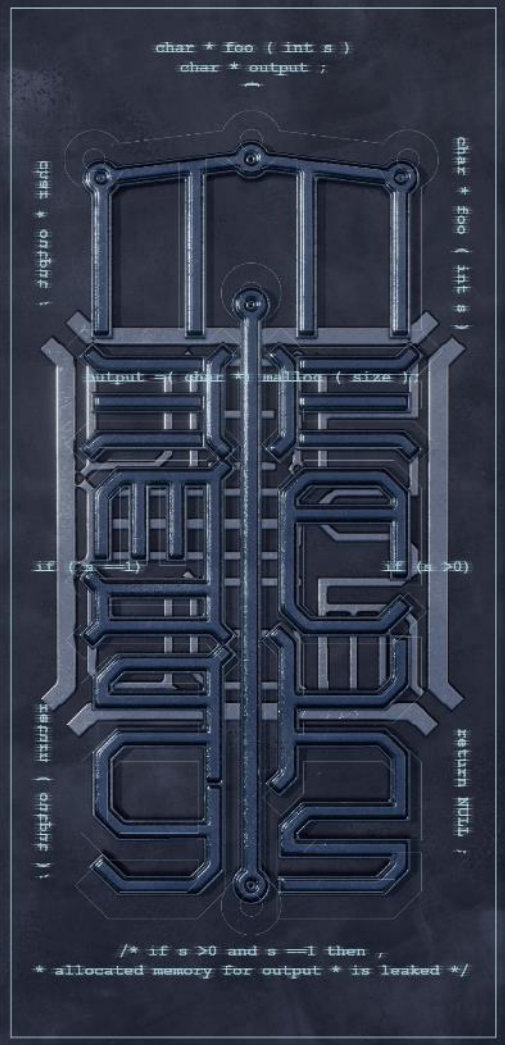
FOLD-Charms Guide

charm_no.1
={
name = **De Memory Leaks;**

咒 = 釋放未結縛记忆流清
}

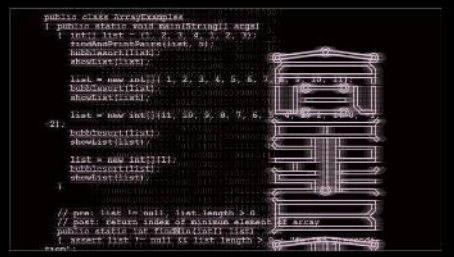


```
char * foo ( int s )
{
char * output ;
if (s >0)
output =( char *) malloc ( size );
if ( s ==1)
return NULL ;
return ( output );
/* if s >0 and s ==1 then ,
* allocated memory for output
* is leaked */
}
```

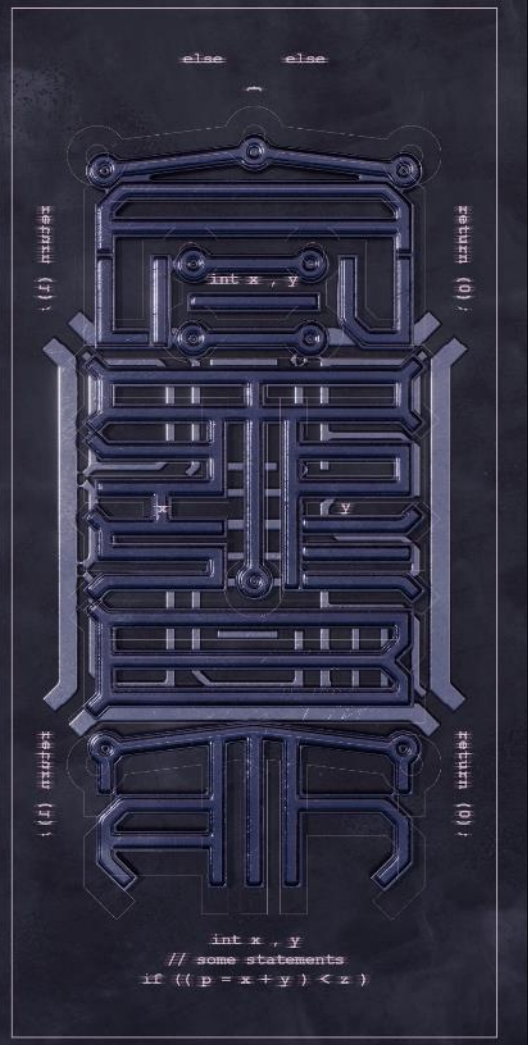



charm_no.7
={
name = **DE Value Outside Domain;**

咒 = 保域內值無誤



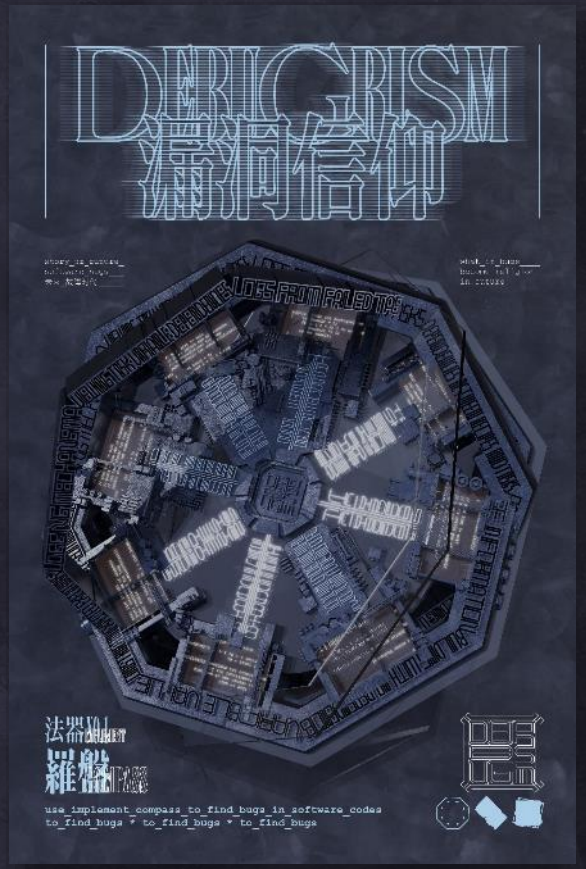
```
int x , y
// some statements
if (( p = x + y ) < z )
return (1);
else
return (0);
```

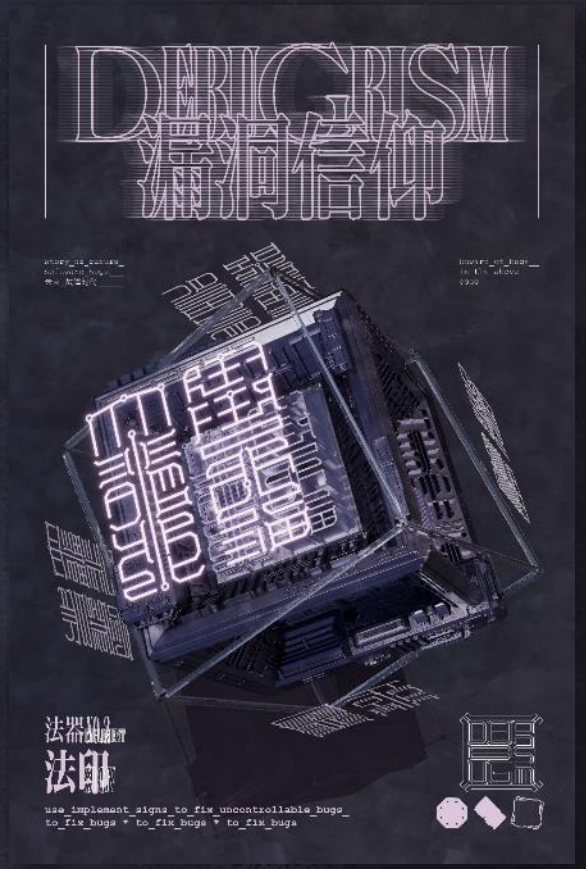



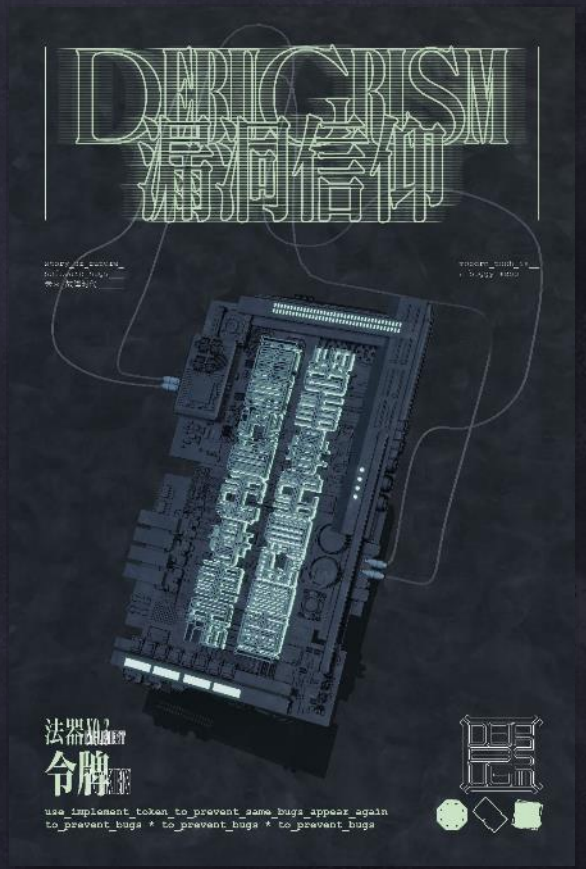

charm_no.10
name = DE Ftruncation;
charm_no.11
name = DE Dead Code;


chapter_no.4
name = DE Unique Addresses;
死 = 避世界交叉

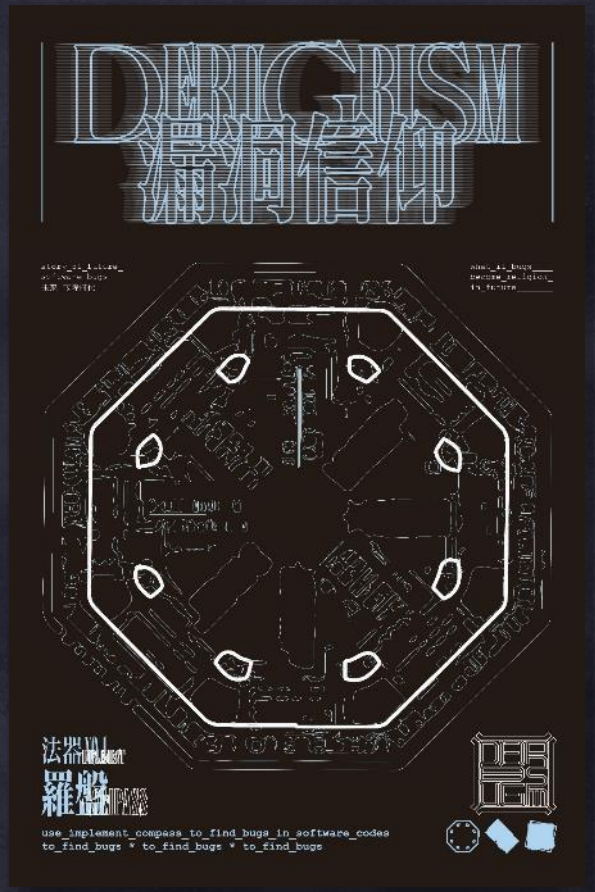
Poster-DEBUGRISM Implement

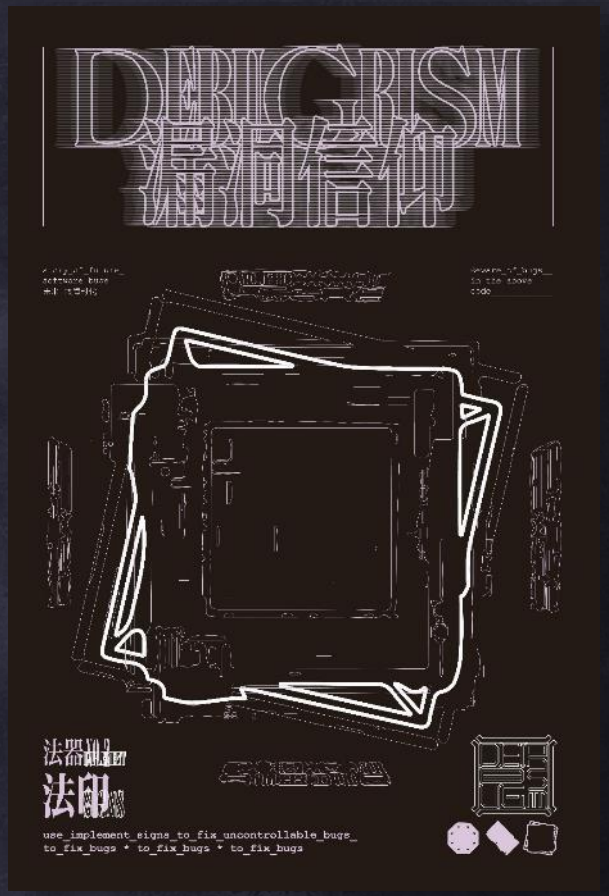


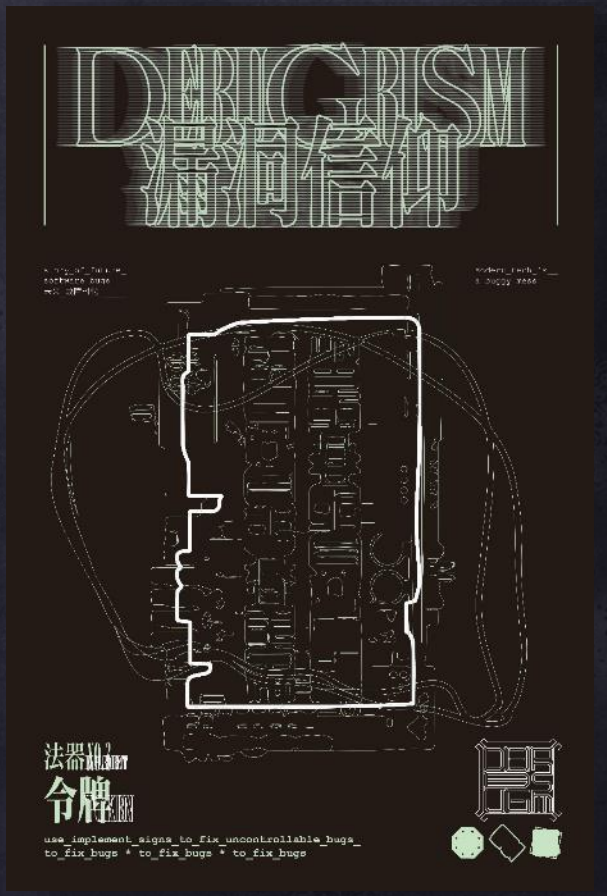


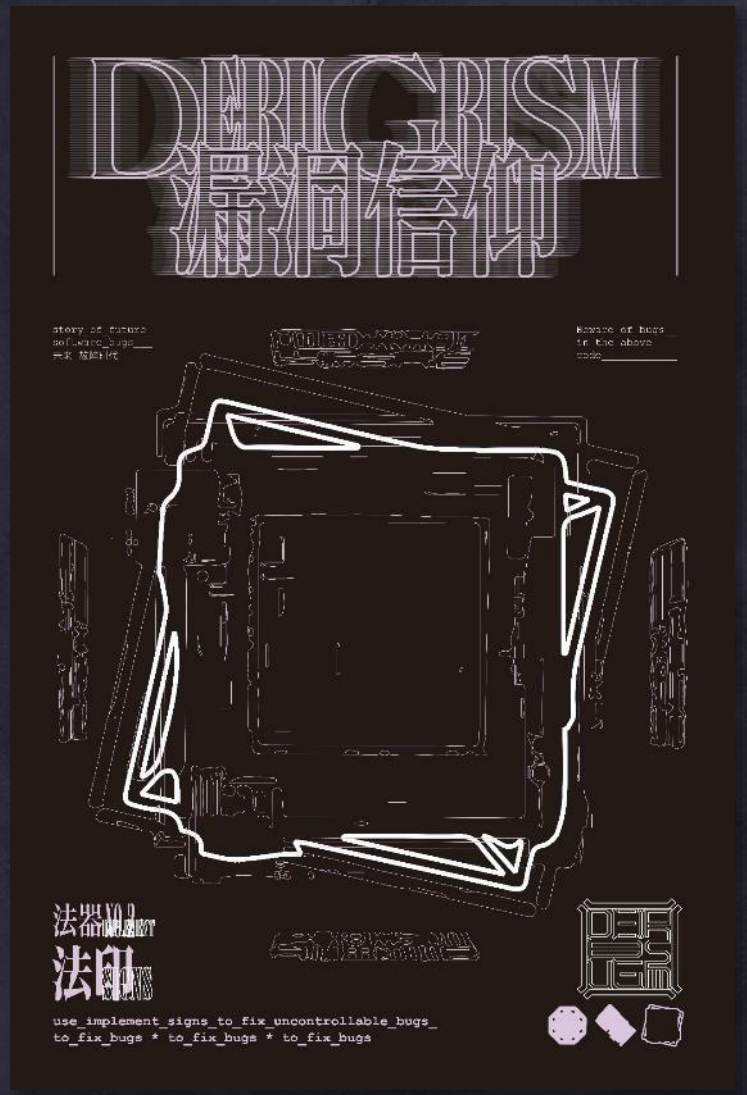


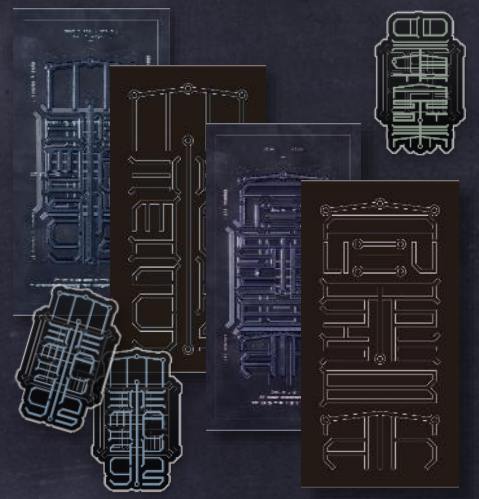
Other Visual Items








DEBUGRISM
漏洞信仰
法器NO.2
use_implement_signs_to_fix_uncontrollable_bugs_
to_fix_bugs * to_fix_bugs * to_fix_bugs




DEBUGRISM
漏洞信仰

THANKS FOR WATCHING