ZilGAM

Zilliqa Generative Art and Music

Table of Contents

01 Project Overview

02 Phases

03 Technical Overview

04 Conclusion

01

Project Overview

How it works!

Step 1

Pick a style you like!

Step 2

Connect your ZILPay wallet to our platform

Step 3

Mint your NFT using our frontend interface

Step 4

View your uniquely generated art piece

Our Team

Project Maintainer

Shyam Sridhar

Core Devs

Madhumitha Balaji, Saakshi Saraf, Kaveri Priya, Aishwarya Nair

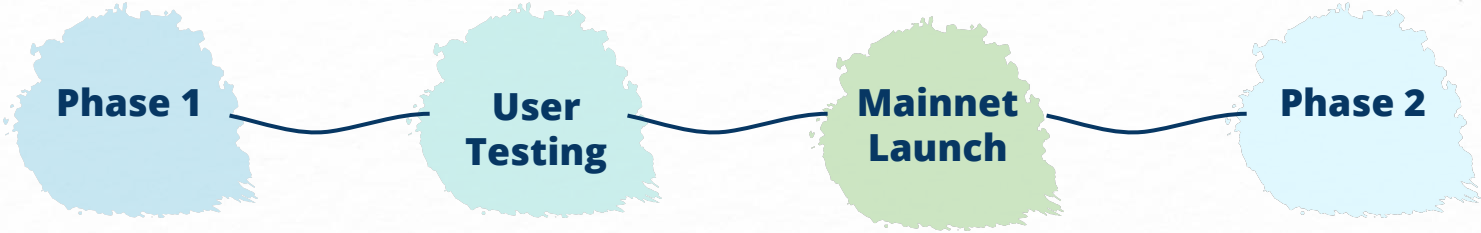
Strategic Advisors

Zilliqa Team, Qing Ze Hum

02

Phases

Overall timeline



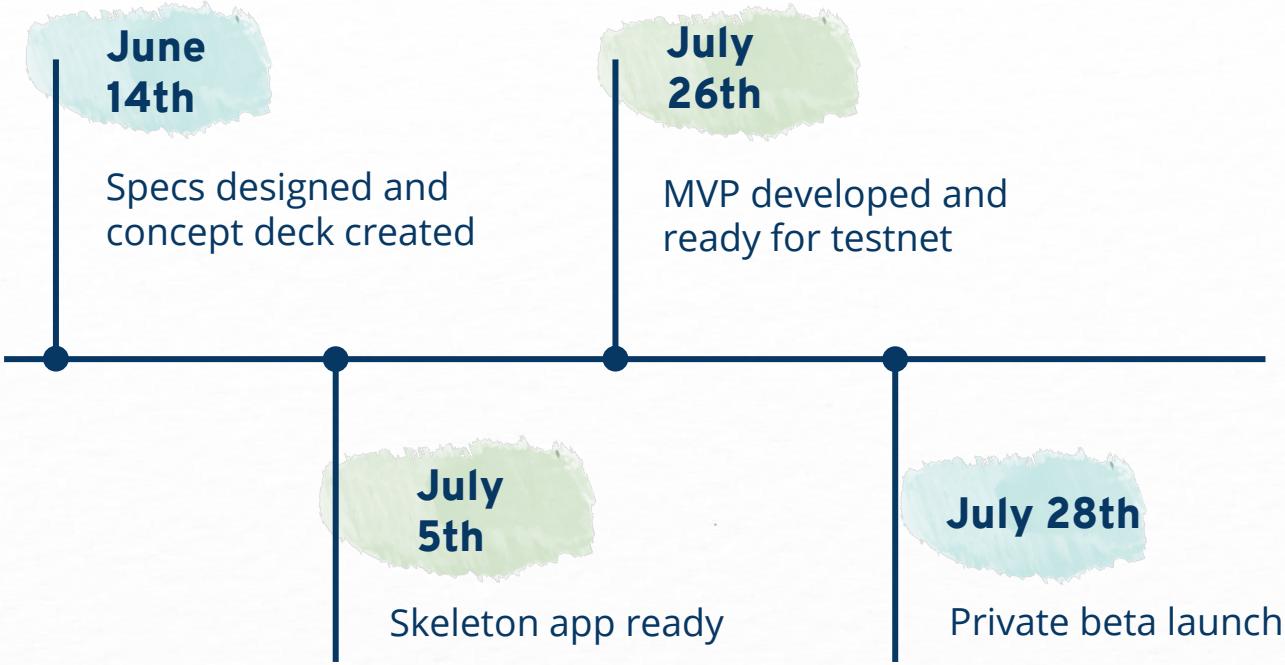
MVP developed and launched on the testnet

Extensive testing done to prevent bugs

To the moon!

API development
Artist onboarding
Tokenized grouping system

Immediate timeline



03

Tech Specs

Skeleton app – Component 1 Generative Algorithm

- 2 initial projects to go live on the platform – nothing fancy just demo projects
  - Generative music project
  - Generative art project

  We will use existing examples as starter codes and modify them to fit our needs

- A generative script based on p5.js is written and stored in a repository.
- Parameters of the script such as the colour, size of objects, angles etc. will depend on a hexadecimal string generated in a pseudo-random manner

```
"0x11ac128f8b54949c12d04102cfc01960fc496813cbc3495bf77aeed738579738"

function random_hash( ) {
        let chars = "0123456789abcdef" ;
        let result = '0x' ;
        for ( let i = 64;  i > 0;  --i ) result  +=  chars[Math.floor(Math.random() * chars.length] ;
        return result ;
        }
```

Reference: https://github.com/ArtBlocks/docs

Skeleton app – Component 1 Generative Algorithm

- Generate a unique number in the range of 0-10 using this hexadecimal string

```
let seed = parseInt(tokenData.hash.slice ( 0, 16), 16 ) ;
let p = [ ];
for (let j = 0; j < 32; j++) {
      p.push(tokenData.hash.slice(2  + ( j* 2), 4 + (j * 2)
) ) ;
}
let rns = p.map( x => {
      return parseInt(x, 16) % 10;
}) ;
```

- Use this to parameterize different variables

```
let dynamic = rns[0] >9
let size = rns[1] > 5? 10 : 20
let color = rns[2] > 7 ? “white” : “black”
```

Reference: https://github.com/ArtBlocks/docs

Skeleton app – Component 1 Generative Algorithm

- **p5.js example: interacting with a movable text [in this case “tickle”]**

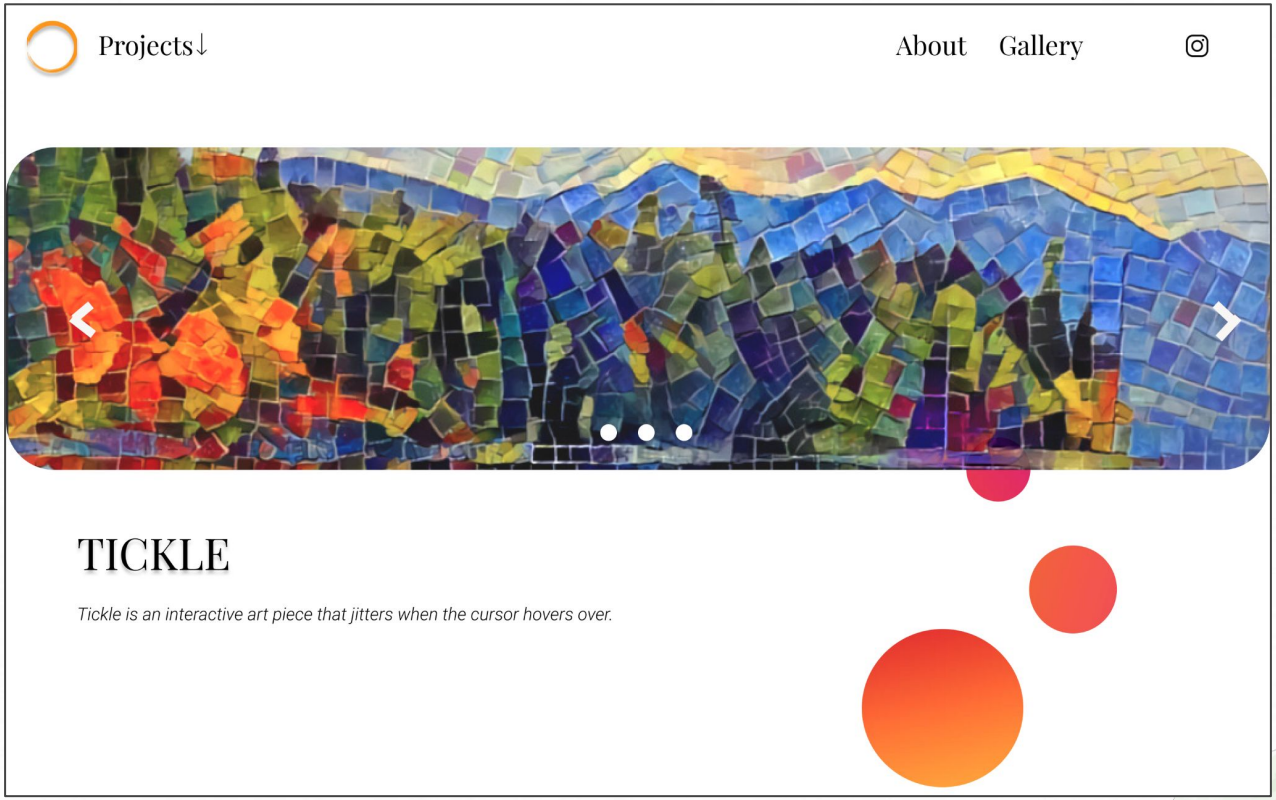
```
let message = ‘tickle’ ,
        font,
        Bounds,
        fontsize = 60
        x,
        y;
function preload() {
        font = loadFont(‘assets/SourcesSansPro-Regular.otf’);
}

function setup() {
        createCanvas(710, 400);
        textFont(font);
        textSize(fontSize);
        bounds = font.textBounds(message, 0 ,  0, fontSize);
        x= width / 2 – bounds.w / 2;
        y= height / 2 – bounds.h / 2 ;
}
function draw() {
        background(204, 120); //write the text in black
                              and get its bounding box
        fill(0);
        text(message, x, y);
        Bounds = font.textBounds(message, x, y, fontsize);
        If(
        mouseX >= bounds.x &&
        mouseX <= bounds.x + bounds.w &&
        mouseY >=bounds.y &&
        mouseY <= bounds.y + bounds.h
        ) {
        x+= random(-5, 5);
        y+= random(-5, 5);
        }
}
```

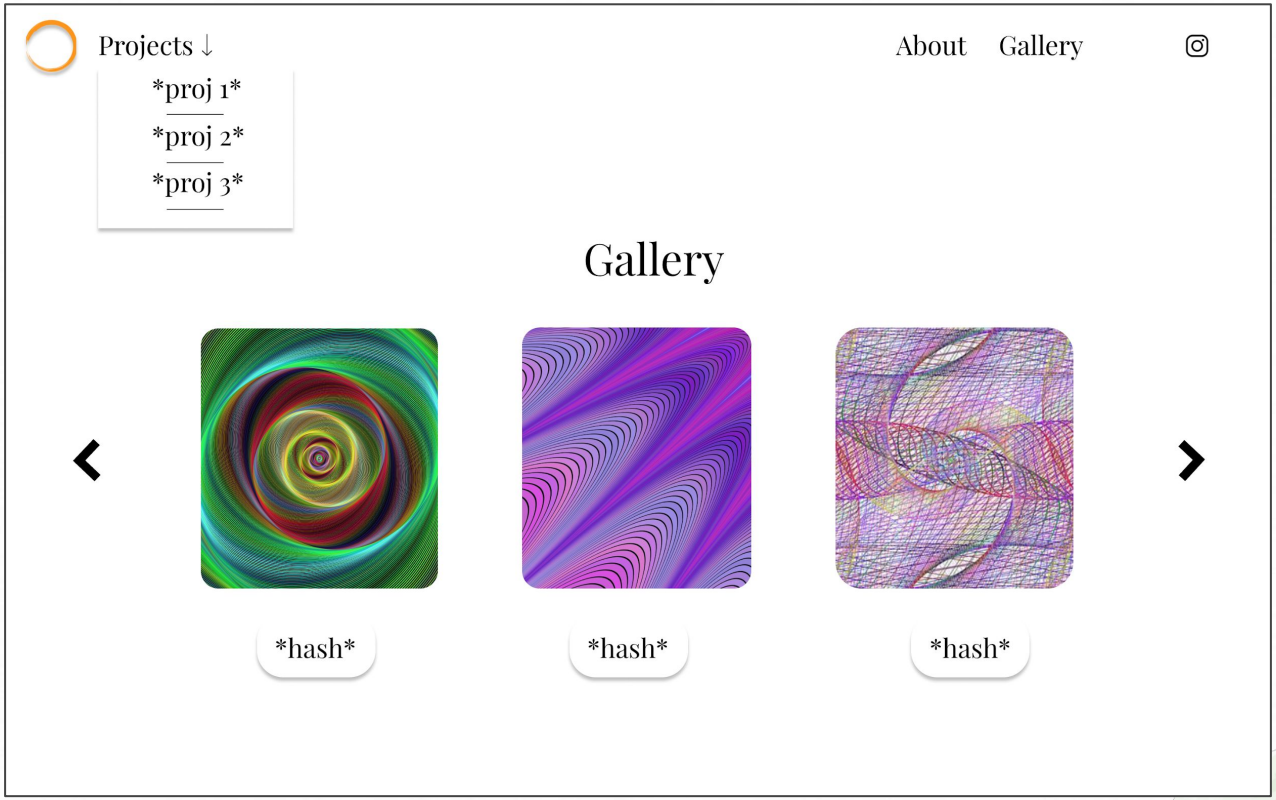
The word "tickle" jitters when the cursor hovers over. Sometimes, it can be tickled off the screen.

Reference:
https://p5js.org/examples/sound-noise-drum-envelope.html

Skeleton app – Component 2
Web interface mockup



Skeleton app – Component 2
Web interface mockup



Skeleton app – Component 1 Generative Algorithm

- **p5.js example: Noise drum envelope**

White noise is a random audio signal with equal energy at every part of the frequency spectrum
An Envelope is a series of fades, defined as time/value pairs
In this, the p5.Env will be used to “play” the p5.Noise like a drum by controlling its output amplitude. A p5.Amplitude will get the level of all sound in the sketch, and we’ll use this value to draw a green rectangle that shows the envelope in action

To run this example locally, you will need the p5.sound library and a sound file

Reference:
https://p5js.org/examples/interaction-tickle.html

```
let noise, env, analyzer;
function setup() {
        createCanvas(710, 200);
        noise = new p5.Noise();    // other types include
      'brown' and 'pink'
        noise.start();
         // multiply noise volume by 0
         // (keep it quiet until we're ready to make noise!)
         noise.amp(0);
         env = new p5.Env();
         // set attackTime, decayTime, sustainRatio,
       releaseTime
         env.setADSR(0.001, 0.1, 0.2, 0.1);
         // set attackLevel, releaseLevel
         env.setRange(1, 0);
         // p5.Amplitude will analyze all sound in the sketch
         // unless the setInput() method is used to specify an
       input.
         analyzer = new p5.Amplitude();
}
function draw() {
         background(0);
         // get volume reading from the p5.Amplitude analyzer
         let level = analyzer.getLevel();
         // use level to draw a green rectangle
         let levelHeight = map(level, 0, 0.4, 0, height);
         fill(100, 250, 100);
         rect(0, height, width, -levelHeight);
}
function mousePressed() {
       env.play(noise);
}
```

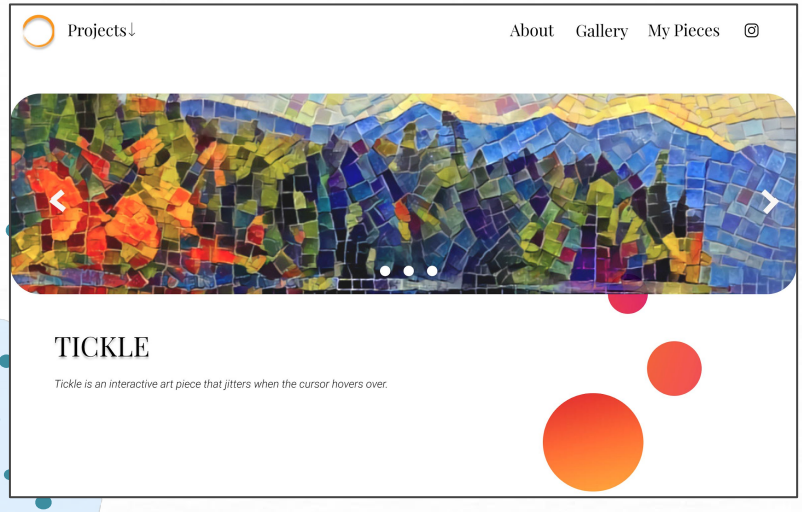
MVP- Component 1
Smart Contract

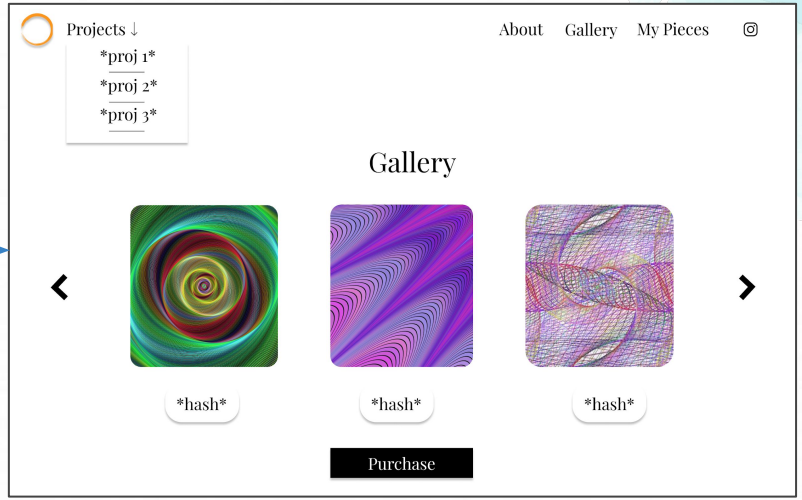
- Modified ZRC – 1 contract (https://github.com/Zilliqa/ZRC/blob/master/reference/nonfungible-token.scilla)
- Set the parameters: contract_owner, name, symbol, total_supply

MVP- Component 2
ZILPay integration

- Integrate the DApp with ZILPay or any other NFT Wallet which is ready for testing with
- Once the project goes mainnet – Mintable integration

MVP- Component 3 Frontend





getAddressTxs

Get the transactions made by an address.

Param	Type	Required	Default
page	Number	false	1

```
await client.getAddressTxs('zil16awfafxs789g8nthnm5s9p4l8vnxs5zpfr3upr', { page: 1 })
```

Output

```
{
  docs: [
    {
      hash: '0xced4731d535e1b0ba72d562d6c1dfd0ef71d1189abfa1ead8341667ca22aa01b',
```

Use this API to get the hash and use that as an input to the generative script

MVP- Component 3 Frontend

If nothing has been purchased, output will be ‘no purchases found’

getAddressTxs

Get the transactions made by an address.

Param	Type	Required	Default
page	Number	false	1

```
await client.getAddressTxs('zil16awfafxs789g8nthnm5s9p4l8vnxs5zpfr3upr', { page: 1 })
```

Output

```
{
  docs: [
    {
      hash: '0xced4731d535e1b0ba72d562d6c1dfd0ef71d1189abfa1ead8341667ca22aa01b',
```

Use this API to get the hash and use that as an input to the generative script

04

Conclusion

User selects the style of art or music that he likes

A detailed description of the project can be found under the ‘About the Project’ tab

↓

Connect to wallet

Integrated with ZilPay

↓

Receive an NFT

Minted using a deployed ZRC-1 contract

↓

View the uniquely generated art or music piece!

The unique transaction hash used as an input to the generative script to parametrize variables such as size, colour, speed etc.

Thank you

CREDITS: This presentation template was created **by Slidesgo,** including icons **by Flaticon, infographics & images by Freepik**

Please keep this slide for attribution