Tetris

廖宏祐

專案架構

01

前端

由html、javascript、css三檔組成

02

後端node.js

用express架構來處理寫入跟輸出的處理

03

JSON

處存排行榜結果

程式主要四大類

01

遊戲邏輯

Tetris主要的遊戲流程。

02

排行榜

負責從tetris server json檔案讀取資料及輸出資料

03

音效與音樂

背景音樂及按鍵音效

04

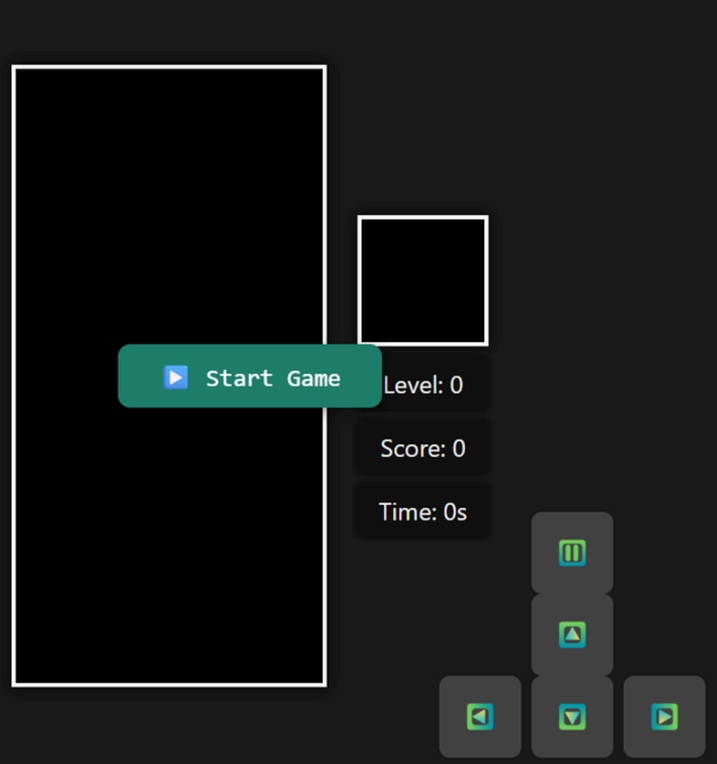
鍵盤與按鍵

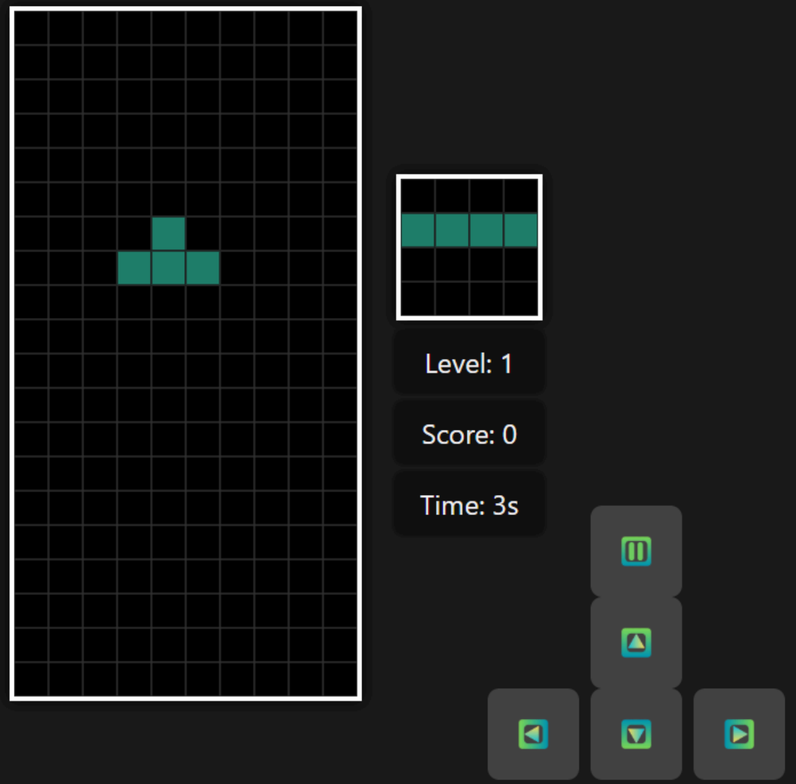
操控方塊的按鍵及暫停鍵

Start()

從startbutton按下後
先畫背景格線
再畫下一個方塊的預覽
進入update()主程式

```
318  function start() {
319      context.fillStyle = "#000";
320      context.fillRect(0, 0, COLS, ROWS);
321      drawGrid();
322      drawGridsmall();
323      drawShapesmall(nextShape, 0, 0, "#1F7F6C");
324      gameLoop = setInterval(update, 500);
325  }
```





update()

是否到底判定

遊戲結束判定

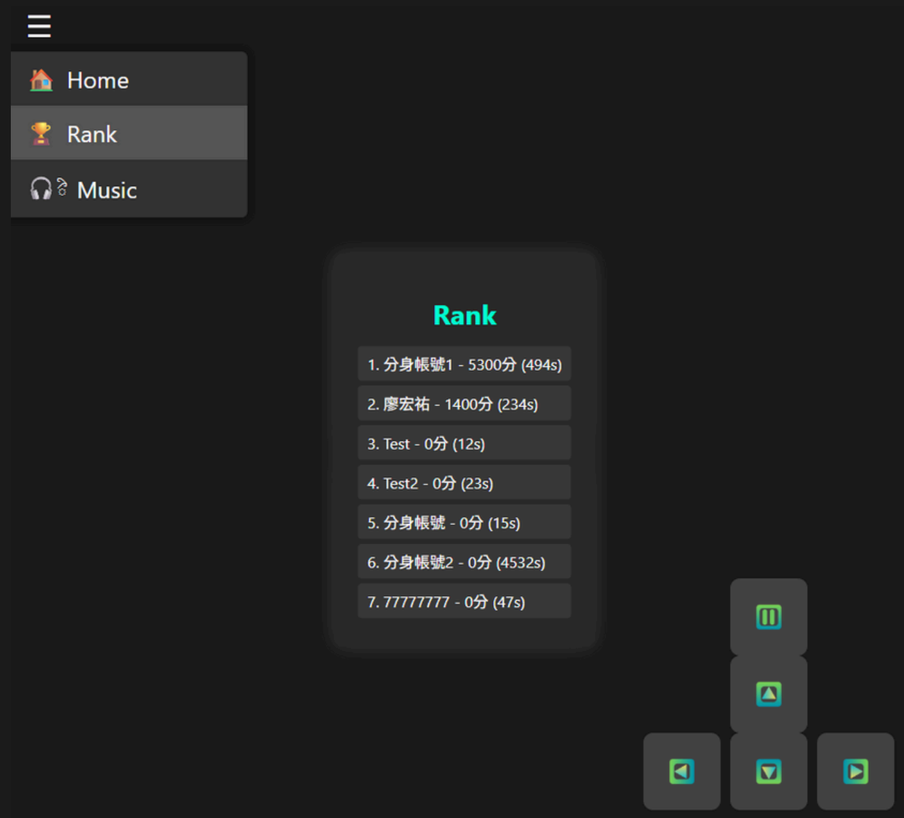
重新畫
Block、Grid

分數時間更新
判斷是否升級

```
281  function update() {
282      if(!paused){
283          if (!collision(currentShape, posX, posY + 1)) posY++;
284          else {
285              merge(currentShape, posX, posY);
286              clearLines();
287              currentShape = nextShape;
288              nextShape = randomShape();
289              drawShapesmall(nextShape, 0, 0, "#1F7F6C");
290              posX = 3;
291              posY = 0;
292              if (collision(currentShape, posX, posY)) {
293                const elapsedTime = Math.floor((Date.now() - startTime) / 1000);
294                playSound("gameover");
295                document.getElementById("gameOverPanel").classList.remove("hidden");
296                clearInterval(gameLoop);
297
298              }
299          }
300
301
302          context.fillStyle = "#000";
303          context.fillRect(0, 0, COLS, ROWS);
304          drawGrid();
305          for (let y = 0; y < ROWS; y++) {
306              for (let x = 0; x < COLS; x++) {
307              if (board[y][x]) drawBlock(x, y, "#1F7F6C");
308              }
309          }
310          drawShape(currentShape, posX, posY, "#1F7F6C");
311          document.getElementById("scoreDisplay").textContent = `Score: ${score}`;
312          const elapsedTime = Math.floor((Date.now() - startTime) / 1000);
313          document.getElementById("timerDisplay").textContent = `Time: ${elapsedTime}s`;
314          LevelUp();
315      }
316  }
```

saveScoreToRank()

從rank fetch資料




```
453  function saveScoreToRank(score, time, name) {
454    const isValid = /^[\u4e00-\u9fa5a-zA-Z0-9_]{1,15}$/.test(name);
455    if (!isValid) {
456      alert("⚠ 名字不合法！請使用中英文、數字或底線，最多15字");
457      return;
458    }
459
460
461    fetch('http://140.113.65.41:3000/rank')
462      .then(res => res.json())
463      .then(data => {
464        const nameExists = data.some(entry => entry.name === name);
465        if (nameExists) {
466          alert("此名字已經存在排行榜中，請換一個！");
467          return;
468        }
469
470        return fetch('http://140.113.65.41:3000/rank', {
471          method: 'POST',
472          headers: {
473            'Content-Type': 'application/json'
474          },
475          body: JSON.stringify({ name, score, time })
476        });
477      })
478      .then(res => res?.json?.())
479      .then(data => {
480        if (data) {
481          console.log("儲存成功", data);
482
483          gameElements.forEach(el => el.style.display = "none");
484          document.getElementById("gameOverPanel").classList.add("hidden");
485          showRank();
486          rankPanel.classList.remove("hidden");
487
488          setTimeout(() => {
```

showRank()

從rank fetch資料

```
328  function showRank() {
329    fetch('http://140.113.65.41:3000/rank')
330      .then(res => res.json())
331      .then(rankData => {
332        const rankList = document.getElementById("rankList");
333        rankList.innerHTML = "";
334        rankData.forEach((entry, index) => {
335          const li = document.createElement("li");
336          li.textContent = `${index + 1}. ${entry.name} - ${entry.score}分 (${entry.time}s)`
337          rankList.appendChild(li);
338        });
339      });
340  }
```

Node.js server

用express架構
require()

```
const express = require('express');
const fs = require('fs');
const cors = require('cors');
const bodyParser = require('body-parser');
const path = require('path');

const app = express();
const PORT = 3000;

app.use(cors());
app.use(bodyParser.json());

app.use(express.static(path.join(__dirname, 'public')));

// 排行榜 API
app.get('/rank', (req, res) => {
  const data = JSON.parse(fs.readFileSync('rank.json'));
  const top10 = data.sort((a, b) => b.score - a.score).slice(0, 10);
  res.json(top10);
});

app.post('/rank', (req, res) => {
  try {
    const newEntry = req.body;
    console.log("收到資料：", newEntry);

    const data = JSON.parse(fs.readFileSync('rank.json'));
    data.push(newEntry);
    fs.writeFileSync('rank.json', JSON.stringify(data, null, 2));
    res.status(201).json({ message: '儲存成功', entry: newEntry });
  } catch (err) {
    console.error("儲存失敗：", err);
    res.status(500).json({ message: '儲存失敗', error: err.message });
  }
});
```

排前10回傳
res.json(top10)

從網頁取得資料儲存

```
app.get('/', (req, res) => {
  res.sendFile(path.join(__dirname, 'public', 'index.html'));
});
app.use((req, res, next) => {
  console.log(`💡 收到請求：${req.method} ${req.url}`);
  next();
});
```

讓請求能夠繼續

http://140.113.65.41:3000/

The end