

Computer Engineering Essentials
Report

Quick Catch

Presented By

6632141921 Pongpitchaya Jaidee

6630154721 Theerach Sae-Ngow

6630306821 Visarut Sangsiri

2110221 Semester 1/2024

Department of Computer Engineering,
Faculty of Engineering, Chulalongkorn University

สารบัญ

Quick Catch	2
URL และวิธีใช้	3
Discussion regarding basic requirements	7
Discussion regarding challenging requirements	11
Project plan and actual execution	13

Quick Catch

Quick Catch เป็นเกมที่ผู้เล่นจะได้รับบทเป็นลิง โดยจะต้องพยายามจับหรือหลบหลีกของที่ตกลงมาเพื่อเพิ่มคะแนน นอกเหนือนี้ยังมีสกินน่ารักๆ ให้เลือกใช้เพื่อปรับแต่งตัวละคร และระบบ leaderboard ที่เพิ่มความท้าทายให้ผู้เล่นสามารถแข่งขันกับคนอื่นๆ เพื่อขึ้นสู่อันดับสูงสุดได้



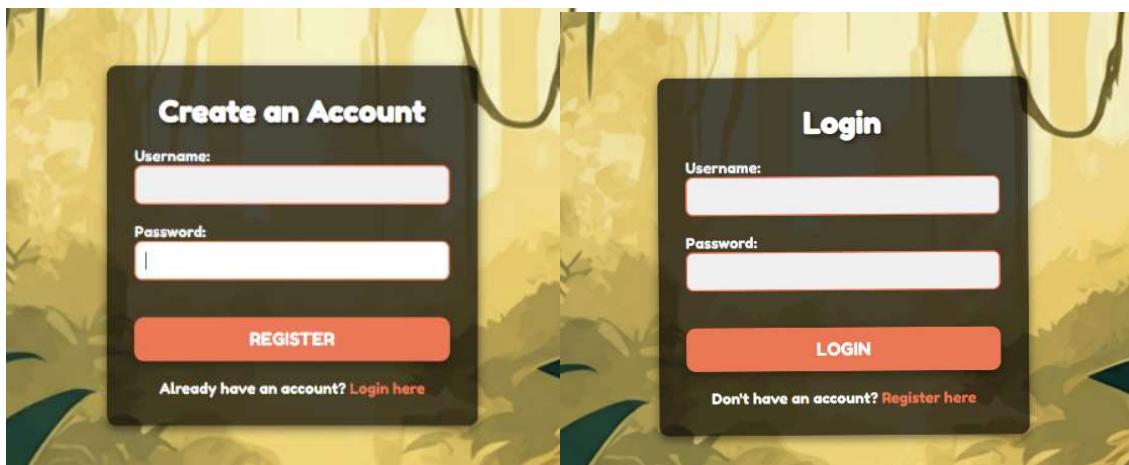
URL and How to play the game

URL

<http://44.195.238.84:3221/>

วิธีใช้งาน

1. เริ่มต้นการใช้งานด้วยการ login หรือ register โดย username ที่ใช้จะเป็นชื่อที่ขึ้นใน scoreboard

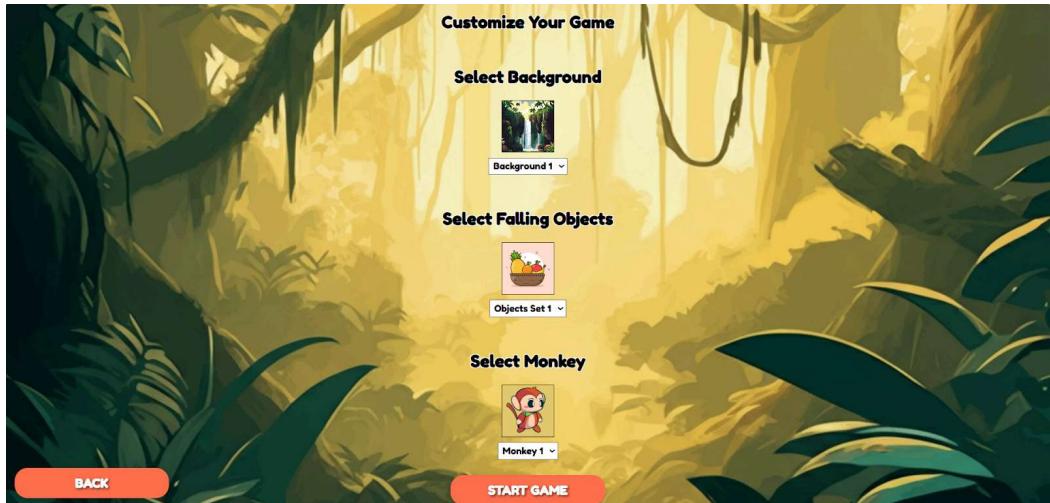


2. เมื่อกดเข้าจะเห็นหน้าเมนู ซึ่งมี 3 ปุ่ม ได้แก่ Play Scoreboard และ Logout



ถ้ากดปุ่ม Play จะเข้าสู่หน้าปรับแต่ง ปุ่ม ScoreBoard จะเข้าสู่หน้าจอแสดงอันดับ และ Logout จะกลับไปหน้า Login ใหม่

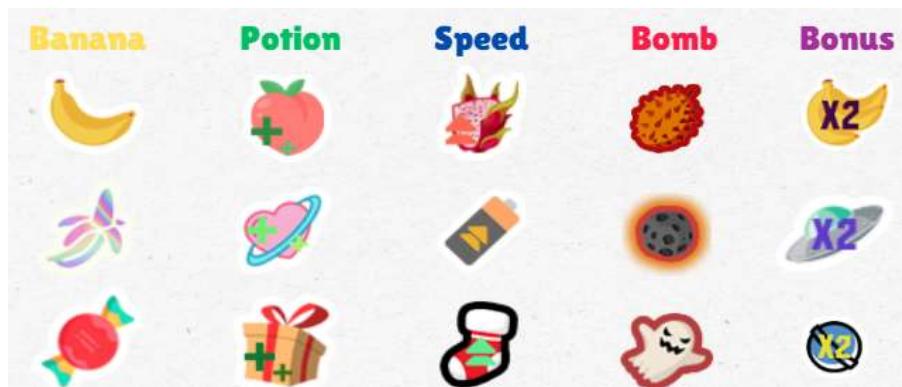
3. เมื่อกดปุ่ม Play จะแสดงหน้าตาปรับแต่ง skin ของวัตถุต่าง ๆ ในเกม ได้แก่ ภาพพื้นหลัง สีสันของ และผู้เล่น โดยแต่ละอย่างจะมีตัวเลือกให้ปรับแต่งได้อย่างละ 3 แบบ ซึ่งสามารถสมมูลสารได้ทั้งหมด 27 รูปแบบ



4. เมื่อกดปุ่ม start game จะแสดงหน้าจอเกมหลัก โดยผู้เล่นต้องบังคับตัวละครโดยใช้ปุ่มลูกศรซ้าย และขวา ให้ไปเก็บหรือหลบวัตถุที่หล่นลงมา เพื่อที่จะได้รับคะแนนเพิ่ม ยิ่งเวลาผ่านไปวัตถุจะหล่นลง มาเร็วขึ้น และมีจำนวนมากขึ้น



โดยวัตถุที่หล่นลงมาจะมีทั้งหมด 5 ประเภท ประกอบไปด้วย Banana Potion Speed Bomb และ Bonus ซึ่งเมื่อเก็บแต่ละอย่าง จะให้ผลที่แตกต่างกันดังนี้



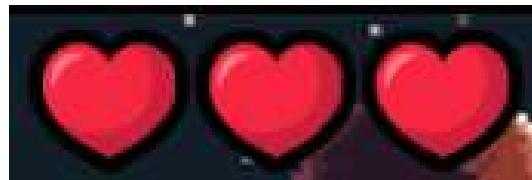
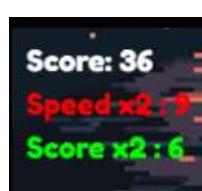
Banana - เมื่อเก็บ จะได้คะแนนเพิ่มขึ้น 1 แต้ม แต่ถ้าไม่เก็บ เลือดจะโดนลบ 1 แต้ม

Potion - เมื่อเก็บ จะได้เลือดเพิ่มขึ้น 1 แต้ม สูงสุด 3

Speed - 10 วินาทีหลังจากเก็บ ความเร็วของผู้เล่นจะเพิ่มขึ้น 2 เท่า

Bomb - เมื่อเก็บ เลือดจะลดลง 1 แต้ม

Bonus - 10 วินาทีหลังจากเก็บ คะแนนของ Banana จะเพิ่มขึ้น กลายเป็น 2



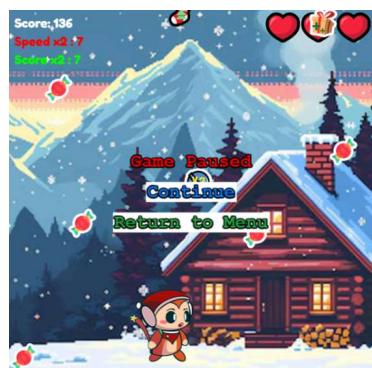
ด้านขวาบนของจอ จะแสดงจำนวนเลือด ณ ปัจจุบัน ซึ่งแสดงด้วยรูปหัวใจ ผู้เล่นจะมีเลือดอยู่ 3 แต้ม

เมื่อเลือด ลดลงเหลือ 0 เกมจะจบลงทันที

ด้านซ้ายบนของจอ แสดงแต้ม ณ ปัจจุบันที่มี แสดงระยะเวลาของ Speed Buff และ Bonus Buff

ตามลำดับ

นอกจากนี้ผู้เล่นยังสามารถกดปุ่ม ESC เพื่อหยุดเกม และเลือกได้ว่าจะเล่นต่อ หรือจบเกม



5. เมื่อจบเกม คะแนนที่เล่นได้จะถูกบันทึกไว้ในฐานข้อมูล ผู้เล่นสามารถเลือกได้ว่าจะเล่นต่อ โดยกดปุ่ม New Game หรือจะกลับไปหน้าเมนู โดยกดปุ่ม Return to Menu



6. ผู้เล่นสามารถกดที่ปุ่ม ScoreBoard เพื่อดู Hall of Fame ได้ โดยจะแสดง 10 user ที่เล่นแล้วได้คะแนนมากที่สุด

Top 10 High Scores		
Rank	Player Name	Score
1	mew	9999
2	jason36	100
3	henryjohn	96
4	kennedyshannan	80
5	faulkneramy	67
6	perezjoseph	65
7	devidtimothy	64
8	hellmonique	40
9	archerpaul	34
10	richardgilbert	15

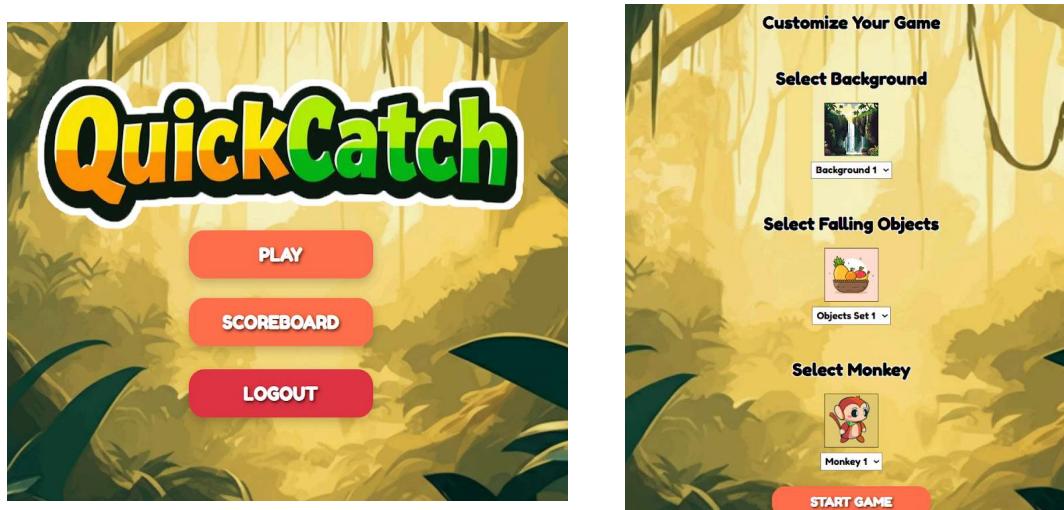
[Back to Game](#)

Discussion regarding basic requirements

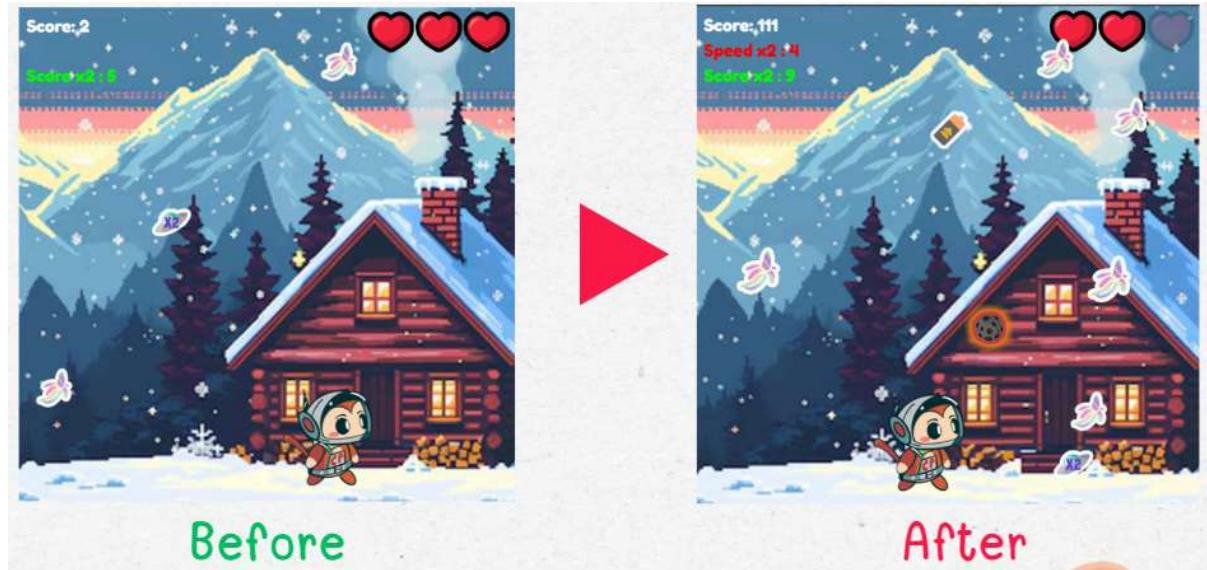
- พัฒนาระบบการยืนยันตัวตนของผู้ใช้งาน เพื่อให้สามารถ login หรือ register โดยใช้ชื่อผู้ใช้ที่จะแสดงบนกระดานคะแนน (Scoreboard)
- พัฒนาระบบที่สามารถบันทึกคะแนนสูงสุด และแสดง 10 อันดับแรกของผู้เล่นที่ได้คะแนนสูงสุด



- พัฒนาเมนูที่มีตัวเลือก เช่น Play, Scoreboard, Logout ออกแบบ UI ที่ใช้งานง่ายและดึงดูดสำหรับการเล่นเกม รวมถึงฟีเจอร์การปรับแต่งตัวละคร (Skins) และควบคุมที่เข้าใจง่าย



- พัฒนาให้ตัวละครเคลื่อนไหวอย่างราบรื่น และสามารถตอบสนองความบันเทิงได้ด้วยลูกเล่นต่างในเกม เช่น ระเบิดที่เก็บแล้วเลือดลด และ Object ตกรากขึ้นตาม Score ที่มากขึ้น



- มีการใช้เครื่องมือพื้นฐานคือ Html CSS Javascript ใน Frontend และ Express, Mongoose (MongoDB), CORS ใน Backend ไม่มี library หรือ framework ใด ๆ เพิ่มเติม
- สามารถทำตามแผนที่วางไว้และเว็บไซต์ออกแบบมาสมบูรณ์ตามต้องการ

Project Structure (GitHub: https://github.com/BuFFaLoNoR/Project_Com_Eng_Ess.git)

```

public/      // Front-end Assets
  -assets/
    -back/
    -monkey/
    -prev/
    -set1/
    -set2/
    -set3/
    -sound/
    -menu_back.png
    -menu_back2.png
    -menu_back3.png
    -title.png
  -scripts/
    -api.js
    -game.js
    -highscores.js
    -main.js
    -menu.js
    -server.js
  -styles.css

```

```
-front.html  
-login.html  
-main.html  
-menu.html  
-register.html  
-table.html  
-package.json  
-package-lock.json
```

```
src/      //backend  
-models/  
    -Highscore.js  
    -User.js  
-routes/  
    -authRoutes.js  
    -highscoreRoutes.js  
-utils/  
    -passwordUtils.js  
-config/  
    -db.js  
-app.js  
-server.js  
-package.json  
-package-lock.json  
  
.package.json  
.package-lock.json
```

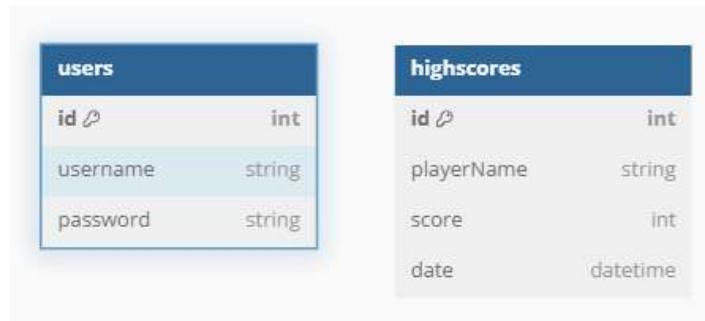
- The back-end of the web application use only Node.js, Express.js and MongoDB

MongoDB

API List

Method	Endpoint	Detail
--------	----------	--------

Authentication API		
Post	/check-username	Check if a username is available for registration.
Post	/register	Register a new user.
Post	/login	Authenticate user login.
Leaderboard API		
Get	/	#pathถูกตั้งไว้ในapp.jsแล้ว
Post	/	#pathถูกตั้งไว้ในapp.jsแล้ว

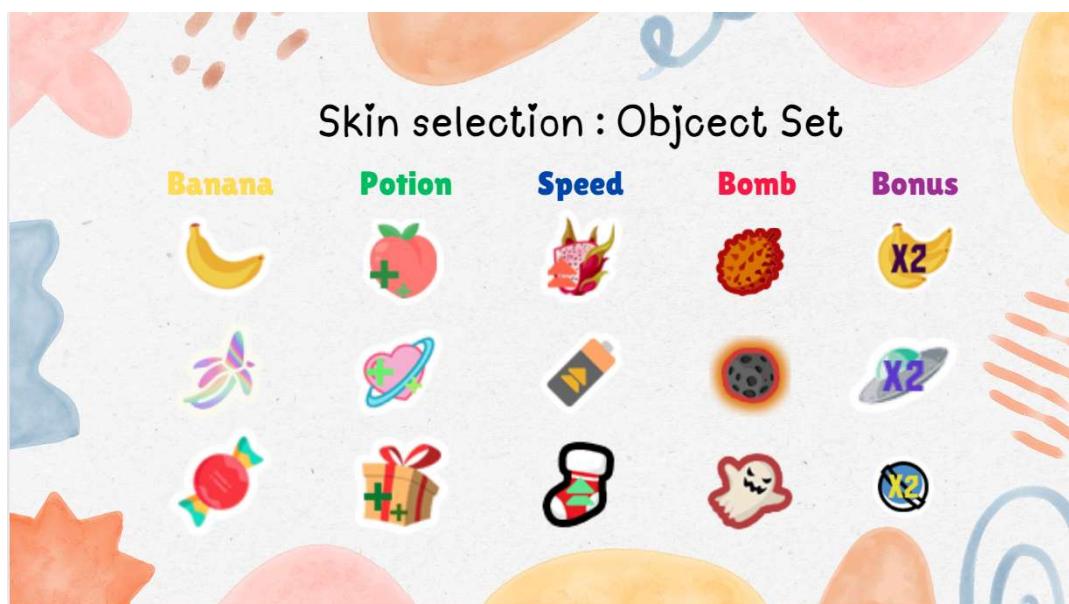


Discussion regarding Challenging Requirements

- การรวม Front-end และ Back-end ให้ทำงานร่วมกันอย่างไร้รอยต่อ เช่น การเชื่อมต่อ API เพื่อจัดการข้อมูลผู้ใช้และคะแนน
- การทดสอบ End-to-End (e2e) และการ Deploy ระบบบน AWS EC2 เพื่อให้สามารถใช้งานได้จริง
- การเพิ่มลูกเล่น เช่น วัตถุแบบสุ่ม (Randomized Items), ระบบบัฟที่เปลี่ยนแปลงได้ตามสถานการณ์ในเกม และ ระบบหยุดเกม



- มีการใช้ Animation ในเกมทำให้ได้การเคลื่อนไหวของเกม, ความหลากหลายของ Object ต่างๆ
- การออกแบบ Animation และกราฟิกที่ไม่สามารถใช้ AI สร้างได้ ต้องพัฒนาเองเพื่อเพิ่มความน่าสนใจ





- ระบบการเล่นเข้าใจง่าย สามารถเล่นได้ง่ายบนหลายอุปกรณ์

Project plan and actual execution

1. Project Plan

- 14 Nov ทดลองกันว่าจะทำอะไร จบทอกมาเป็น QuickCatch แบบคร่าว ๆ วางแผนการในขั้นต่อไป และให้แต่ละคนไปหาข้อมูลเพิ่มเติมมา ว่าอย่างจะเพิ่มฟังก์ชันอะไรในเกม
 - 20 Nov เคาะใจเดีย QuickCatch ทั้งหมด และแบ่งงาน Front-end และ Back-end แล้วเริ่มทำงานตามที่ list ไว้
 - 23-26 Nov ทำ Front-end ของ web พัฒนาฟังก์ชันและลูกเล่นต่าง ๆ ของ web และทำกราฟิกต่าง ๆ เช่นภาพประกอบ และอนิเมชัน
 - 27-30 Nov นำข้อมูล Front-end ที่ได้มาทำ Back-end เพิ่ม เพื่อเพิ่มข้อมูลหลังบ้านของผู้ใช้งาน 1-2 Nov แก้ Front-end บางส่วนที่ไม่สอดคล้องกับ Back-end ที่ได้ทำมา และทำ Back-end เพิ่ม
 - 1-2 Dec แก้ Front-end บางส่วนที่ไม่สอดคล้อง Back-end ที่ได้ทำมาและทำ Back-end เพิ่ม
 - 3 Dec รวม Front-end และ Back-end แบบ Final เข้าด้วยกันแล้ว test ระบบพร้อมแก้ปัญหาต่างๆ

2. Project Execute

2.1 Front-end Development

- สร้าง Directory public บน Repository
- ออกแบบหน้าจอเกม, UI ต่างๆ และระบบต่างๆ ในเกม
- จัดภาพ และเสียงประกอบจากสื่อต่าง ๆ
- เริ่มทำการ POC และ Implement ตัวเกมหลัก
- Implement graphic และ animation ต่างๆ ที่ AI ไม่สามารถ generate ได้
- ตกแต่ง UI ตาราง และปุ่มต่าง ๆ ในเกม
- นำเกมและ animation มาประกอบกัน

2.2 Back-end Development

- สร้าง repository บน Github
- สร้าง database cluster บน MongoDB Atlas
- พัฒนา api และ route handler สำหรับ backend และ frontend
- API smoke testing ด้วย Postman
- deploy โปรแกรมบน EC2