

**เอกสารการออกแบบเกม**

**Game Design Document**

ชื่อเกม (ภาษาไทย) : นักผจญบรรจุภัณฑ์

ชื่อเกม (ภาษาอังกฤษ) : Adventurer Packing

สมาชิก : นายวัชกร บุตร์ดีวงษ์

|  |  |  |  |
| --- | --- | --- | --- |
| รุ่น | วันที่ | ผู้เขียน | คำอธิบาย |
| 1.0 | 28-12-2564 | watchakorn-18k | ปล่อยตัวเกมลงใน github.com ที่ https://github.com/watchakorn-18k/Adventurer-Packing-Game |
| 1.1 |  |  |  |

คำอธิบาย

เป็นเกมเกี่ยวกับการส่งของให้ผู้เล่นส่งกล่องพัสดุที่อยู่กระจายในเมืองโดยตัวละครหรือผู้เล่นจะขี่ควายไล่เก็บพัสดุไปส่งของตามจุดที่ระบุไว้ โดยมีเวลาเป็นตัวกำหนดหากหมดเวลาเกมจะจบลง หากทำสำเร็จจะทำภารกิจสำเร็จ แรงบันดาลใจได้มาจากการพบเจอหนักงานมารับพัสดุตามบ้านแล้วเอาไปจัดส่งให้ที่ศูนย์หลัก เป็นเกม 2มิติ ใช้อุปกรณ์ คีย์บอร์ดในการเล่น

วัตถุประสงค์ของเกม

1. เพื่อศึกษาคำสั่ง อง๕ประกอบต่างๆของ Engine Game Unity3D

2. เพื่อฝึกฝนทักษะการคิดเชิงตรรกะ

3. เพื่อเสริมทักษะในการสร้างเกม

4. เพื่อทำความเข้าใจเกี่ยวกับการทำงานของเกมและการเขียนโปรแกรมเกม

5. เพื่อศึกษาและฝึกฝนเขียนภาษา C#

ความเป็นมาและความสำคัญของเกม

เป็นเกมเกี่ยวกับการส่งของให้ผู้เล่นส่งกล่องพัสดุที่อยู่กระจายในเมืองโดยตัวละครหรือผู้เล่นจะขี่ควายไล่เก็บพัสดุไปส่งของตามจุดที่ระบุไว้ โดยมีเวลาเป็นตัวกำหนดหากหมดเวลาเกมจะจบลง หากทำสำเร็จจะทำภารกิจสำเร็จ แรงบันดาลใจได้มาจากการพบเจอหนักงานมารับพัสดุตามบ้านแล้วเอาไปจัดส่งให้ที่ศูนย์หลัก เป็นเกม 2มิติ ใช้อุปกรณ์ คีย์บอร์ดในการเล่นโดยสามารถบังโดยการกดปุ่ม Arrow Up Down Left Right เป็นเกมประเภท Simulation Game เทคนิคในการใช้การพัฒนาส่วนใหญ่จะใช้คำสั่งพื้นฐานง่ายๆของ Unity อย่างเล่น การเช็คว่าผู้เล่นชนอะไร หรือผู้เล่นเข้ามาจุดที่ต้องหรือไม่ โดยส่วนใหญ่โค้ดโปรแกรมจะใช้ ตัวแปรประเภท Bool

กลุ่มเป้าหมาย (พัฒนาเพื่อใคร เพศ อายุ สถานที่ตั้ง อื่นๆ)

พัฒนาเพื่อเด็กตั้งแต่อายุ 6 ปีขึ้นไป เล่นได้ทุกเพศทุกวัย เหมาะสำหรับคนไทยเพราะตัวเกมเป็นภาษาไทย

เนื้อเรื่อง (ตั้งแต่ต้นจนจบเกม)

เรื่องราวของพนักงานตกอับที่ถูกไล่ออกจากบริษัทอย่างกระทันหันเขาจึงตัดสินใจมาเป็นพนักงานส่งของแต่ ทว่า ความฝันของเขายังหางไกลเกินไป เพราะตัวเขาเองไม่มีเงินพอที่จะซื้อรถในการขนส่งของได้ ตัวเขาเองจึงคิดได้ว่าตัวเองมีควายอยู่ที่บ้านตัวนึงเขาจึงตัดสินเอาควายตัวนั้นมาขี่แทนรถส่งของและพร้อมที่จะรับมือกับบรรจุภัณฑ์จากลูกค้า เพื่อที่จะเป็น นักผญจบรรจุภัณฑ์ให้ได้

วิธีการเล่นเกม

กดปุ่ม Up เพื่อเดินหน้า

กดปุ่ม Down เพื่อถอยหลัง

กดปุ่ม Right เพื่อเลี้ยวขวา

กดปุ่ม Left เพื่อเลี้ยวซ้าย

มาตราส่วน (ระบุความละเอียดหน้าจอที่ใช้พัฒนา)

16:9

เครื่องมือและโปรแกรมที่ใช้ในการพัฒนา

Unity3D

Photoshop

ตัวละคร

|  |  |  |
| --- | --- | --- |
| ภาพ | ชื่อ | คุณสมบัติ |
|  | tooy | ผู้เล่น |

ศัตรู

|  |  |  |
| --- | --- | --- |
| ภาพ | ชื่อ | คุณสมบัติ |
|  | package | กล่องพัสดุ |

อุปสรรค

|  |  |  |
| --- | --- | --- |
| ภาพ | ชื่อ | คุณสมบัติ |
|  | Timer | ตัวจับเวลา |

วัตถุอื่นๆ

|  |  |  |
| --- | --- | --- |
| ภาพ | ชื่อ | คุณสมบัติ |
|  | Barrier | ประกอบฉาก |
|  | BgClock | ภาพแสดง |
|  | BgScore | ส่วนต่อประสานกับผู้ใช้ |
|  | BgTask | ส่วนต่อประสานกับผู้ใช้ |
|  | Boost | วัตถุ |
|  | Bridge | ประกอบฉาก |
|  | Bus | ประกอบฉาก |
|  | Car 1 | ประกอบฉาก |
|  | Car 2 | ประกอบฉาก |
|  | Car 3 | ประกอบฉาก |
|  | Corner 1 | ประกอบฉาก |
|  | Corner 2 | ประกอบฉาก |
|  | Corner 3 | ประกอบฉาก |
|  | Corner 4 | ประกอบฉาก |
|  | Road 1 | ประกอบฉาก |
|  | Road 2 | ประกอบฉาก |
|  | Road 3 | ประกอบฉาก |
|  | Road 4 | ประกอบฉาก |
|  | Rocks 1 | ประกอบฉาก |
|  | Stadium | ประกอบฉาก |
|  | Tree 1 | ประกอบฉาก |
|  | Tree 2 | ประกอบฉาก |
|  | Tree 3 | ประกอบฉาก |
|  | Tree 4 | ประกอบฉาก |
|  | UI\_Finish\_text | ส่วนต่อประสานกับผู้ใช้ |
|  | UI\_package | ส่วนต่อประสานกับผู้ใช้ |
|  | UI\_placement | เป้าหมาย |
|  | Curve 1 | ประกอบฉาก |
|  | Curve 2 | ประกอบฉาก |
|  | FoundTain | ประกอบฉาก |
|  | GameOver\_title | ส่วนต่อประสานกับผู้ใช้ |
|  | House 1 | ประกอบฉาก |
|  | House 2 | ประกอบฉาก |
|  | House 3 | ประกอบฉาก |
|  | Intersection 1 | ประกอบฉาก |
|  | Intersection 2 | ประกอบฉาก |
|  | LogoGame | ส่วนต่อประสานกับผู้ใช้ |
|  | ParkCar | ประกอบฉาก |
|  | placement | ส่วนต่อประสานกับผู้ใช้ |
|  | Pool | ประกอบฉาก |
|  | UiRight | ส่วนต่อประสานกับผู้ใช้ |
|  | หยุดชั่วคราว | ส่วนต่อประสานกับผู้ใช้ |
|  | SYMB\_REPLAY | ส่วนต่อประสานกับผู้ใช้ |
|  | cursor | ส่วนต่อประสานกับผู้ใช้ |
|  | BTN\_ORANGE\_CIRCLE\_OUT | ส่วนต่อประสานกับผู้ใช้ |
|  | BTN\_ORANGE\_CIRCLE\_IN | ส่วนต่อประสานกับผู้ใช้ |
|  | BackgroundMenu | ส่วนต่อประสานกับผู้ใช้ |
|  | Clock | ส่วนต่อประสานกับผู้ใช้ |

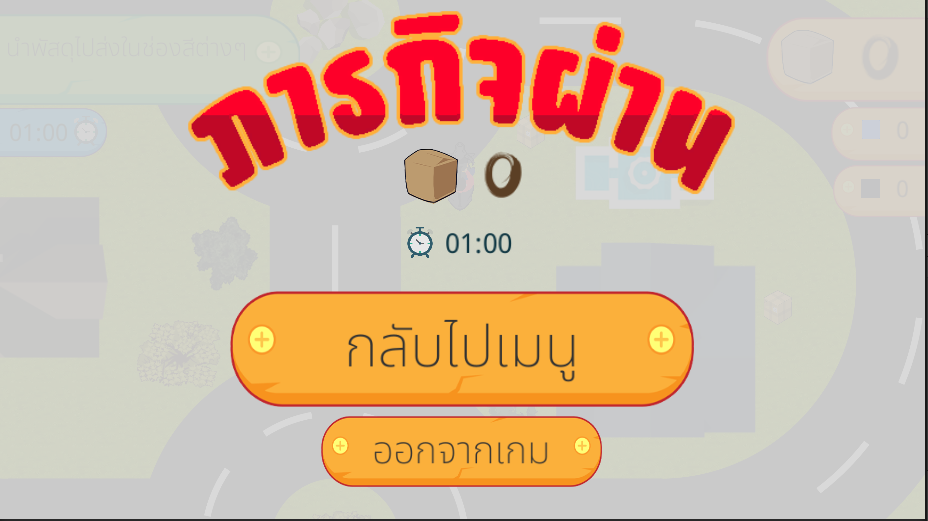
Flowchart

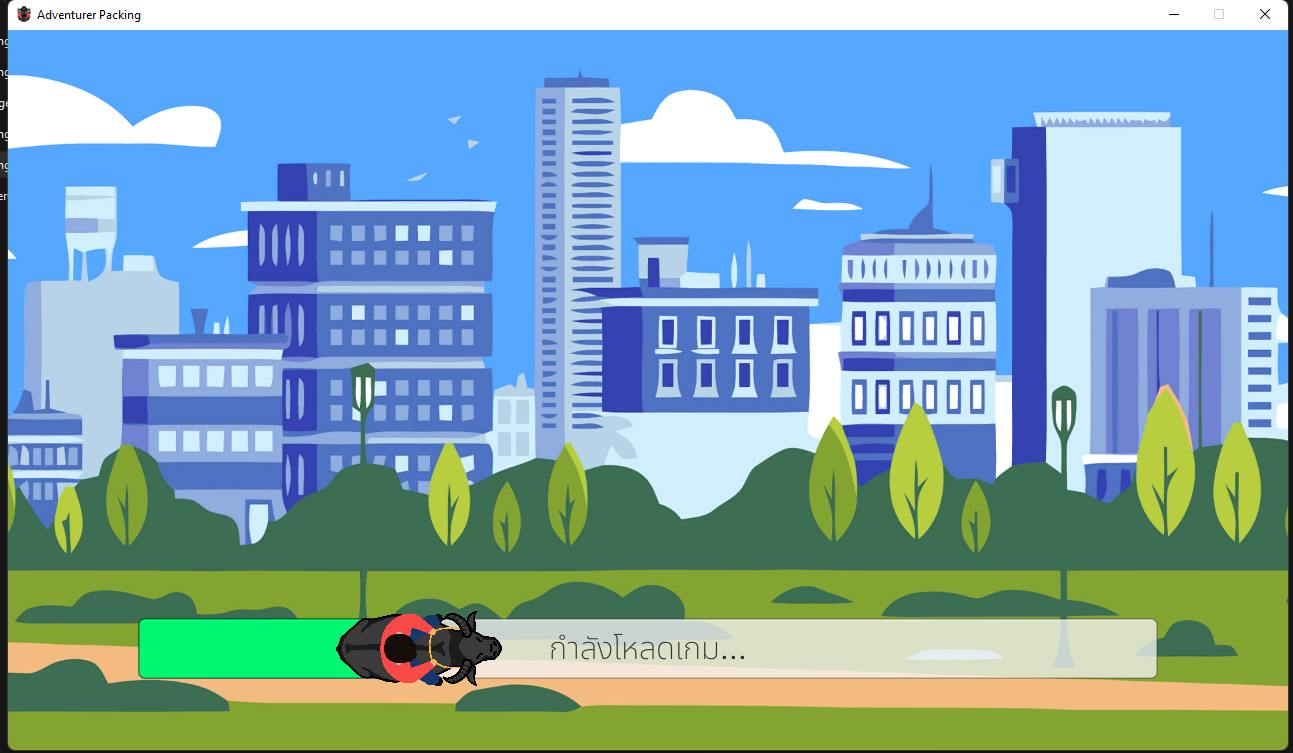




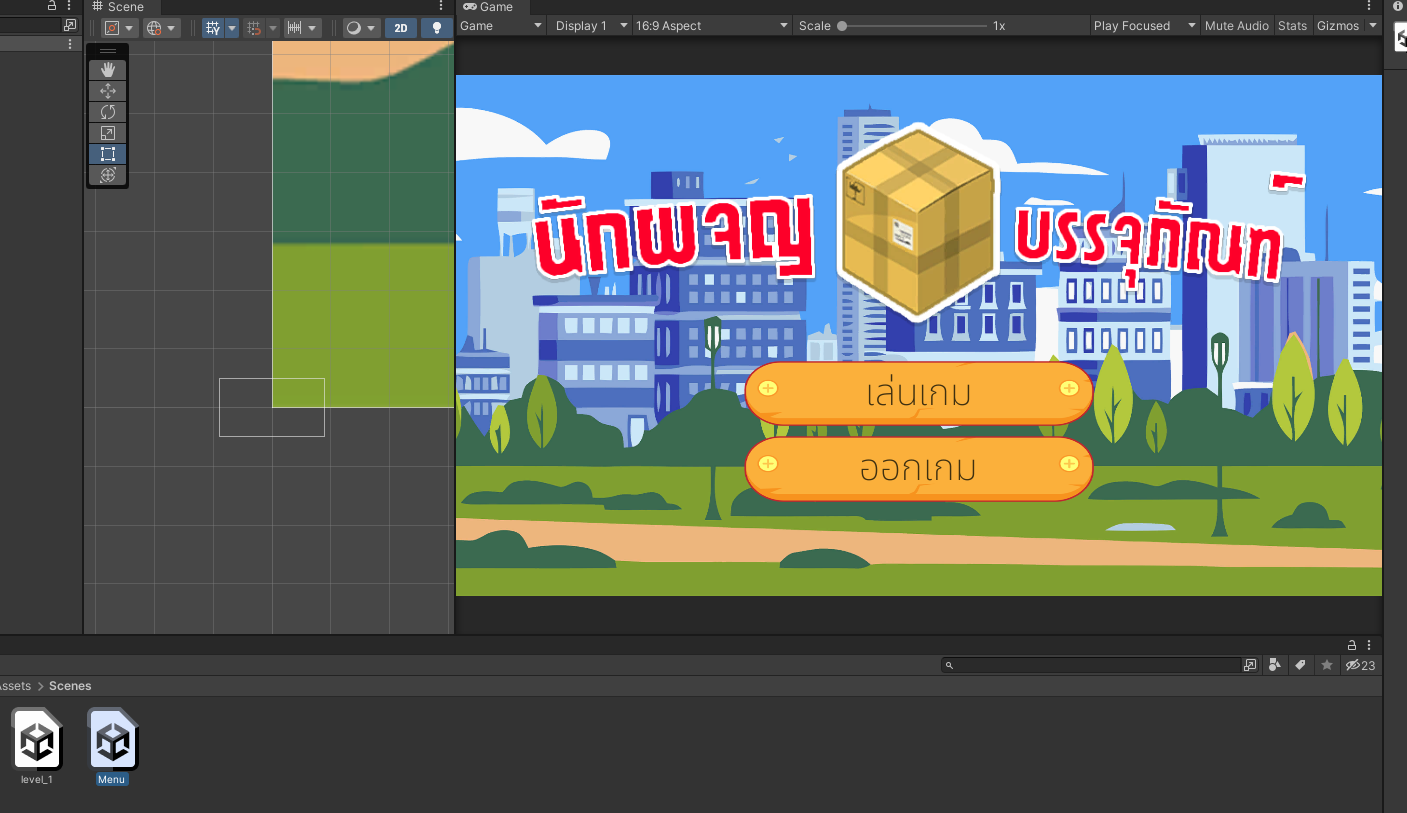


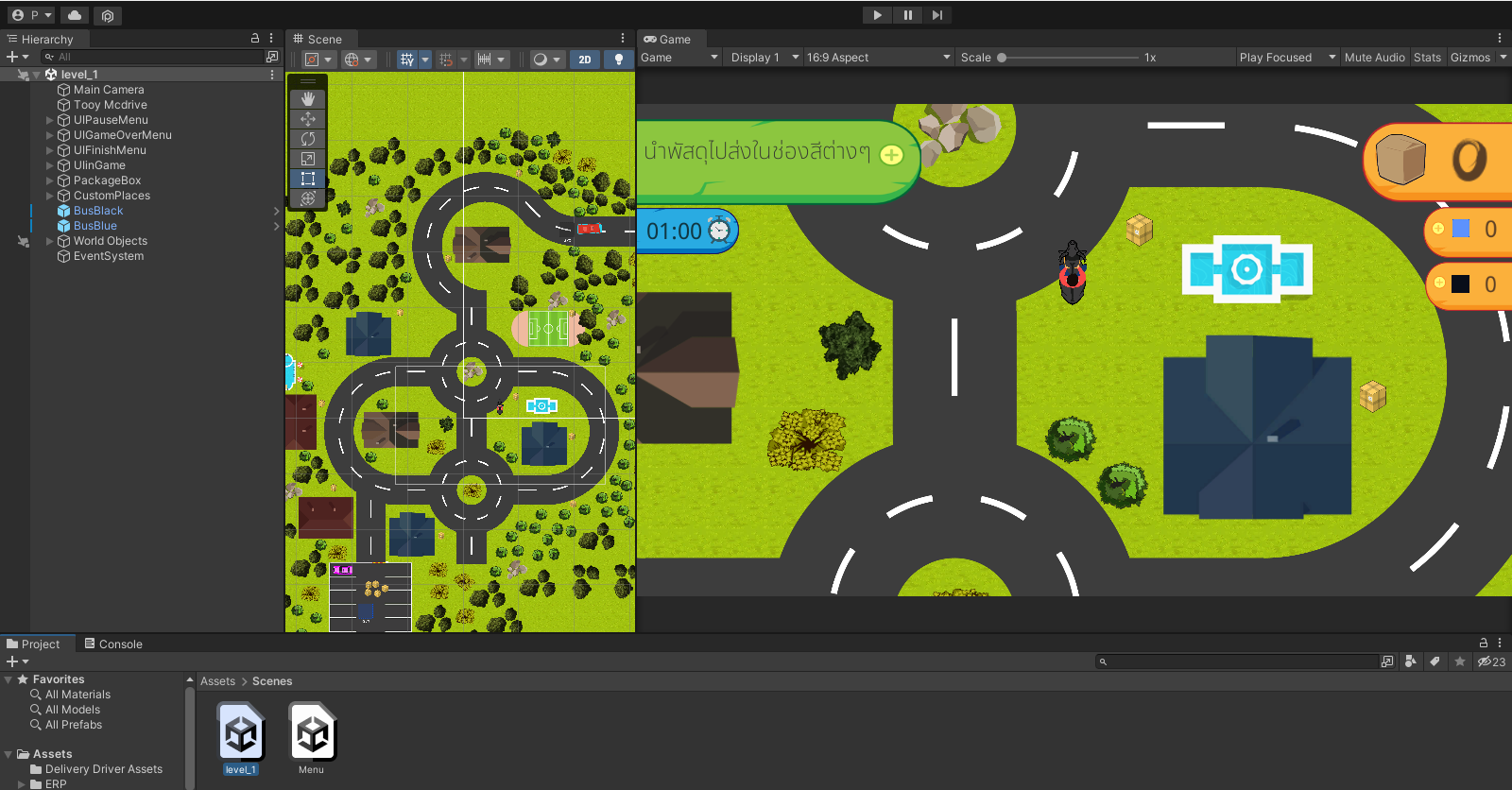




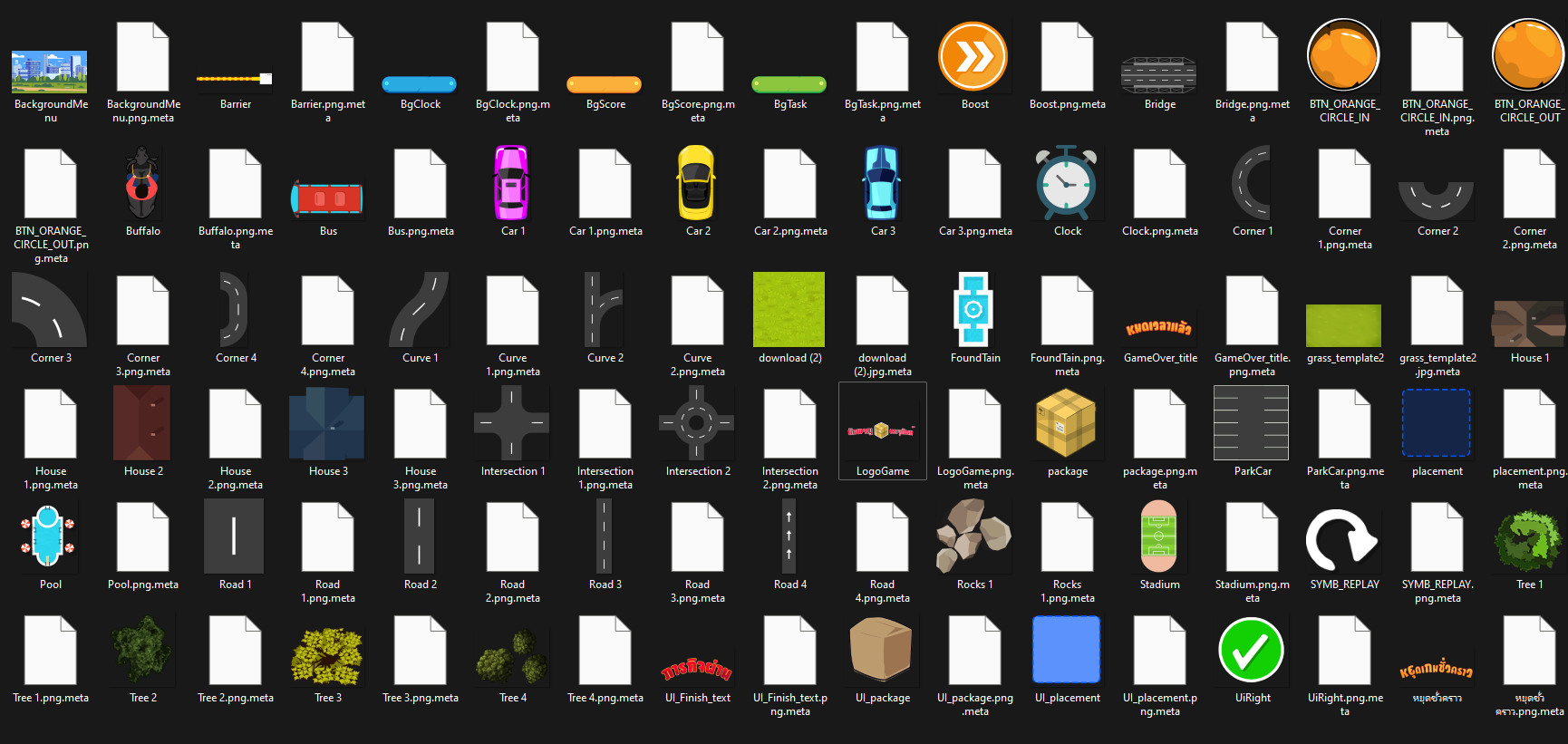


การออกแบบ Storyboard (หน้าจอทั้งหมดที่มีในเกม)





ภาพประกอบ/สไปรต์ชีต



เพลงประกอบ/เสียงเอฟเฟ็กต์

|  |  |  |
| --- | --- | --- |
| ชื่อไฟล์ | ตำแหน่งเฟรม | คำอธิบาย |
|  | Scene Menu | เสียงพื้นหลัง |
|  | Scene Level 1 | เสียงประกอบ |
|  | Scene Level 1 | เสียงประกอบ |
|  | Scene Level 1 | เสียงประกอบ |
|  | Scene Level 1 | เสียงประกอบ |
|  | Scene Level 1 | เสียงประกอบ |
|  | Scene Level 1 | เสียงประกอบ |
|  | Scene Level 1 | เสียงประกอบ |
|  | Scene Loading | เสียงพื้นหลัง |
|  | Scene Level 1 | เสียงพื้นหลัง |

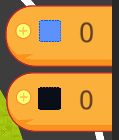
คะแนน

ได้มาจากหลังส่งพัสดุตามจุดที่กำหนดครบแล้วจะได้มาครั้งละ 1 คะแนน

คะแนน



จุดที่กำหนด





เงื่อนไขการชนะ

ส่งพัสดุจบครบก่อนเวลาหมด

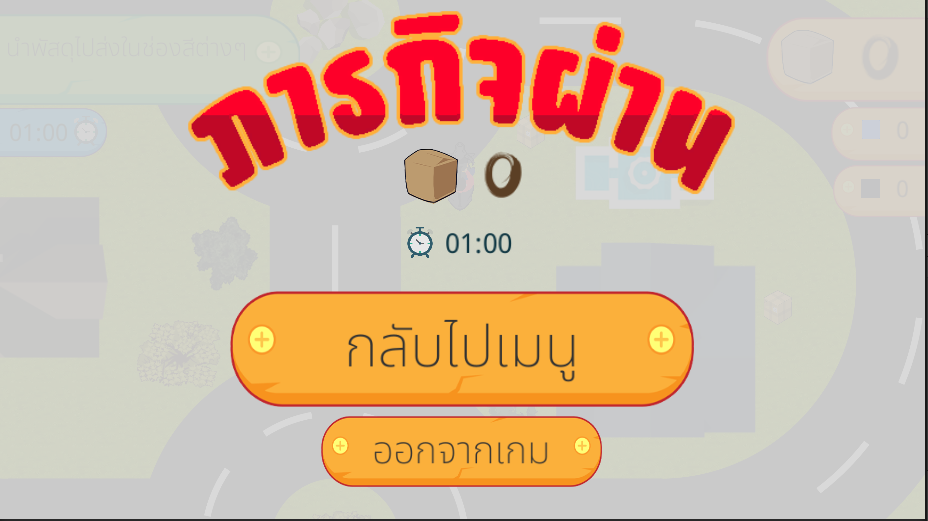
ผลการดำเนินงาน (บันทึกหน้าจอเกมจริงทั้งหมดที่มีในเกม)

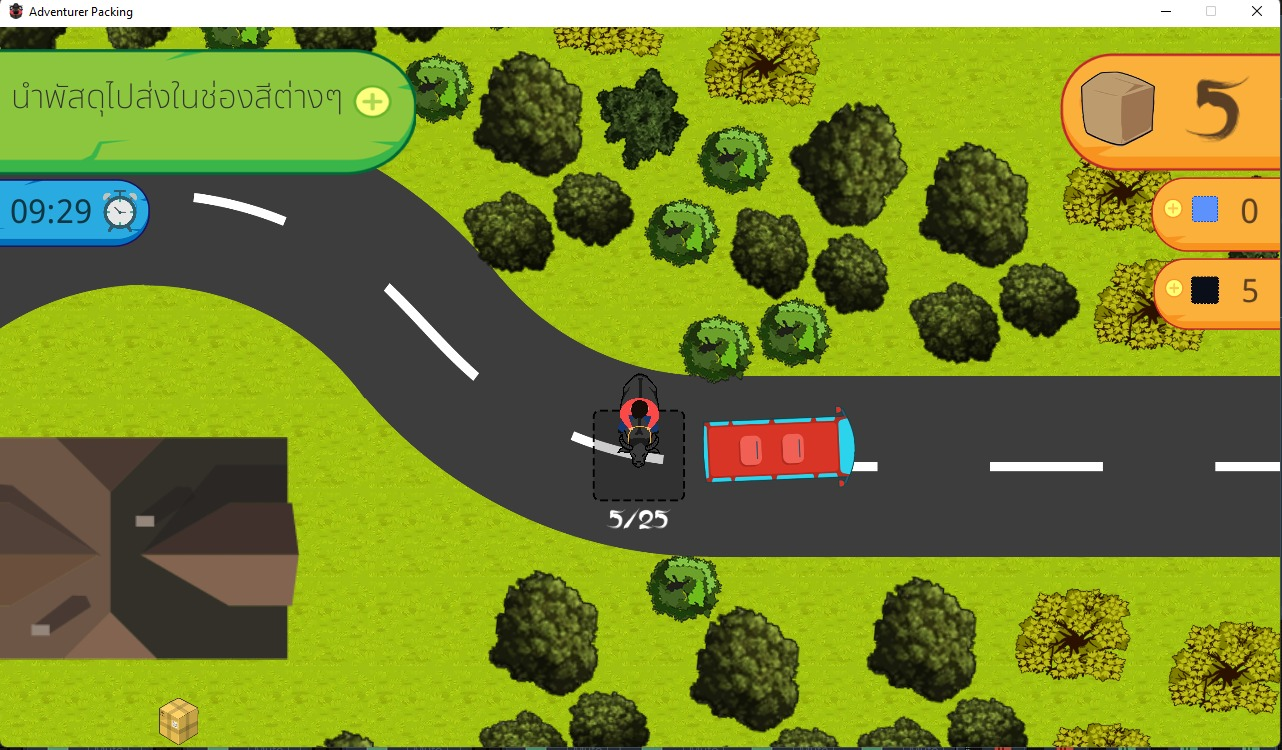


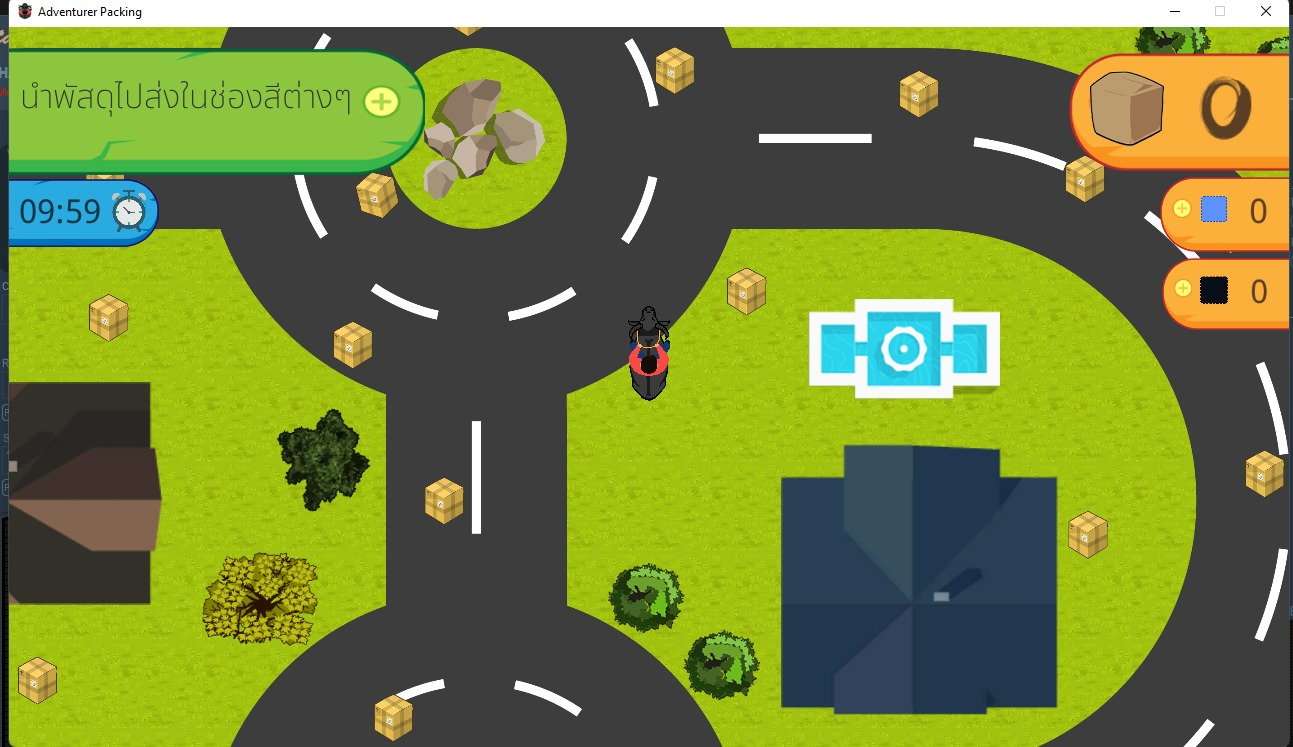












โปรแกรมการทำงาน (อธิบายโค้ดบางส่วนที่สำคัญ)

คำสั่งเคลื่อนย้ายตัวละคร

        steerAmout = Input.GetAxis("Horizontal") \* steerSpeed \*Time.deltaTime;

        moveAmout = Input.GetAxis("Vertical") \* moveSpeed \* Time.deltaTime;

        transform.Rotate(0, 0, -steerAmout);

        transform.Translate(0, moveAmout, 0);

steerAmout เป็นตัวแปรที่เก็บเป็น float โดยรับค่าจากแป้นพิมพ์ หากกดให้หมวดหมู่ แนวนอน ก็จะไปคูณกับจำนวน ความของ steer ก็คือ steerSpeed แล้วคูณ Time.deltaTime ในส่วนนี้การทำให้ความเร็วในเกมเหมือนกันทุกเครื่อง เพราะคอมพิวเตอร์เร็วไม่เท่ากันดังนั้นจริงทำให้มันเป็นมาตรฐาน

moveAmout เป็นตัวแปรที่เก็บเป็น float โดยรับค่าจากแป้นพิมพ์ หากกดให้หมวดหมู่ แนวตั้ง ก็จะไปคูณกับจำนวน ความของ move ก็คือ moveSpeed แล้วคูณ Time.deltaTime ในส่วนนี้การทำให้ความเร็วในเกมเหมือนกันทุกเครื่อง เพราะคอมพิวเตอร์เร็วไม่เท่ากันดังนั้นจริงทำให้มันเป็นมาตรฐาน

transform.Rotate(0, 0, -steerAmout); คำสั่ง transform.Rotate คือคำสั่งที่ใช้ในการหมุน gameobject โดยจะรับอาร์กิวเมน 3 ตัวคือ x , y , z รับค่าเป็น float ในส่วนตรงนี้ต้องการให้ แกน z หมุน โดยค่าเดิมของ steerAmout หากผู้ใช้กดปุ่ม Arrow left กับ Arrow right ทิศทางจะสลับกัน ดังนั้นจึงแก้ปัญหานี้โดยการให้ -steerAmout หาก steerAmout มีค่าเป็นลบก็จะให้มันมันบวก เพราะ ลบ บวก ลบ จะเท่ากับ + แต่ถ้า บวก กับ ลบ จะให้เป็น ลบ เพื่อให้ทิศทางในการหมุนเป็นอย่างที่มันควรเป็น