

# Programming clinic

Group ID60-07

สมาชิก

1. ศุภชาติ นาคจันทร์ รหัส 60130500113
2. ณรงค์ฤทธิ์ เกษร รหัส 60130500119
3. นายธนกร เลิศสุดคณีง รหัส 60130500123

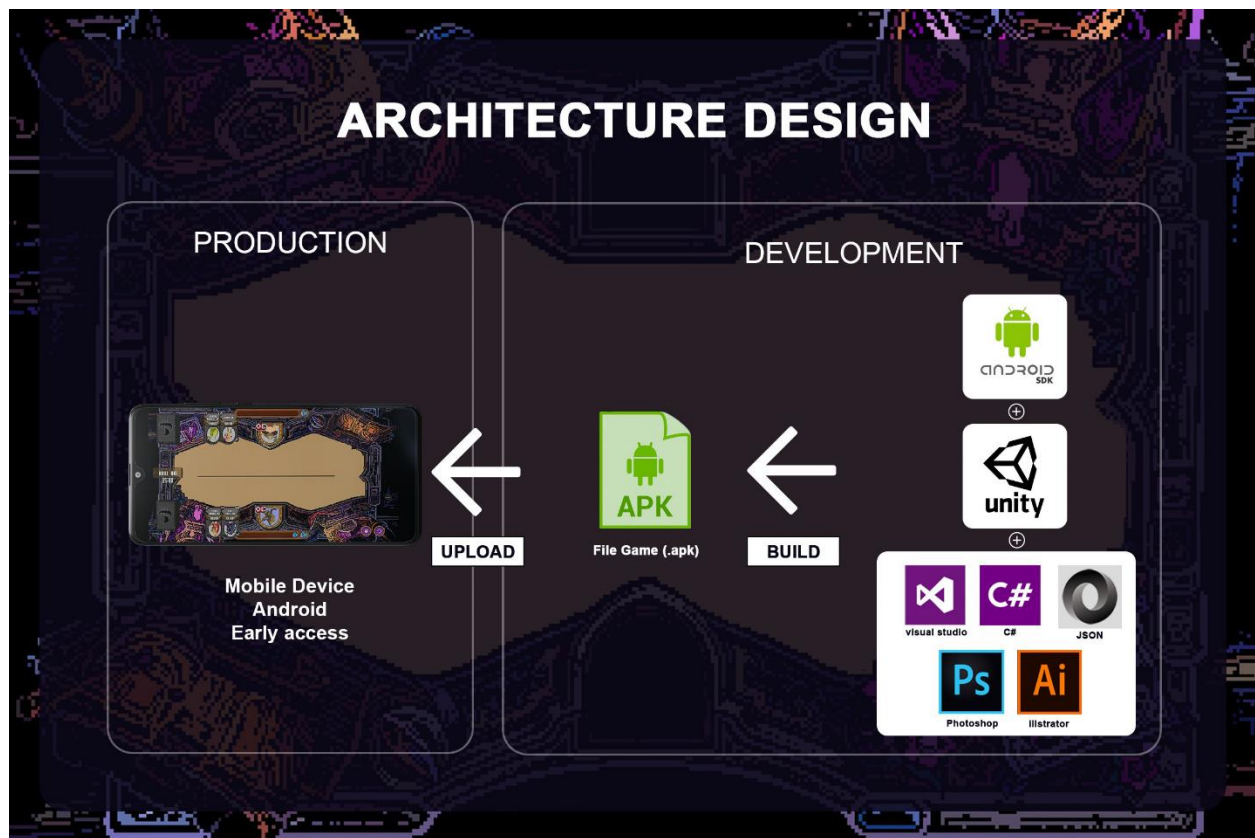
Requirement (ปัญหา) ของโครงการที่ทำได้จริง

- ผู้ใช้สามารถทายคำศัพท์ผ่านการสลับคำศัพท์ได้
- ผู้ใช้สามารถสลับการ์ดตัวละครบนมือได้
- ผู้ใช้สามารถนำการ์ดตัวละครลงบนสนามได้
- ผู้ใช้สามารถต่อสู้กับตัวละครบอทได้

รายการ Feature (บริการต่าง ๆ ของระบบหรือแอปพลิเคชันที่ไปแก้ปัญหาในข้อ 1) ที่ทำได้จริง

- Feature ทายคำศัพท์ผ่านการสลับตัวอักษร
- Feature วางการ์ดตัวละครลงบนสนาม
- Feature การต่อสู้กับบอท
- Feature เวลาในการเล่นแต่ละรอบ

แผนภาพเชิงระบบที่ประกอบไปด้วยการทำงาน เช่น module/package/component/service ใดบ้าง พร้อมคำอธิบาย



**Game engine:** unity version 2019.4.7f1 ที่เลือกในการสร้าง เพราะเป็นengineที่ได้รับความนิยมสูง สามารถเข้าถึงได้ง่าย

**Language:** C#, json C#เป็นภาษาพื้นฐานในการสร้างเกมของunity และjsonเป็นภาษาในการเขียนไฟล์databaseของfirebase

**Develop Program:** visual studio 2019 เลือกใช้โปรแกรมนี้เพราะมีฟังก์ชันรองรับ game development

**Version Control:** git hub, Unity collaboration เป็นserviceของทางunity เพื่อให้สามารถทำงานเป็นทีมได้อย่างสะดวกขึ้น

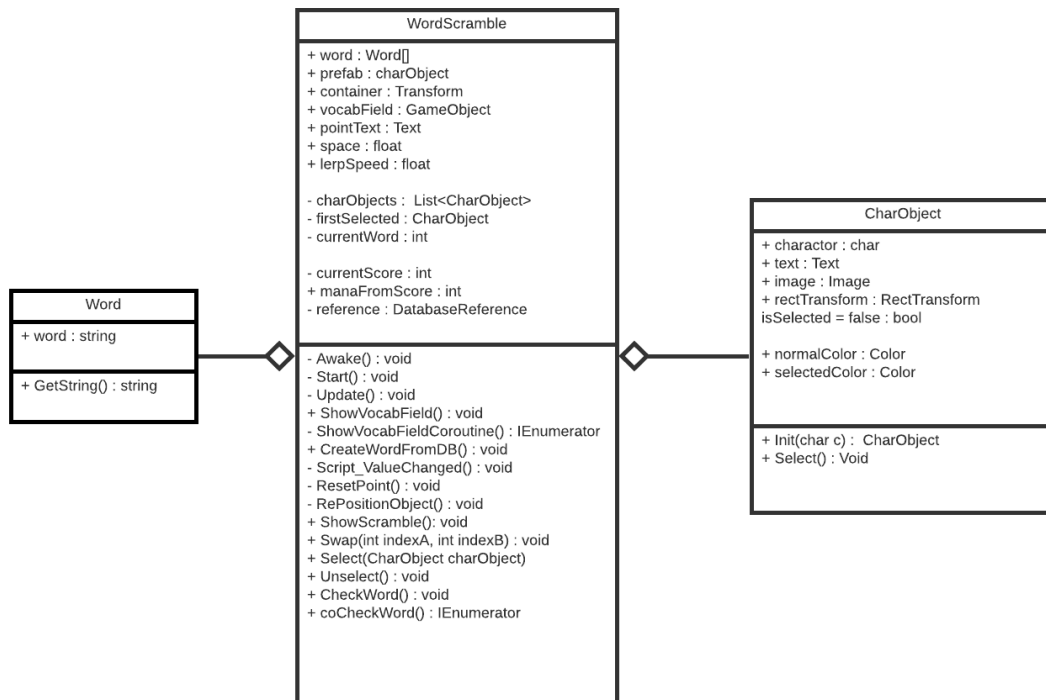
**Graphic:** Adobe Photoshop ใช้ออกแบบ และวาดงาน Graphic ภายในเกม

**Database:** firebase realtime database

แผนภาพ Class Diagram ที่แสดงรายละเอียดภายในระบบ ชื่อคลาส รายการ attribute และรายการ method (ไม่ต้องระบุ getter/setter) พร้อมคำอธิบาย

## ระบบสลับตัวอักษร

### Scramble



มีทั้งหมด 3 คลาสที่เกี่ยวข้อง

1. Class word เป็นคลาสสำหรับเก็บค่าคำศัพท์ และมี method GetString() สำหรับทำให้คำศัพท์กลายเป็นตัวอักษรที่ใช้ในการสลับ
2. Class CharObject จะเป็นคลาสสำหรับควบคุมตัวอักษร ควบคุมสีขณะกดและยกเลิกกดตัวอักษร
3. Class WordScramble จะเป็นคลาสสำหรับควบคุมระบบการเล่นทายคำศัพท์

โดยการทำงานจะเริ่มจาก

1. method ShowVocabField() เพื่อโชว์พื้นที่ในการเล่น
2. ShowScramble() เพื่อดึงคำศัพท์จาก class Word มาแล้วทำการสร้างobject ตัวอักษรให้เท่ากับจำนวนคำศัพท์ เล่นถึงคำว่า ant ก็จะสร้าง game object มา 3 object คือ a n t
3. RePositionObject() เพื่อจัดเรียงลำดับตัวอักษรใหม่ เช่น ant ให้กลายเป็น tan
4. และ method อื่นๆจะทำงานระหว่างที่ผู้เล่นเล่นเกมอยู่ เช่น Swap() เป็นmethod ไว้สลับตำแหน่งของตัวอักษร Select(), Unselect() ทำงานเมื่อผู้เล่นกดเลือก หรือยกเลิกการเลือกตัวอักษร เมื่อผู้เล่นทำการสลับตำแหน่งตัวอักษร method CheckWord() จะทำการตรวจสอบลำดับของคำที่ผู้เล่นทำการสลับกับคำที่ถูกต้อง

ภาพตัวอย่างการทำงาน

คำที่ถูกต้องคือ company ระบบทำการสลับคำเป็น o-m-n-c-p-a-y และนำมาโชว์ให้ผู้เล่นเล่น



ก่อนสลับ



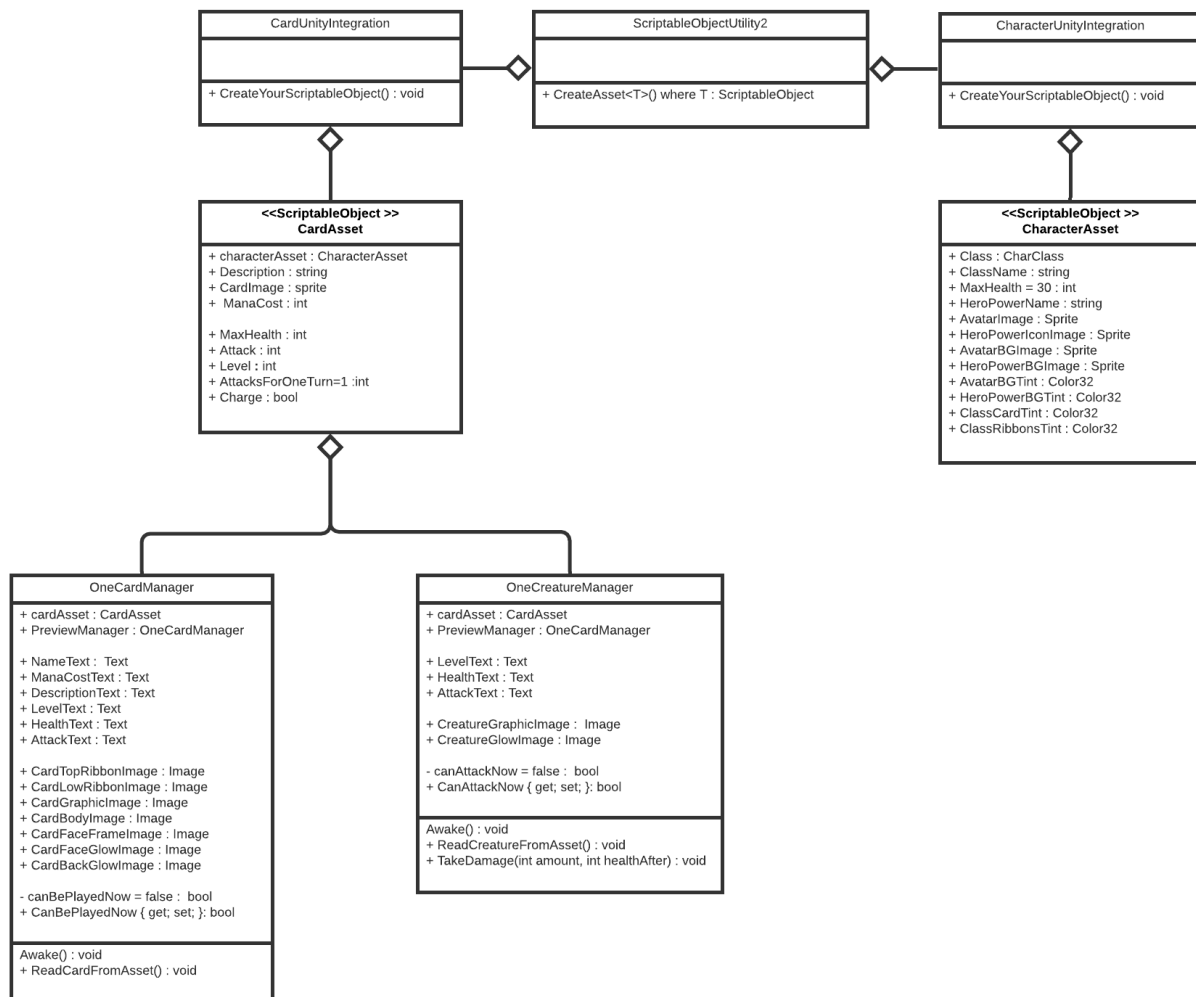
ขณะเลือก ตัวอักษร m



หลังสลับ m กับ O

## ระบบ Object Asset

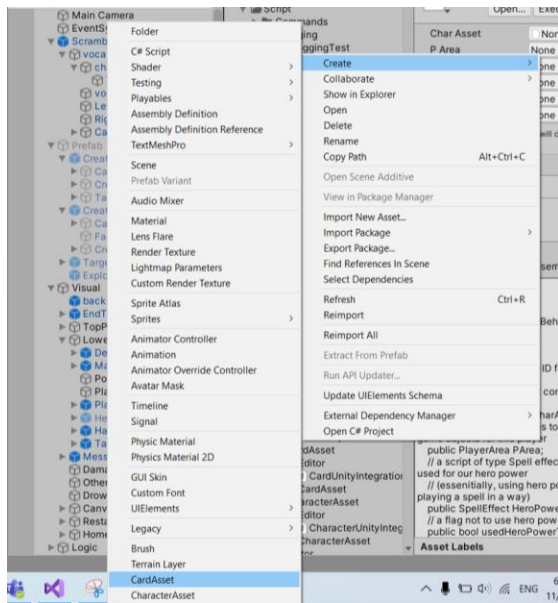
### object Asset



### ประกอบไปด้วย 3 part

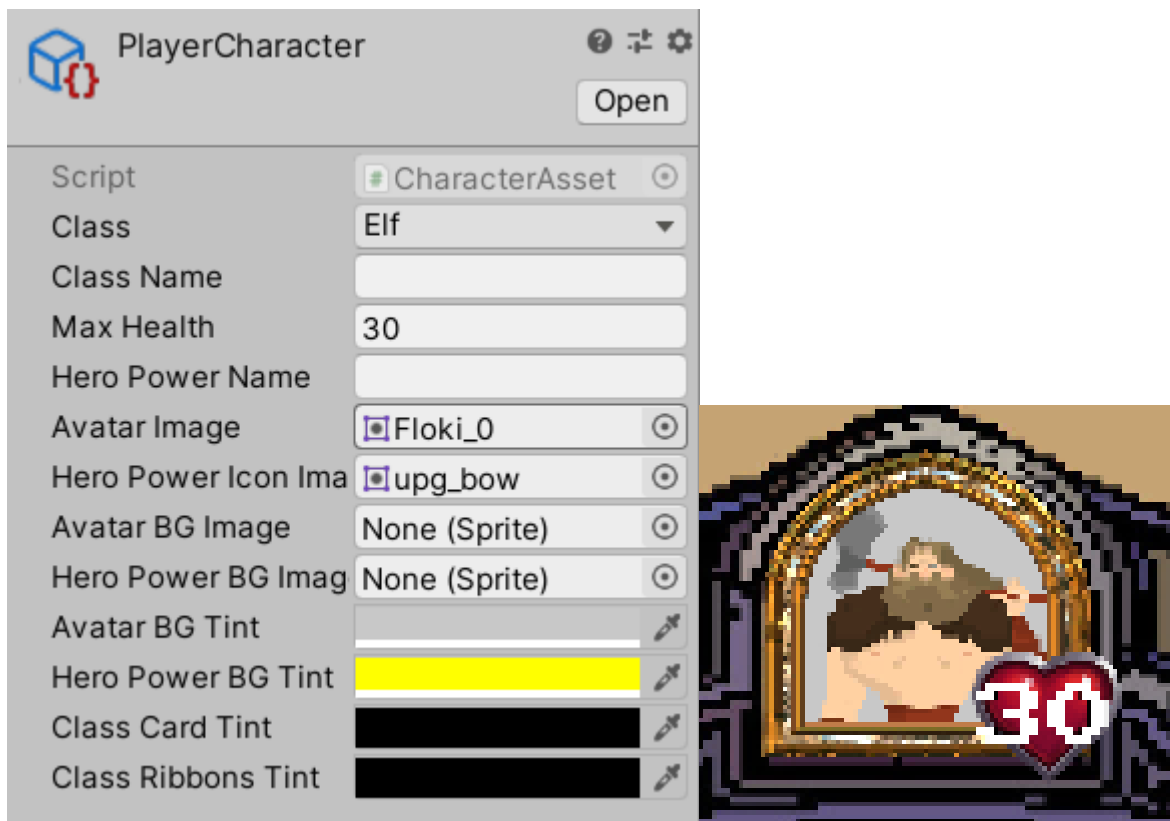
Part 1 คือคลาสที่มีไว้สร้าง command การสร้างobject asset แบ่งเป็น 3 คลาส

1. **ScriptableObjectUtility2** เป็นคลาสไว้สร้างทางลัดในการสร้าง object asset เวลาคลิกขวา
2. **CardUnityIntegration** และ **CharacterUnityIntegration** เป็นคลาสเอาไว้แบ่งจะสร้าง **cardasset** หรือ **charactorasset**



จากรูปด้านบน เมื่อคลิกขวาและกด create จะมีให้เลือกว่าจะสร้าง cardasset หรือ characterasset

Part 2 คือคลาส CardAsset และ CharacterAsset ที่เอาไว้สร้างและกำหนดค่าลง asset ที่จะสร้าง



จากรูปด้านซ้ายคือหน้าต่างในการสร้าง characterAsset และรูปด้านขวาคือ object ที่ได้จากรูปซ้าย โดนสังเกตได้ว่า max health ถูกตั้งกำหนดไว้ที่ 30 และรูปด้านขวาจะรูปตัวเลข 30 ขึ้นอยู่บริเวณรูปหัวใจ

Script CardAsset

**General info**

Character Asset None (Character As

Description  
The Minner

Card Image Minner\_0

Mana Cost 1

**Creature Info**

Max Health 2

Attack 2

Level 1

Attacks For One Turn 1

Charge ☐

Creature Script Name

Special Creature Am 0

**SpellInfo**

Spell Script Name

Special Spell Amount 0

Targets No Target



จากรูปด้านซ้ายคือหน้าต่างในการสร้าง cardAsset และรูปด้านขวาคือ object ที่ได้จากรูปซ้าย โดนสังเกตได้ว่า ตัวการ์ดนั้นมี description ว่า the minner และมีค่า cost(สัญลักษณ์ซ้ายบน) เป็น1 ค่าพลังโจมตี(สัญลักษณ์ซ้ายล่าง) เป็น2 และมีค่าพลังชีวิต(สัญลักษณ์ขวาล่าง) เป็น2 ตามที่กำหนดไว้

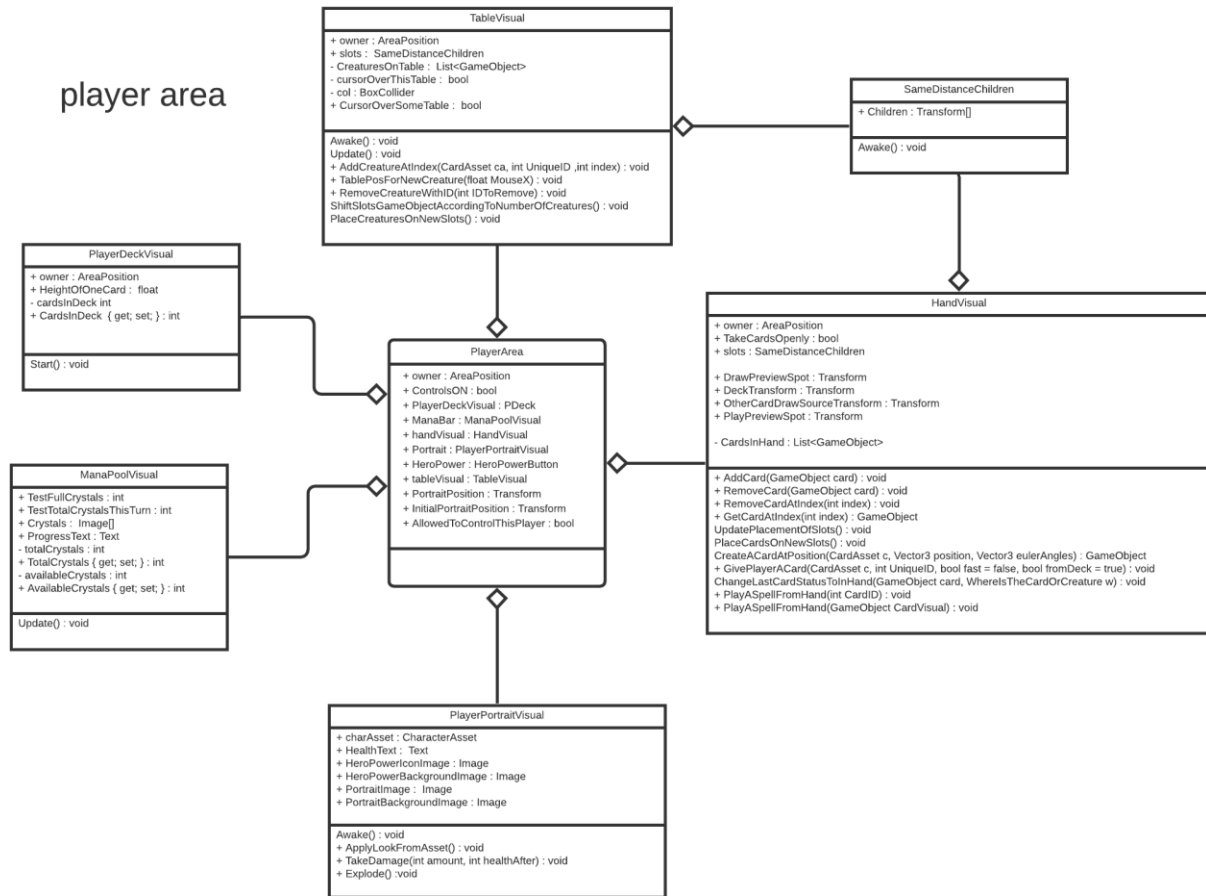
Part 3 คือคลาส OneCardManager และ OneCreatureManager ทั้ง 2 คลาสจะมีหน้าที่คล้ายกันคือทำหน้าที่ในการจัดการบันทึกองค์ประกอบ object ให้ครบตามที่ระบุไว้เพื่อให้การ์ดasset ทุกใบมี pattern ที่เหมือนกัน



จากรูปด้านบนจะเห็นว่าการ์ดทุกใบมี pattern ที่เหมือนกันแต่ต่างกันที่รูปและค่าต่างๆเพราะเป็น cardAsset ที่ต่างกัน

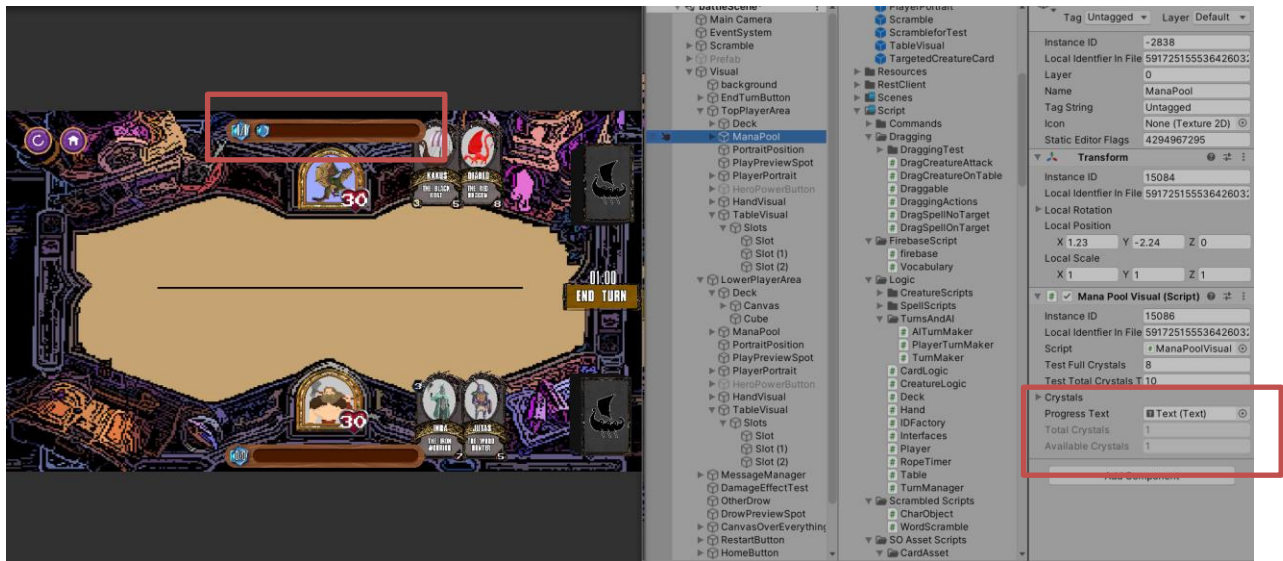


## ระบบจัดการ layout บนกระดาน

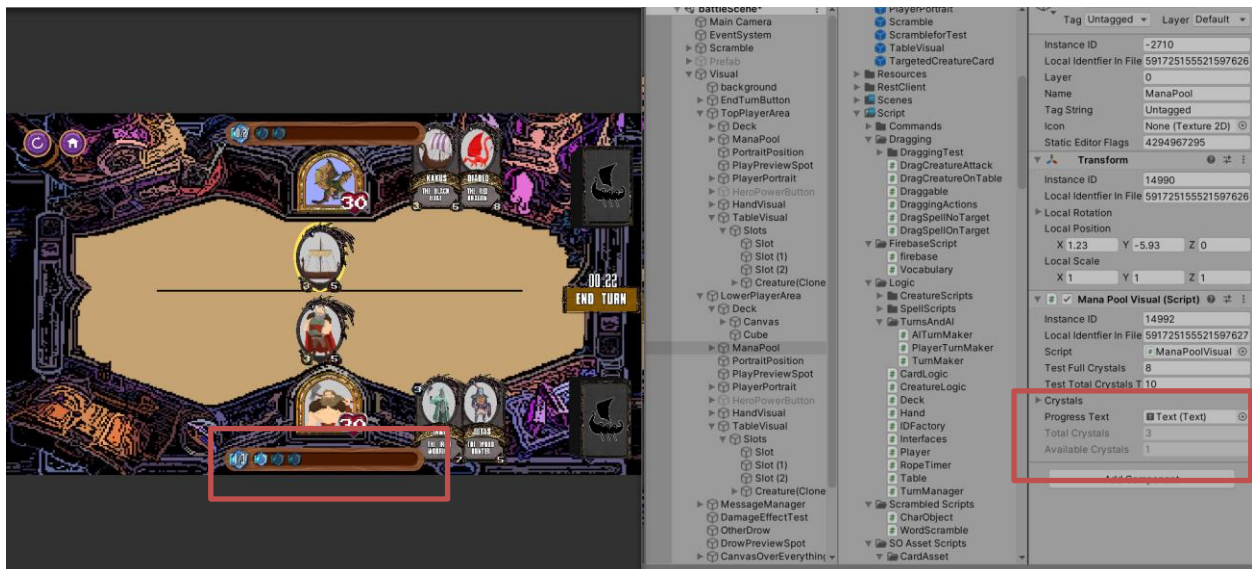


ระบบนี้ทำงานเป็นตัวจัดการประกอบของพื้นที่ในการเล่น โดยมีคลาส **PlayerArea** เป็นตัวจัดการรวมส่วนต่างๆ ให้กลายเป็นส่วนเดียว โดยบนกระดานจะแบ่งเป็น 2 ส่วนคือส่วนบนและล่าง ทั้ง 2 ส่วนจะมีองค์ประกอบเหมือนกัน คือ deck manapool table hand และ playerportrait ส่วน **SameDistanceChildren** คลาส เป็นคลาสสำหรับจัดระยะห่างการ์ด

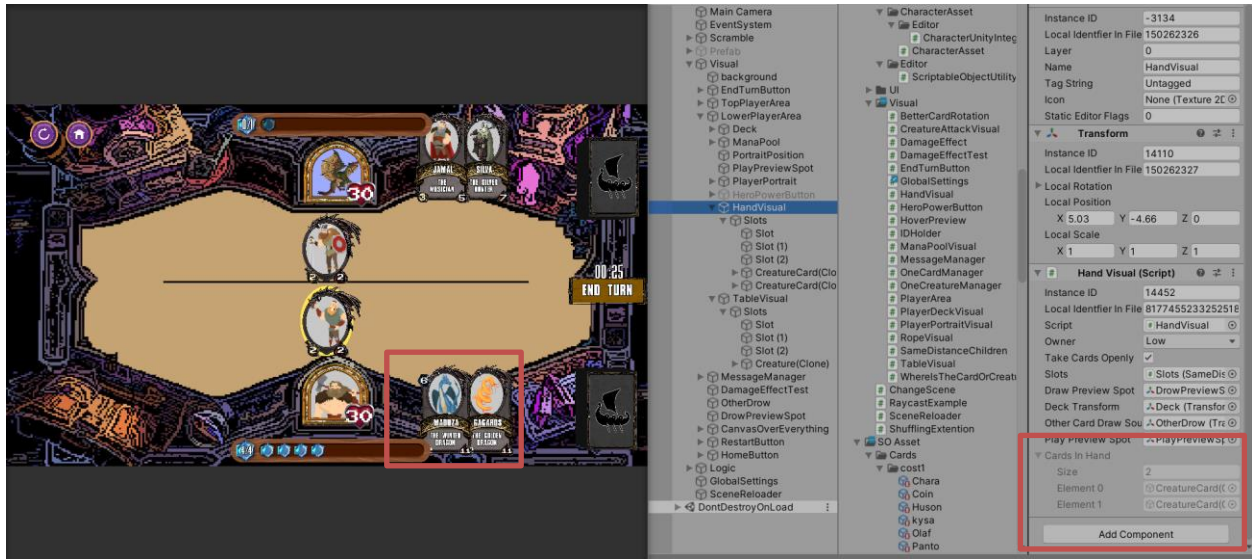
## ภาพตัวอย่างการทำงาน



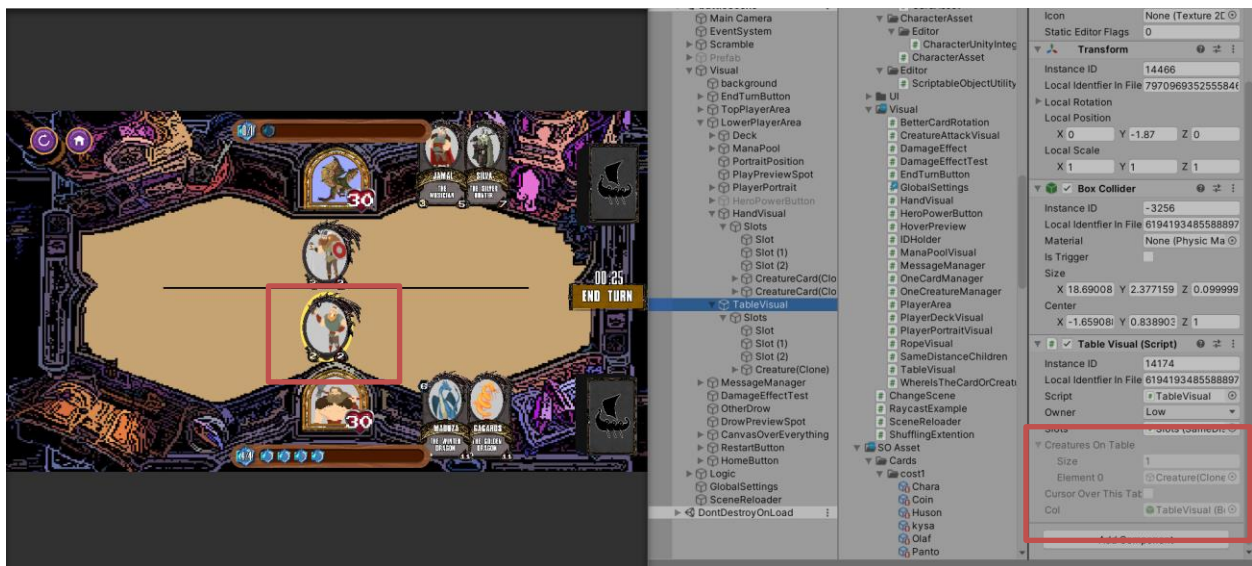
จากรูปด้านบนจะเห็นได้ว่า manapool มี 1 และตามสี่เหลี่ยมด้านขวาก็เป็น 1 เช่นกัน



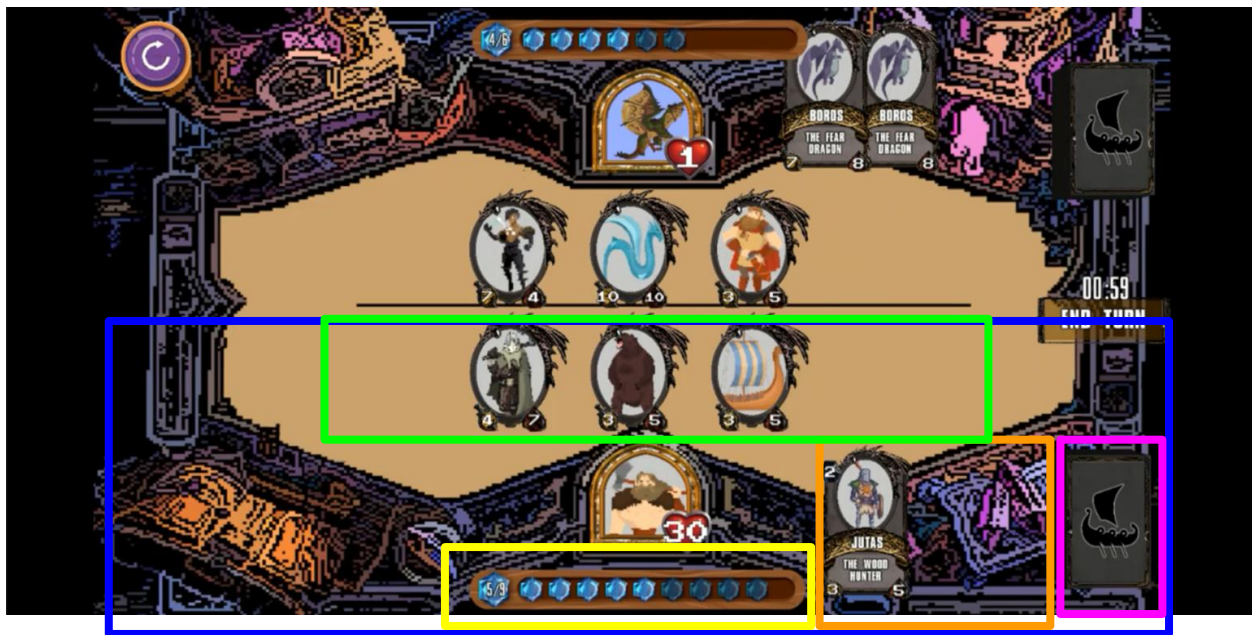
จากรูปด้านบนจะเห็นว่ามานานั้นมี 1/3 และตามสี่เหลี่ยมด้านขวาก็มีค่าเท่ากัน



จากรูปด้านบนจะเห็นว่ามีการ์ดบนมือของเรา 2 ใบและสี่เหลี่ยมด้านขวาก็มี cards in hand อยู่ 2 เซ็นกัน



จากรูปด้านบนจะเห็นว่า มี creature บนสนามของเรา 1 ใบและสี่เหลี่ยมด้านขวาก็มี creature on table อยู่ 1 เซ็นกัน



กรอบสี่เหลี่ยม = TableView class

กรอบสี่เหลี่ยม = HandVisual class

กรอบสี่เหลี่ยม = PlayerDeckVisual class

กรอบสี่เหลี่ยม = ManaPoolVisual class

กรอบสี่เหลี่ยม = PlayerArea class

จากรูปจะแสดงให้เห็นว่า object อยู่ในตำแหน่งตามที่ตั้งค่าไว้

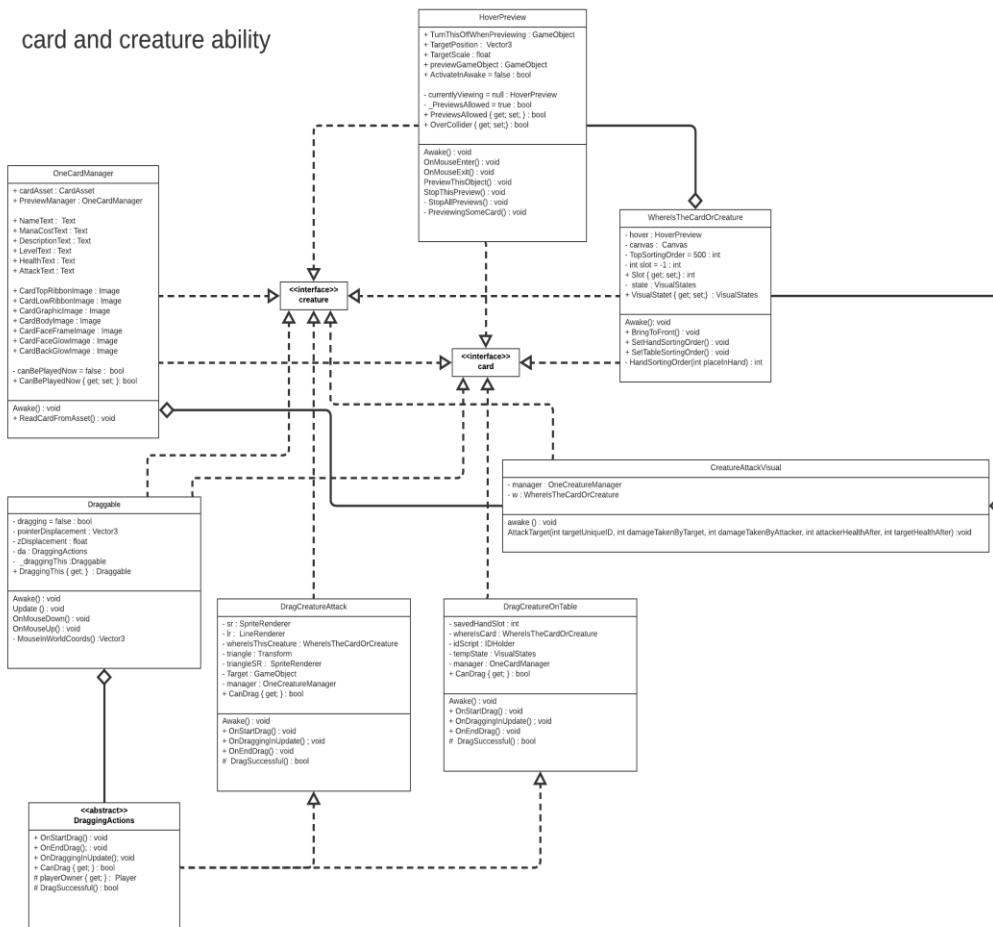


## ระบบความสามารถของการ์ดบนสนามและการ์ดบนมือ

การ์ดบนมือ(กรอบแดง) จากนั้นจะขอเรียกว่า card และการ์ดบนสนาม(กรอบสีน้ำเงิน) เรียกว่า creature

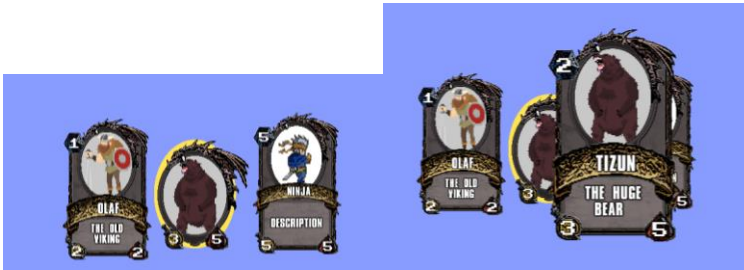


### card and creature ability



โดยทั้ง card และ creature จะมีความสามารถคล้ายถึงกัน คือ

1. Class HoverPreview จะถูกเรียกใช้เมื่อผู้เล่นทำการแตะค้างไว้ที่ card หรือ creature

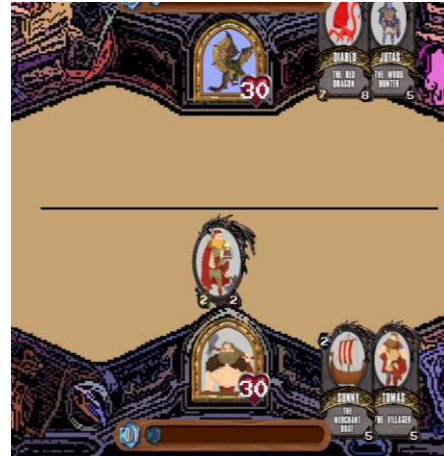


ในภาพจะเป็นการแตะค้างที่ creature หมี เพื่อดูรายละเอียด

2. Class WhereIsTheCardOrCreature เป็นclass ที่ไว้ใช้เปลี่ยน card เป็น creature เมื่อผู้เล่นย้ายการ์ดบนมือไปยังสนาม



ก่อนลงสนาม



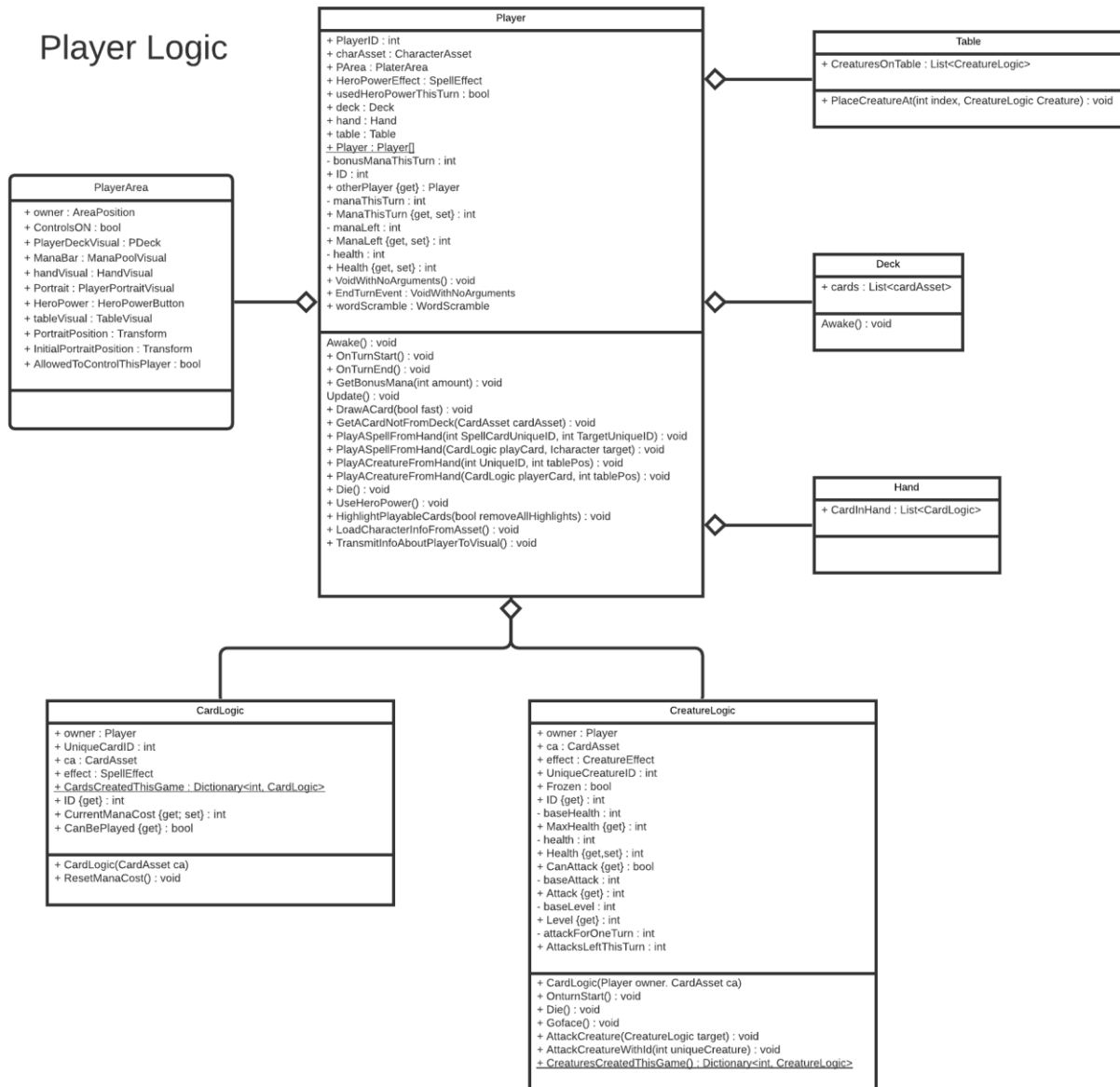
หลังลงสนาม

3. Class Draggable เป็นคลาสสำหรับเคลื่อนย้ายcardหรือ creature ไปยังposition ต่างๆ

ความต่างของความสามารถระหว่าง card และ creature คือ

1. Class DragCreatureOnTable จะใช้สำหรับการ์ดเท่านั้น มีไว้เพื่อย้ายการ์ดไปวางลงบนสนาม
2. Class DragCreatureAttack และ class CreatureAttackVisual จะใช้สำหรับ creature เท่านั้น 2 class นี้จะมีไว้ใช้สำหรับโจมตี creature ของอีกฝ่าย

## ระบบ player Logic



## เป็นระบบดูแล logic ของการเล่นเกม

Class table จะเป็นคลาสที่ใช้สร้าง List<> CreaturesOnTable เพื่อใช้วางcreature บนสนามตามจุดที่ตั้งไว้ใน TableVisual class ในระบบplayer area

Class Deck จะเป็นคลาสที่ใช้สร้าง List<> cards เพื่อใช้เพิ่มจำนวนขนาดในการเก็บการ์ดใน deck

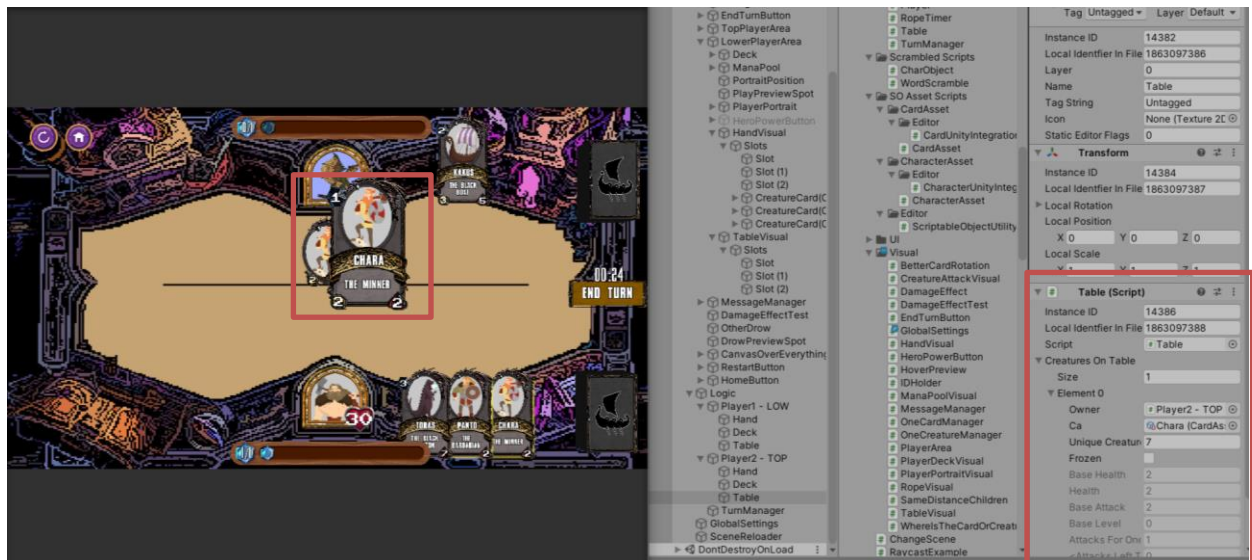
Class Hand เป็นคลาสใช้สร้าง List<> CardsInHand เพื่อใช้วางการ์ดบนสนามตามจุดที่ตั้งไว้ใน HandVisual class ในระบบ player area

Class CardLogic เป็นคลาสกำหนดพฤติกรรมของการ์ดและตรวจสอบว่าการ์ดนั้นสามารถนำลงสนามได้หรือไม่

Class CreatureLogic เป็นคลาสสำหรับกำหนดพฤติกรรมของ creature เช่น ถ้า creature เหลือหลังชีวิต 0 การ์ดจะต้องถูกทำลาย หรือพฤติกรรมต่อสู้เมื่อ creature ของเราทำการโจมตี creature อีกฝ่าย creature ที่ทำการโจมตี พลังชีวิตของ creature จะต้องลดลงตามพลังโจมตีของอีกฝ่าย

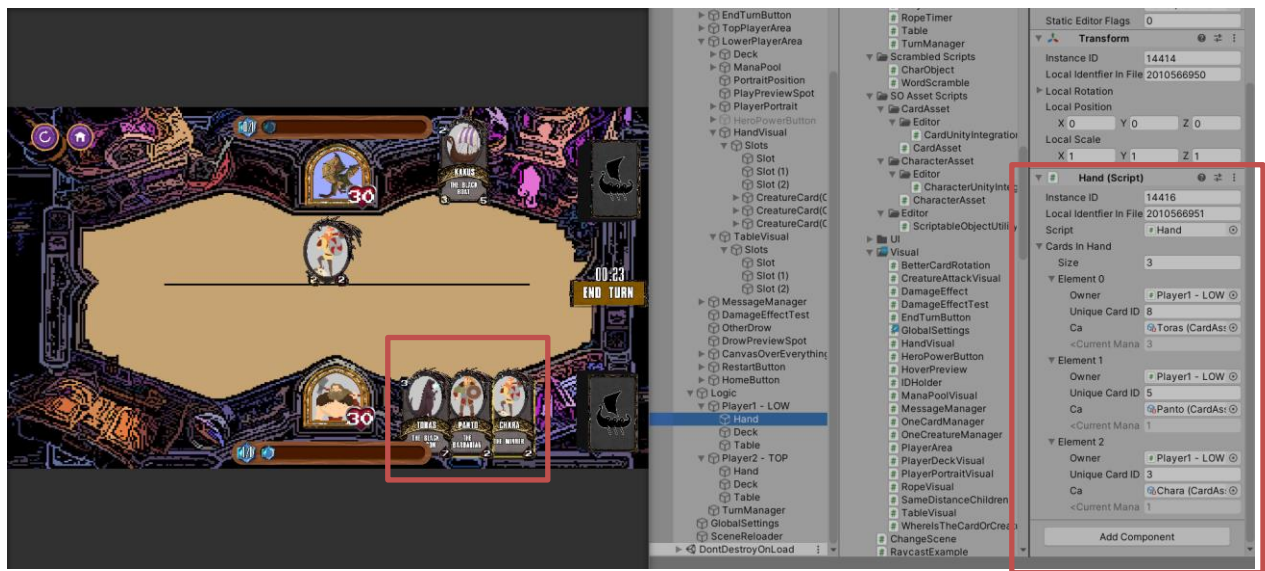
Class Player จะเป็นคลาสสำหรับควบคุมพฤติกรรมของผู้เล่นให้เป็นไปตามกฎ และเป็นไปตาม stage pattern ที่ตั้งไว้

### ภาพตัวอย่างการทำงาน



จากภาพบนสนามฝ่ายบนมี creature อยู่ 1 ตัว ชื่อ chara ซึ่งในสี่เหลี่ยมด้านขวาจะแสดงให้เห็นเหมือนกันว่ามี creature on table อยู่ 1 ตัว owner คือ top และ ca(Card Asset) คือ chara

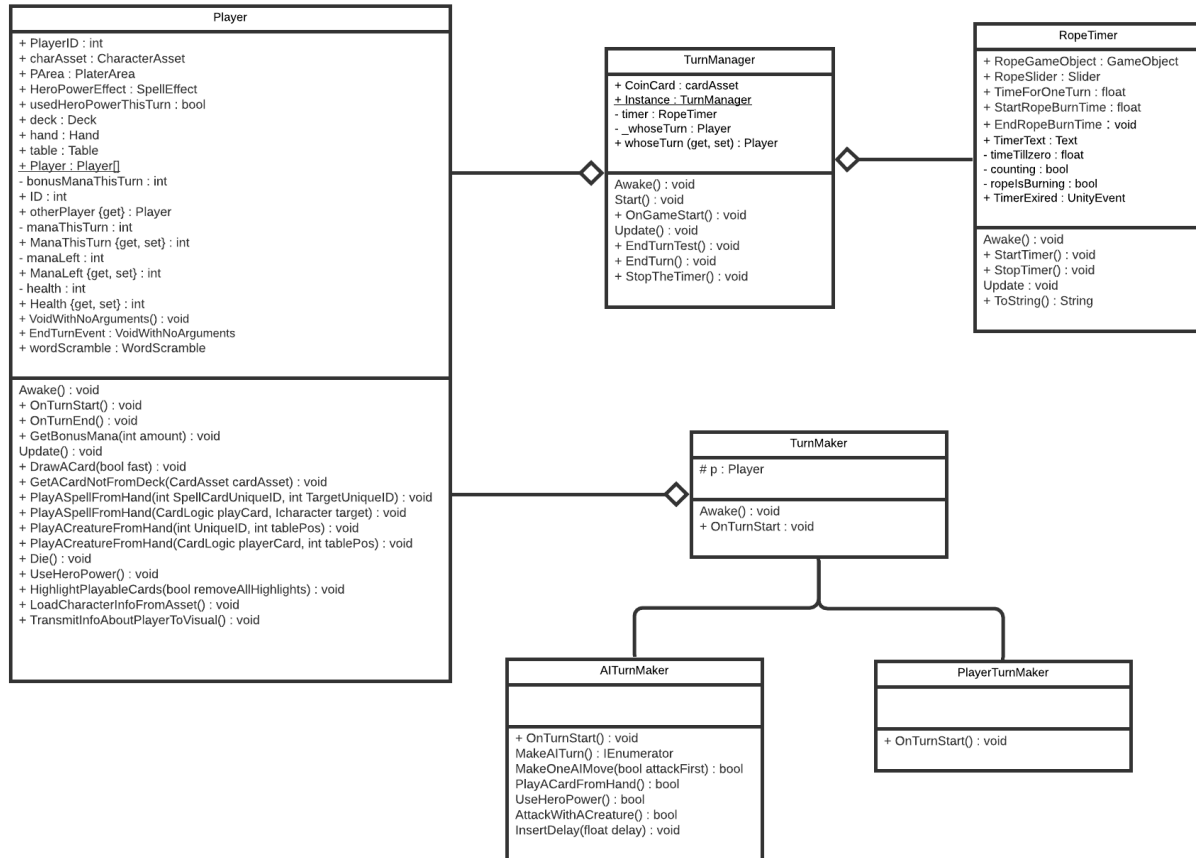




จากภาพด้านบนในมือของฝ่ายล่างมีการ์ด 3 ใบ ซึ่งตรงกับภาพในสี่เหลี่ยมด้านขวาที่แสดงให้เห็นว่า cards in hand มีจำนวน 3 ใบ owner คือ low และชื่อตัวละครทั้ง 3 ใบตรงกันกับการ์ดที่อยู่ในมือ

## ระบบ turn management

### turn management



เป็นระบบที่มีไวน์รอบในการเล่นมี

class RopeTimer เป็นตัวจับเวลา

Class TurnManager เป็น class สำหรับควบคุมพฤติกรรมในแต่ละ stage

Class Turnmaker เป็น class ไว้กำหนดว่า turn นี้เป็น turn ของบอทหรือคน

Class AiTurnMaker และ PlayerMaker เป็นคลาสสำหรับกำหนด stage การเล่น ของบอทและคน