

# Individual Study Final Report

## Making Mario 3D game with Unity3D

Nisaruj Rattanaaram 6031033521

Sorawit Sunthawatrodorn 6031057621

### ลักษณะเกม

เป็นเกมส์ openworld ในรูปแบบ 3 มิติ โดยได้รับแรงบันดาลใจมาจาก Super Mario 64 โดยตัวละครของเราจะสามารถเคลื่อนที่ได้แบบอิสระ ไม่ว่าจะเป็นการวิ่ง กระโดด โดยมุมมองในเกมจะถูกปรับให้ตามตัวละครของเราอยู่เสมอ นอกจากนี้ยังมีมอนสเตอร์ที่สามารถโจมตีเราได้ แต่เราไม่สามารถตกเป็นผู้ถูกกระทำอย่างเดียวยัง เราจะสามารถปาหินเพื่อกำจัดมอนสเตอร์ได้ด้วย และที่ขาดไม่ได้เลยคือการเก็บเห็ดเพื่อเพิ่มพลังชีวิตให้กับตัวเอง ซึ่ง project นี้จัดทำขึ้นเพื่อการเรียนรู้การสร้างเกมส์ด้วย Unity3D ไม่ได้มีเจตนาจะลอกเลียนแบบหรือหาผลประโยชน์ทางธุรกิจใดๆทั้งสิ้น



Fig. 1. Super Mario 64 gameplay

### Features

- ควบคุมตัวละครได้ 360 องศา พร้อมมุมมองติดตามตัวละคร
- ใช้มุมมองแบบบุคคลที่สาม
- กระโดดเป็น combo ได้ด้วยการกระโดดติดกัน 3 ครั้ง (ครั้งที่ 3 จะเป็น super jump ซึ่งจะกระโดดสูงกว่าปกติ)
- มอนสเตอร์จะเดินเข้าหาเราเมื่อเราอยู่ในระยะโจมตี
- ถูกโจมตีโดยมอนสเตอร์
- Game Over เมื่อ HP หมด
- เก็บเห็ดเพื่อเพิ่ม HP
- โจมตีมอนสเตอร์ด้วยการปาหิน

Game Components (Github: <https://github.com/nisaruj/indiv-game>)

## Player

เราใช้ ThirdPersonCharacter ของ Unity Standard Asset เป็น class หลักของตัวละครเรา สำหรับตัวโมเดล เพื่อไม่ให้เหมือน Mario จนเกินไป เราจึงเลือกใช้ตัวโมเดลของ Unity Chan SD แทน

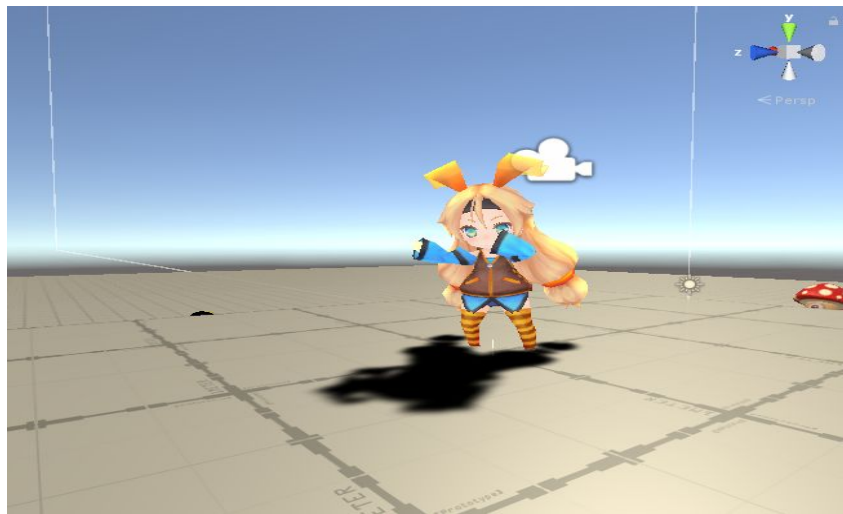


Fig. 2. Our main character (Unity Chan)

## Animations

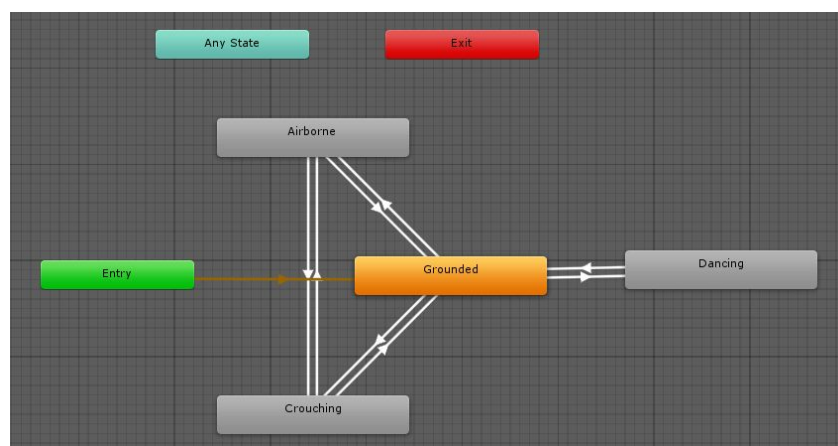


Fig. 3. State diagram of player's animator

Animation State ของตัวละครจะแบ่งแยกได้เป็น 4 State ได้แก่

1. **Grounded** สถานะที่ตัวละครยืนอยู่บนพื้น โดยจะใช้ระยะห่างตามแกน Y ระหว่างตัวละครกับพื้น
2. **Airborne** สถานะที่ตัวละครลอยอยู่กลางอากาศ
3. **Crouching** สถานะที่ตัวละครย่อตัว โดยกดปุ่ม “C” ค้างไว้ หากกดปล่อยจะกลับสู่สถานะ Grounded เว้นแต่ตัวละครลุกขึ้นไม่ได้ (หัวติดเพดาน)
4. **Dancing** เป็นสถานะที่ตัวละครจะเต้นท่า The Sprinkler (Fig. 2.) เมื่อ idle เกิน 10 วินาที

## Jumping mechanics

เราแก้ไข script ใน class ThirdPersonCharacter โดยการเพิ่ม attribute jumpCombo เพื่อทำการนับ combo ขณะนั้น และทำการนับเพิ่มทุกครั้งที่มีการกระโดด และเมื่อครบครั้งที่ 3 จะเป็นการกระโดดสูง แต่ค่า jumpCombo จะถูก reset กลับเป็น 0 เมื่อไม่ได้กระโดดเป็นเวลา 2 วินาที

## Camera

เราพยายามทำมุมกล้องให้ใกล้เคียงเกมต้นฉบับ คือเป็นแบบ Third Person แต่จะไม่ใช้เมาส์ในการควบคุม แต่กล้องจะหมุนไปยังตรงหน้าเราโดยอัตโนมัติ

## Map

เรานำแผนที่จาก Super Mario 64 Course 1 มา simplified ให้ง่ายขึ้น

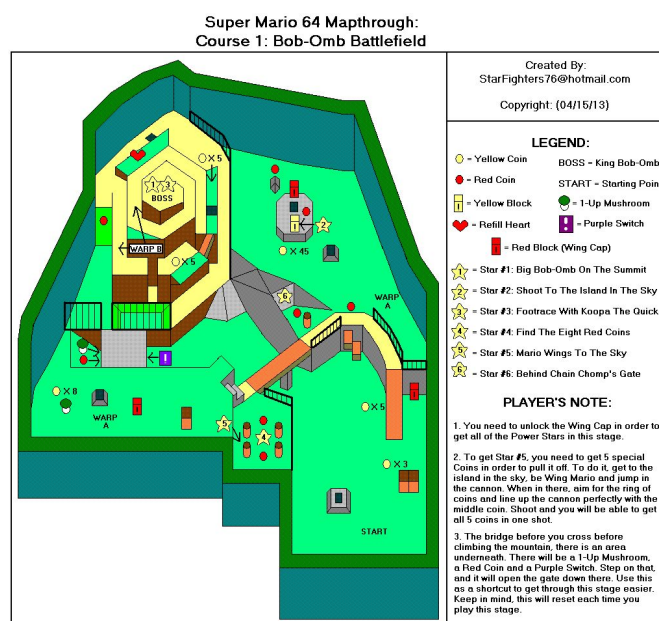


Fig. 4. Original Map from Super Mario 64



Fig. 5. Simplified Map

ตัวแผนที่จะอยู่ในรูปแบบ Low poly Model ใช้ Free Low-Poly Simple Nature Pack สำหรับตัวหอคอยเรา  
ได้ลองใช้โปรแกรม Blender ในการออกแบบ และ export ออกมาใน format .fbx

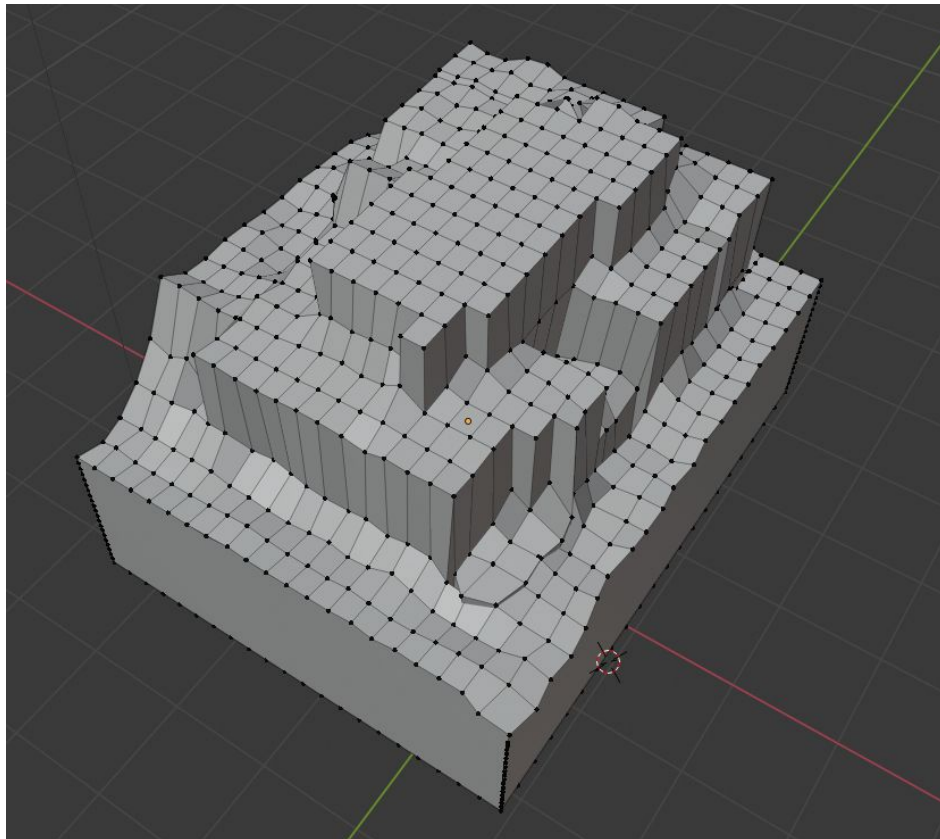


Fig. 6. A tower model created with Blender

## Monsters



Fig. 7. Goomba

ศัตรูในเกมของเราคือ Goomba ซึ่งมีระบบ AI ที่จะเดินตามเราเมื่อระยะห่างถึงตามที่กำหนด โดยหาก Goomba เดินชนตัวละครของเรา จะทำให้ HP ลดลง 1 แล้วตัวละครจะอยู่ในสถานะ invincible (สังเกตจากตัวละครจะกระพริบๆ) ประมาณ 5 วินาที นอกจากนี้ยังมีศัตรูรูประเบิด (Bob-omb) อีกด้วย

## HUD



Fig. 8. Heads-up display (HUD)

ตัวเกมมี Status Bar เพื่อบอก HP และจำนวนเหรียญที่เก็บได้

## Items

ในที่นี้หมายถึง item ที่เราสามารถเก็บได้ ซึ่งมี 2 อย่าง คือ เหรียญ เพื่อเพิ่มพลังชีวิต และ เห็ดยาเพื่อเก็บเป็นแต้ม ซึ่งทั้ง 2 อย่างนี้จะเกิดขึ้นเมื่อตัวละครชนกับ items(OnCollisionEnter) เหมือนกัน ดังนั้นเราจึงใช้คุณสมบัติของ abstract class มาจัดการ items ทั้ง 2 ดัง UML นี้

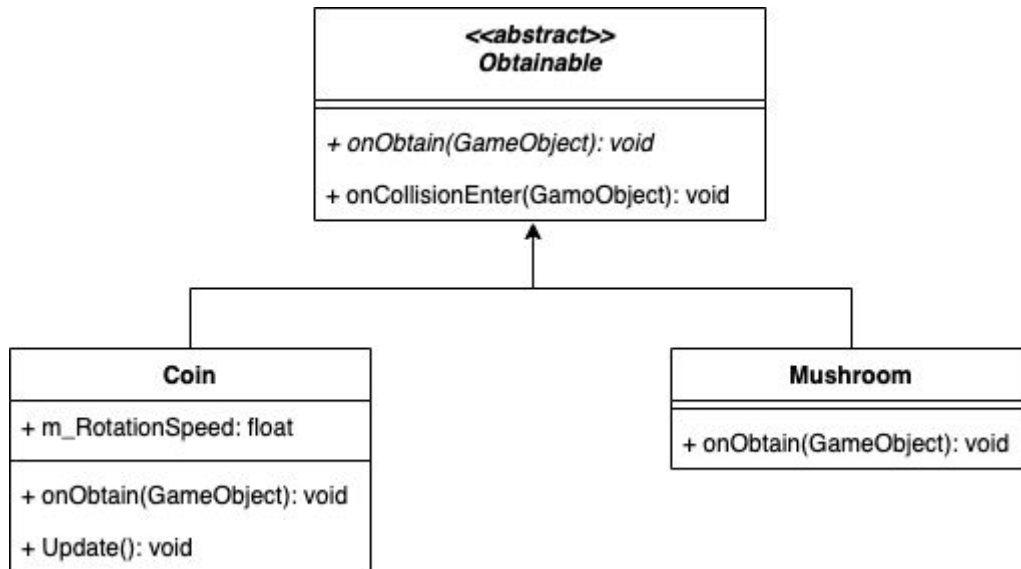


Fig. 9. UML of Obtainable Object



Fig. 10. Coins and Mushrooms

## Attacking

ตัวละครเราสามารถโจมตีศัตรูกลับได้ โดยการ กด E ตัวละครเราจะปาหินพุ่งไปข้างหน้า โดยหากหินชนกับ collider ใดๆ หินจะหายไปทันที



## Gameover controller

เมื่อตัวละครของเราโดนโจมตีจน HP หมด(น้อยกว่าหรือเท่ากับ 0) จะมีหน้าจอ Gameover ขึ้น ในที่นี้ใช้ canvas ซึ่งเป็น Object พื้นฐานและ panel โดยเลือก background เอง เพิ่ม text เพื่อบอกว่า game over แล้วต้องทำยังไง และเขียน script GameOverController เพื่อรอการกด Enter และสั่ง restart แต่จะต้องซ่อน UI นี้ไว้ก่อน รอตอน game over ถึงจะ display ออกมา ซึ่งจัดการโดยการเพิ่ม GameObject ที่เป็น game over scene นี้ และจะถูก set active เมื่อ game over จริงๆ

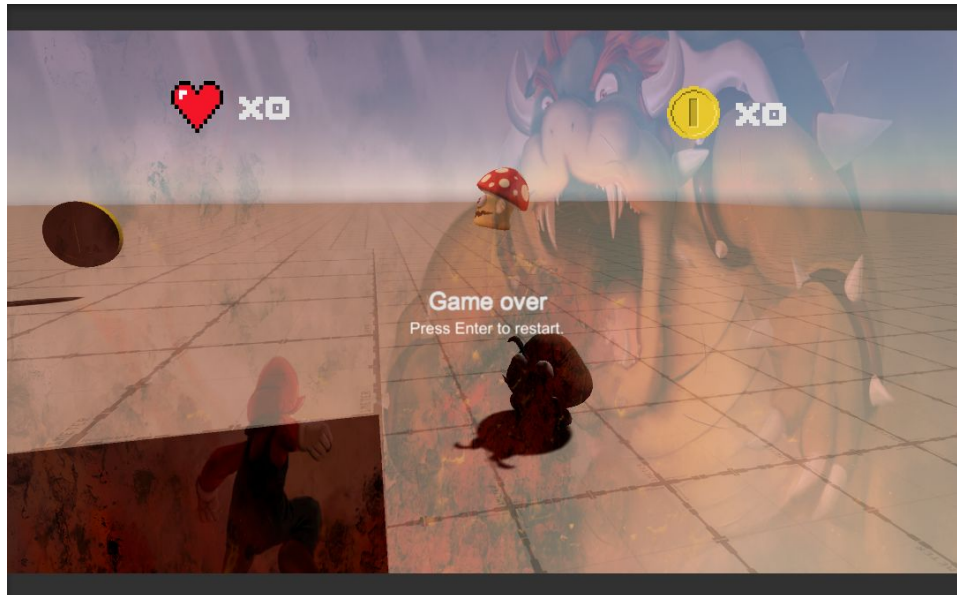


Fig 11. Gameover scene

## Scaling Character

ถ้าพูดถึงเกม mario สิ่งหนึ่งเลยที่ขาดไม่ได้ก็คือการขยายหรือหดของตัวละคร ซึ่งจะเปลี่ยนไปตาม HP ขณะนั้น ในเกมนี้จึงได้มีการ implement ไว้ด้วย โดยสร้าง array scaleMapping ขึ้นมาเพื่อกำหนดว่า HP เท่าไรจะ scale ตัวละครเป็นเท่าไร ซึ่งค่าคือ {0, 0.5, 0.75, 1, 1.25, 1.5} เมื่อเราได้ ratio แล้ว ค่าที่เราจะต้องเปลี่ยนเพื่อให้ scale ของตัวละครเปลี่ยนก็คือ transform.localScale โดยจะสั่งเปลี่ยนค่าเมื่อมีการเปลี่ยนของ HP เท่านั้นเพื่อลดการทำงานใน function Update ในที่นี้จึงทำแค่ใน method takeDamage และ heal

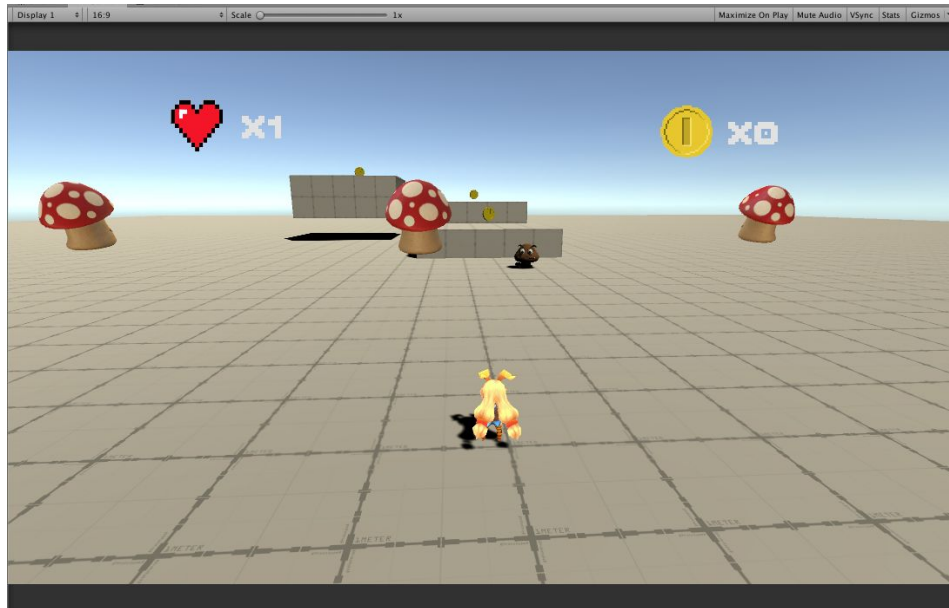


Fig. 12. Character size when HP is 1

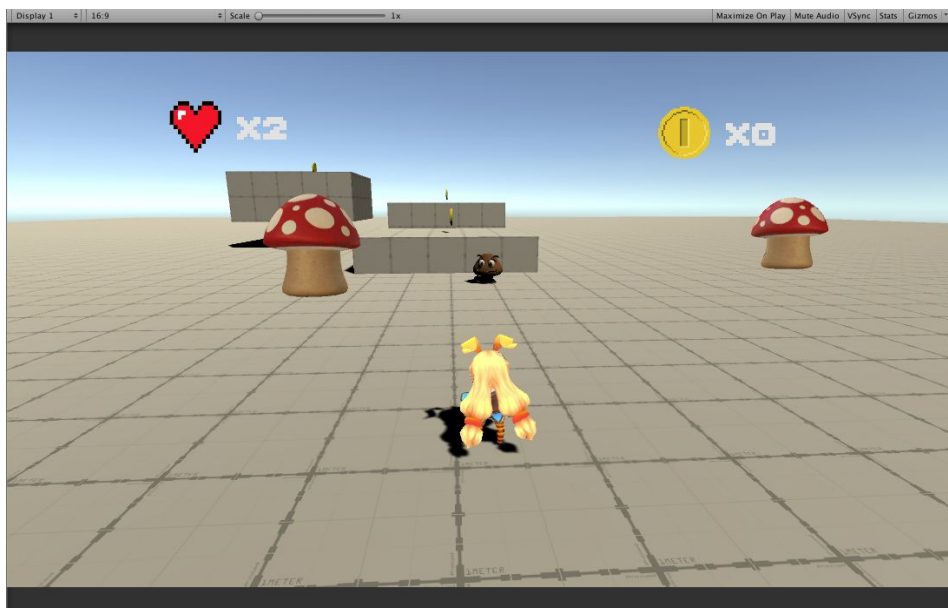


Fig. 13. Character size when HP is 2





Fig. 14. Character size when HP is 3



Fig. 15. Character size when HP is 5

## งานที่รับผิดชอบ

Feature	Name
Character + Camera	Nisaruj
Animation	Nisaruj
Jumping Mechanic	Both
Map	Nisaruj
Attacking	Nisaruj
HUD	Nisaruj
Coin	Nisaruj
Goomba	Both
Gameover	Sorawit
Powerup mushroom	Sorawit
Scaling	Sorawit