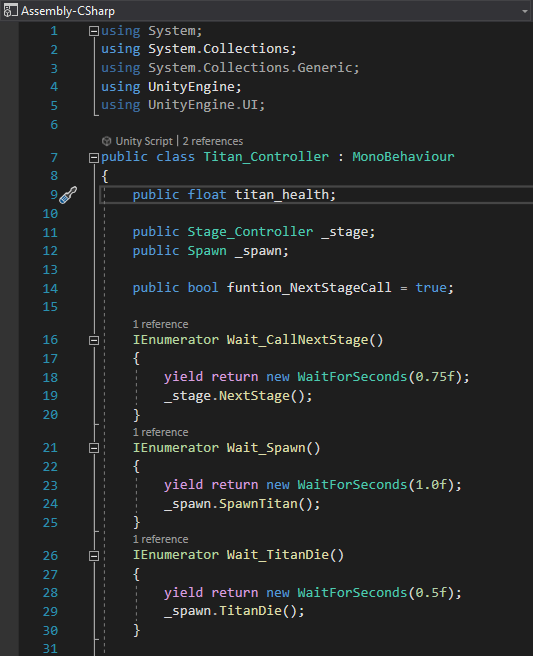
1. ในส่วนของโค้ด Titan\_Controller มีหน้าที่จัดการในส่วนของตัวมอนเตอร์ ซึ่งในเกมนี้จะถูกเรียกว่า Titan



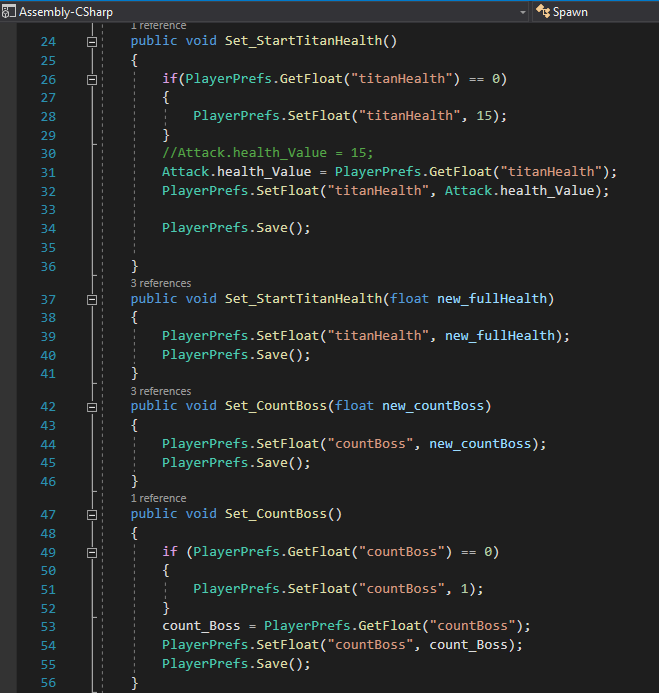
เมื่อ Script เริ่มทำงานจะทำการรับค่า Stage ที่อยู่ในปัจจุบันมาจาก Script Stage\_Controller และใช้ฟังก์ชัน SpawnTitan() เพื่อสร้าง Titan ขึ้นมาเมื่อเริ่ม Scene

และใน Update() จะทำการเช็คว่า เมื่อเลือดของ Titan ที่สู้อยู่ในปัจจุบันน้อยกว่าหรือเท่ากับศูนย์ และ bool function\_NextStageCall มีค่าเท่ากับ true จะทำงาน StartCoroutine() ข้างใน โดยจะมี สามอย่างที่ทำคือ TitanDie() NextStage() และ SpawnTitan() ที่อยู่ใน Script อื่นโดยที่ใช้เป็น StartCoroutine() เพื่อที่ขั้นตอนการทำงงานจะไม่ทำงานพร้อมกัน

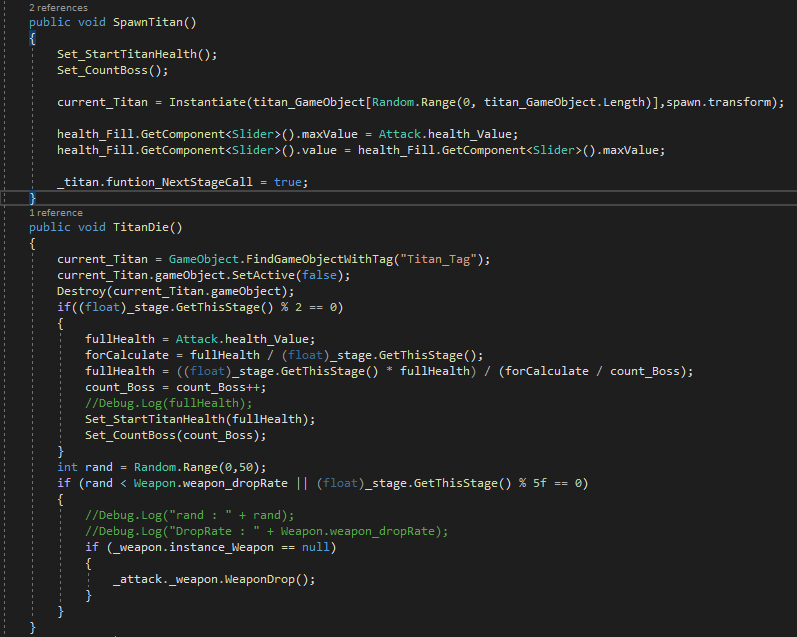




2. ในส่วนของโค้ดที่มีใน Script Spawn ฟังก์ชันส่วนใหญ่จะเป็นการเซฟค่า titanHealth ไว้ใน PlayerPrefs ชื่อ titanHealth ว่าเมื่อเริ่ม Scene มาให้ตั้งค่าเลือดของ Titan เป็น titanHealth ที่เล่นถึงไว้ และเซฟค่า countBoss ไว้ใน PlayerPrefs เพื่อใช้ในการคำนวณค่า titanHealth ต่อไป



ส่วนนี้ คือฟังก์ชันในโค้ด Spawn ที่นำไปใช้ใน Script Titan\_Controller โดยใน TitanDie() จะหา Object ที่มี Tag “Titan\_Tag” เพื่อทำลายและ ตั้งค่าเลือดของ Titan ตัวต่อไปให้เพิ่มมากขึ้น เมื่อผ่านไปทุก ๆ 2 Stage และ SpawnTitan() จะสุ่มสร้าง Titan ตัวใหม่ขึ้นมาโดยมีเลือด ตามที่ถูกตั้งค่าไว้ และจะมีการสุ่มอาวุธขึ้นมาตามโอกาศ dropRate แต่ในทุก ๆ 5 ด่านจะสุ่มอาวุธขึ้นมาแน่นอน

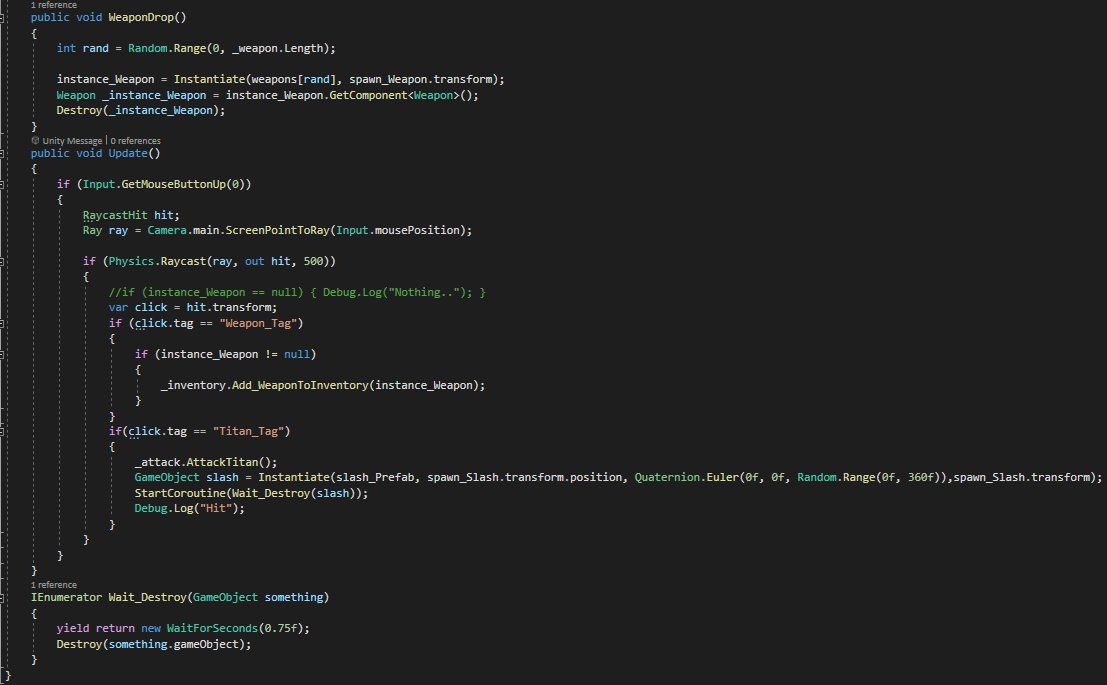


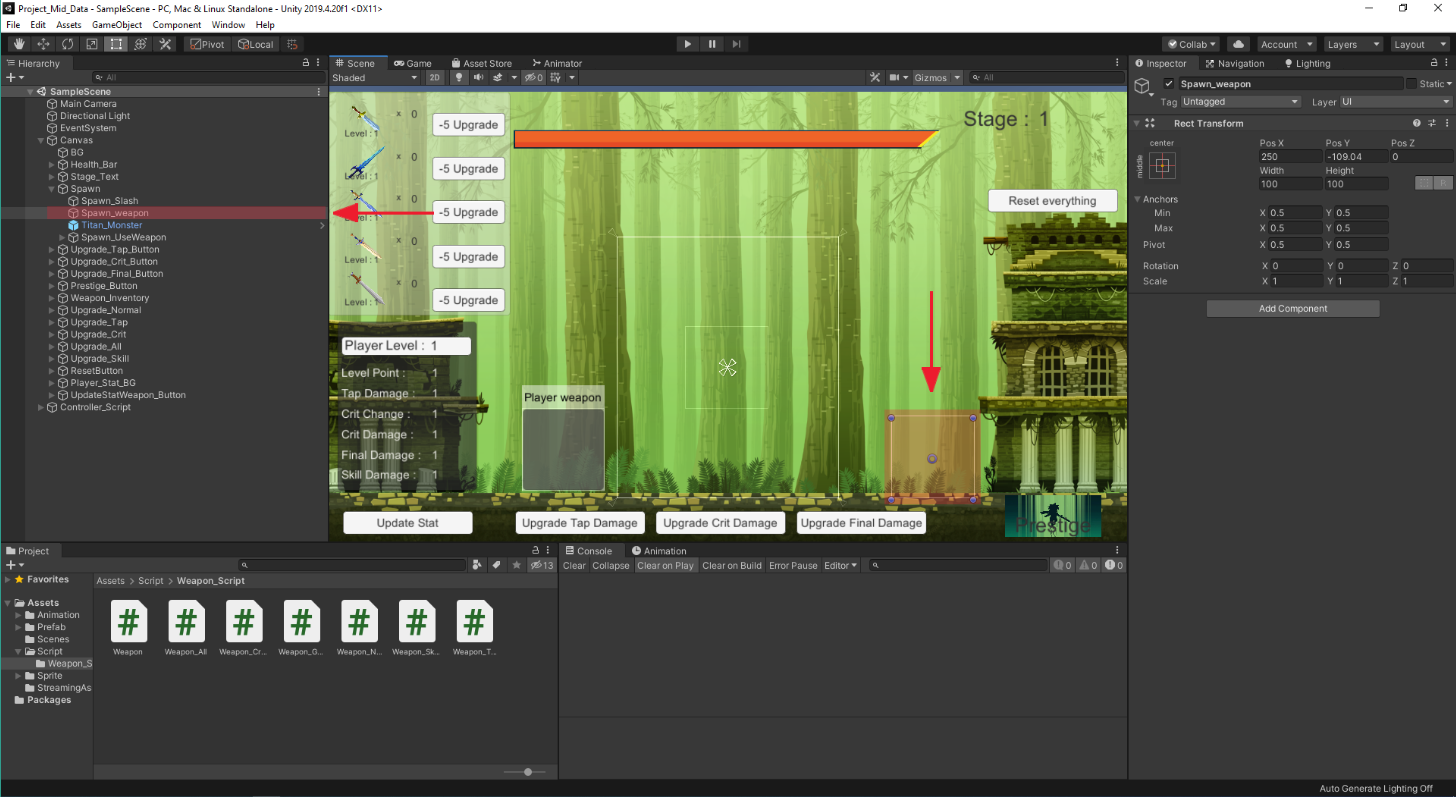
1. ส่วนของ Script Weapon จะเป็นการตั้งค่า Stat ของอาวุธ โดยจะคูณ stat เริ่มต้นของอาวุธนั้น ๆ กับ level ของอาวุธนั้น ๆ ทั้งยังเปลี่ยน Text stat weapon ที่อยู่ใน Scene และมี method SetStartStatWeapon() เพื่อใช้ใน Script ที่คุมควบเกี่ยวกับการเริ่มเล่นใหม่



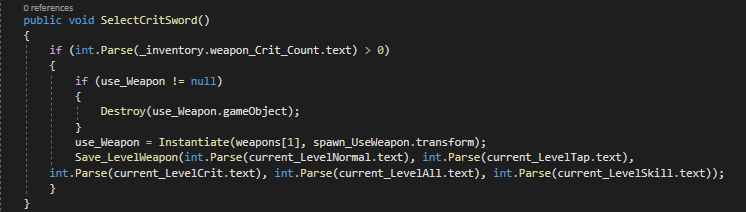


โค้ดในส่วนนี้ จะมีฟังก์ชัน การสุ่มดรอปของไปเทมโดยจะสุ่ม Instantiate อาวุธในจุด spawn\_Weapon ใน Update() จะเป็นการเช็คว่ามีการ กดเมาส์ซ้ายหรือไม่ หากกด จะเป็นการใช้ Raycast จากนั้นเช็คว่า Ray ชนอยู่กับอะไรหากเป็นอาวุธที่ดรอป จะเป็นการทำงานฟังก์ชัน Add\_WeaponToInventory() โดยนำอาวุธที่ Ray นั้นไปชนเข้าไปใช้งาน และ Ray นี้ยังเช็คอีกว่าถ้าหากไปชนกับ Titan จะทำการ Instatiate Prefab ของเอฟเฟกค์การฟันชื่อ Slash บนตัวของ Titan โดยสุ่มองศา จากนั้น 0.75 วิ slash นั้นก็จะถูกทำลาย



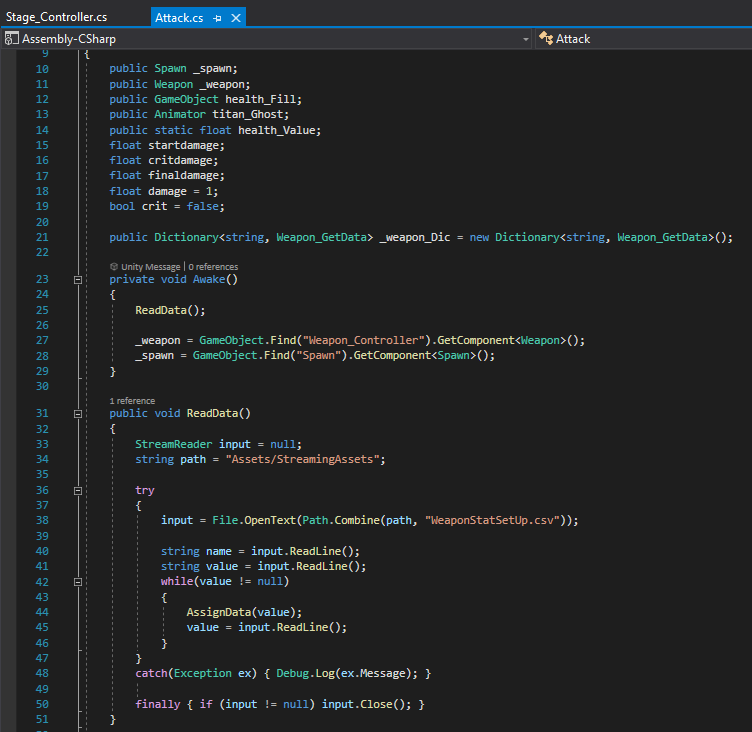
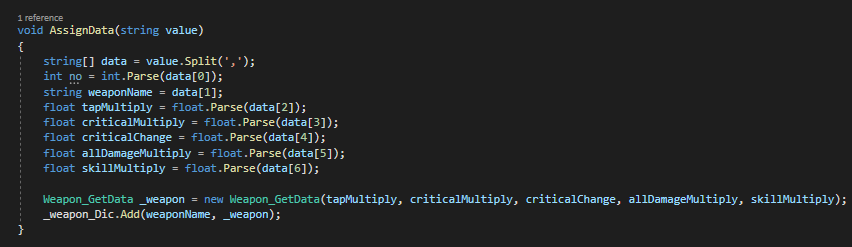


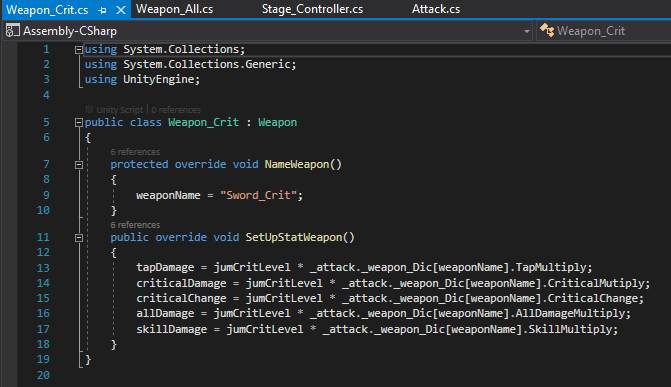
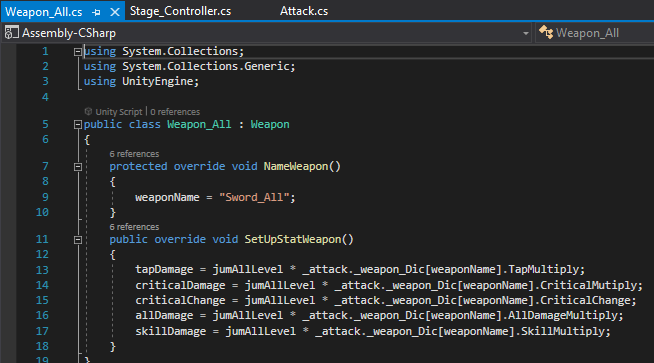
และเมื่อกดที่ภาพของอาวุธนั้น ๆ แล้วจะเป็นการใช้งานอาวุธนั้นซึ่งเพิ่ม Damage ให้กับตัวผู้เล่น และแสดงค่าต่าง ๆ ของอาวุธทางหน้าจอ อย่างสุกท้ายคือ Script นี้เป็น Class แม่ที่ถูก Inheritance ไปโดย Script อาวุธแต่ละชนิดเพื่อรับค่า Stat ของตัวเองจากไฟล์ CSV



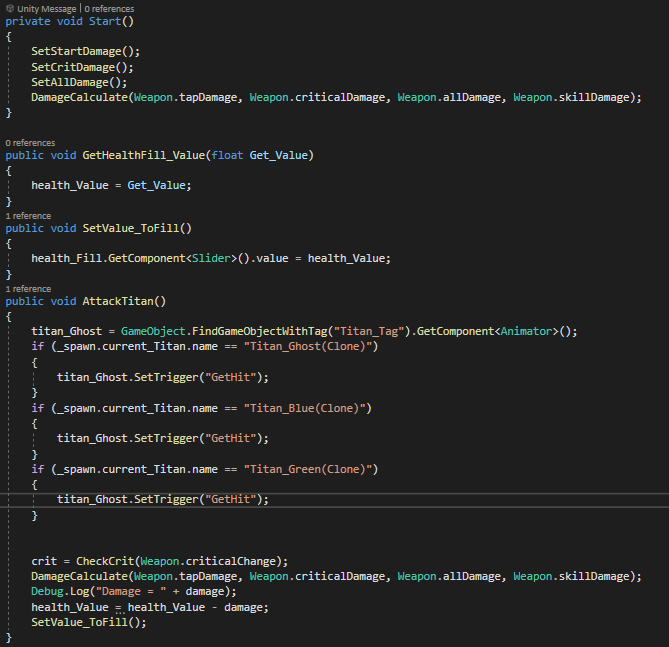


1. ในส่วนของ Script Attack นั้นจะเป็นการคำนวณค่าต่าง ๆ ที่ถูกส่งมาเพื่อนำไปใช้เป็น Damage ของตัวผู้เล่น โดยจะทำการอ่านไฟล์ CSV เพื่อนำค่า Stat ของอาวุธกลับไปที Script Weapon เพื่อเช็ค Stat ของอาวุธ โดยใช้ dictionary ในการตั้งค่า Stat ของอาวุธ

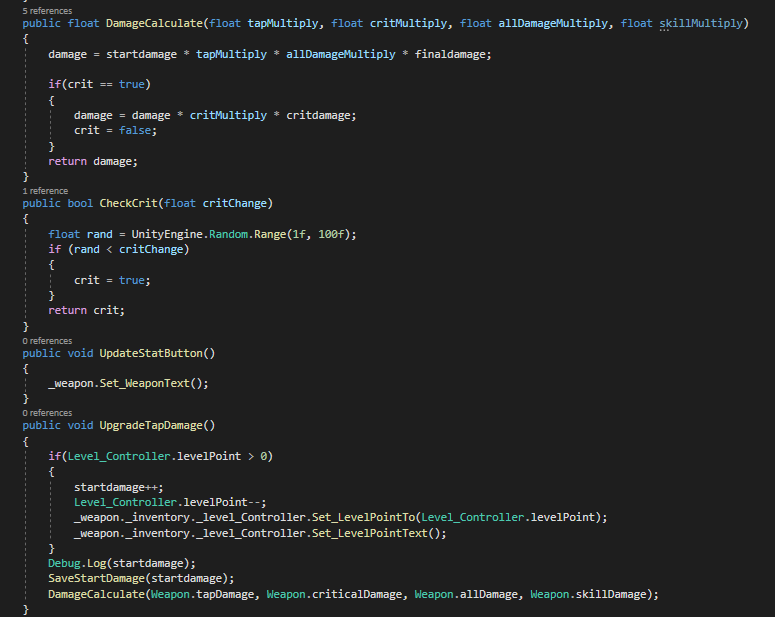
 

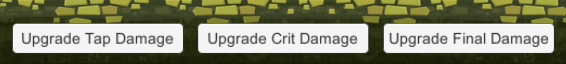


โค้ดส่วนนี้เมื่อเริ่มทำงานจะตั้งค่า Damage เริ่มต้นต่าง ๆ ของผู้เล่นที่มาจากการอัพเกรด จากนั้นจะทำการคำนวนDamage ของผู้เล่น และ ฟังก์ชัน AttackTitan(); จะถูกเรียกใช้ใน Script Weapon เมื่อ Ray ชนกับ Titan โดยเมื่อทำงานแล้ว จะเช็คว่า Titan ที่เจออยู่นั้นชื่ออะไร และเล่น Animation ของ Titan ชื่อนั้น จากนั้นจะทำการเช็คว่าการตีครั้งนี้ติดคริหรือไม่ และคำนวน Damage ใหม่อีกครั้ง และลบเลือดของ Titan ไปตาม Damage ที่ได้

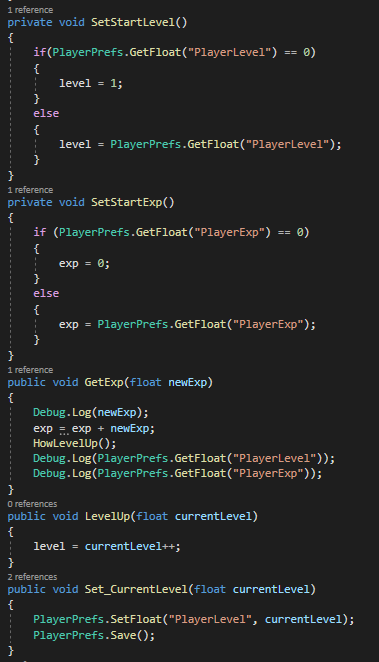


โค้ดที่เป็น ฟังก์ชัน UpgradeTapDamage()จะทำการเช็คว่า ผู้เล่นมี levelPoint มากกว่าศูนย์หรือไม่ โดยได้รับมาจาก Script Level\_Controller ซึ่งเป็น Static จากนั้นจึงเพิ่ม Damage ชนิดที่เลือกอัพ และลบค่า levelPoint จากนั้นบันทึกค่า Damage ที่อัพเอาไว้ใน PlayerPref และคำนวน Damage อีกครั้ง





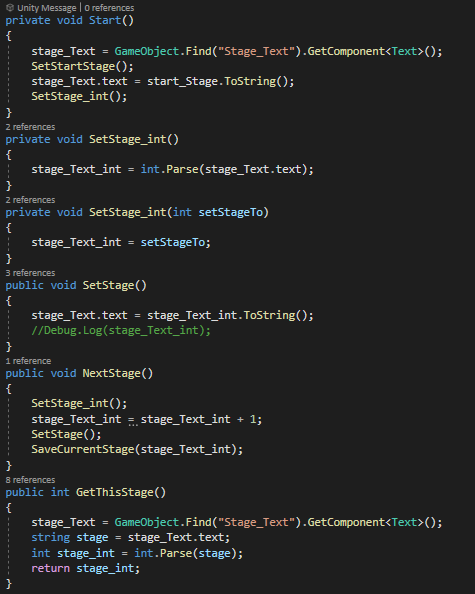
1. Script Level\_Controller นั้นจะเป็นจากตั้งค่า level levelPoint exp ปัจจุบันของผู้เล่น โดยมีหลักการเดียวกันคือ เมื่อเริ่ม Scene จะทำการเช็คค่า PlayerPref ของ level และ exp หากทั้งสองไม่มีค่า ให้ตั้งทั้งสองค่าเป็น 1 และ 0 ตามลำดับ แต่ levelPoint นั้นจะเท่ากับค่า level เมื่อเริ่มเล่นใหม่ หากไม่ได้กดเริ่มเล่นใหม่ ค่า levelPoint จะเท่ากับค่า PlayerPref ที่เก็บเอาไว้ อย่างสุดท้ายคือ เมื่อทำการเริ่มเล่นใหม่ จะใช้งานฟังก์ชัน HowLevelUp() ซึ่งถูกเรียกใน Script Stage\_Controller โดยการทำงานคือคำนวณ ค่า expCap โดยคิดจาก level ปัจจุบันของผู้เล่น และ หากเลเวลของผู้เล่นเพิ่มจะทำการบันทึก level ปัจจุบันของ Player เละเพิ่ม levelPoint ขึ้นหนึ่ง



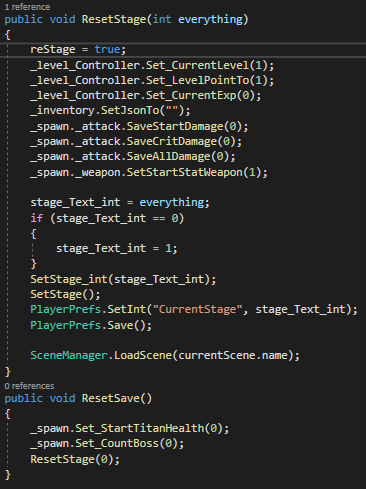
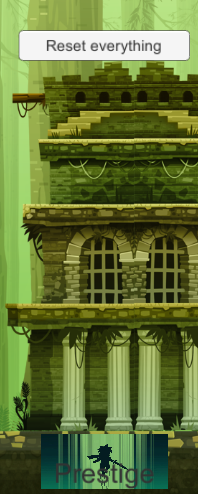


1. Script Stage\_Controller เมื่อเริ่มทำงาน จะเซ็ตค่า Stage ที่อยู่ใน PlayerPref ให้เป็น Stage ที่ผู้เล่นอยู่ในปัจจุบัน ส่วนฟังก์ชัน NextStage() ซึ่งถูกเรียกใช้ใน Script Titan\_Controller คือการเพิ่มค่า Stage ขึ้นหนึ่ง

และ GetThisStage() จะทำการหา Text ชื่อ Stage\_Text ที่อยู่ใน Scene จากนั้นรับค่า text และเปลี่ยนเป็นค่า int ของ stage ที่ผู้เล่นอยู่

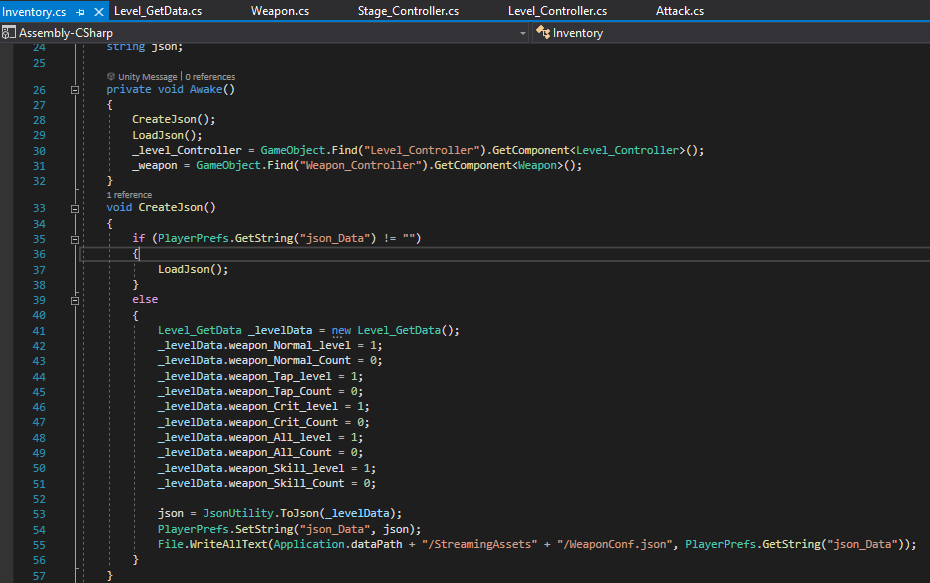


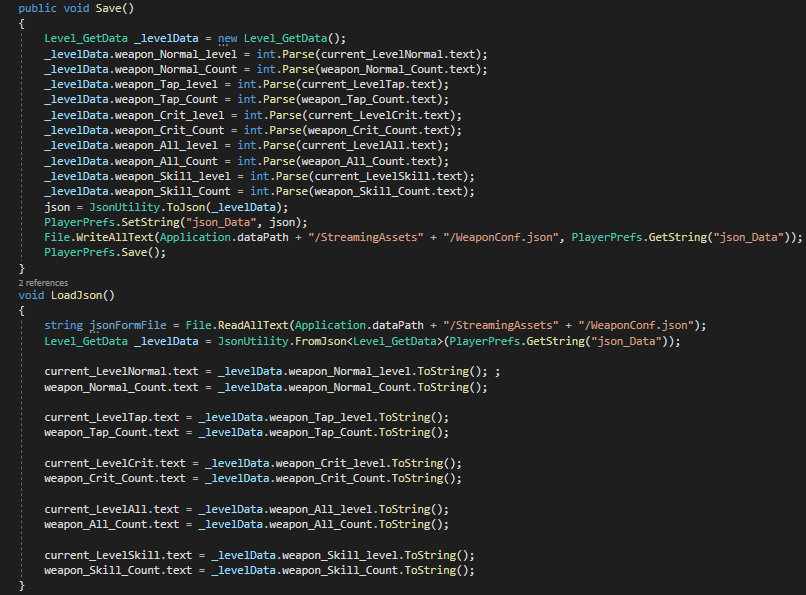
อย่างสุดท้ายคือฟังก์ชัน ResetStage() ซึ่งมันจะทำงานเมื่อผู้เล่นกดปุ่ม เริ่มเล่นใหม่ โดยการทำงานของมันคือตั้งค่าต่าง ๆ ที่บันทึกมาเกือบทั้งหมดกลับ ไปเริ่มแรกอีกครั้งโดยสิ่งที่ผู้เล่นจะได้รับคือ ค่า exp เพื่อนำไปอัพเกรด Damage ต่าง ๆ ได้มากขึ้นเมื่อเริ่มเล่นใหม่

แต่ฟังก์ชัน ResetStage(everything) จะทำงานเหมือนกับ ฟังก์ชัน ResetStage() แต่จะไม่ได้รับ exp และตั้งค่าทุกอย่างที่กลับไปเป็นเริ่มต้นใหม่อีกครั้ง

1. Script ส่วน Inventory จะเป็นการบันทึก ข้อมูลต่าง ๆ ใน JSON ไว้ใน PlayerPref โดยเมื่อเริ่มทำงานจะสร้างไฟล์ JSON ขึ้นมาโดยทำการเช็คว่าไฟล์ JSON นั้นมีข้อมูลอะไรอยู่หรือไม่ หากไม่มีจะสร้างไฟล์ JSON ที่มีค่า level และ จำนวน ของอาวุธเริ่มต้นของผู้เล่น แต่ถ้าหากใน PlayerPref มีข้อมูลของ JSON อยู่จะทำการตั้งค่า level กับจำนวนของอาวุธที่ผู้เล่นเคยเล่นเอาไว้





อีกส่วนหนึ่งของโค้ดนี้คือ เมื่อมีการเรียกใช้ฟังก์ชัน Add\_WeaponToInventory ซึ่งถูกเรียกใช้เมื่อกดลงบนอาวุธที่ดรอปอยู่บนพื้น จะทำงานเพิ่มจำนวนอาวุธชนิดนั้นขึ้นหนึ่ง และเมื่อกดอัพเกรดอาวุธ จะทำการเช็คว่าจำนวนของอาวุธนั้นที่กดอัพเกรดมีมากกว่า 5 หรือไม่หากมากกว่าจะทำการเพิ่ม level ของอาวุธซึ่งเพิ่ม stat ของอาวุธนั้น ๆ

