Rule #1: Avoid Magic Number

Code: ตัวเลข เช่น 10, -2, 5, 20, 3, 30, ถูกใช้ในที่ต่าง ๆ

สิ่งที่ควรปรับปรุง: แทนที่ตัวเลขเวทมนต์ด้วยค่าคงที่ที่มีชื่อ เพื่อเพิ่มความอ่านง่ายในโค้ด เช่น BASE_RUN_SPEED, DEFAULT_ATTACK, หรือ MAX_HP_MULTIPLIER

Rule #2: Don't Repeat Yourself (DRY)

Code: บล็อกโค้ดที่คล้ายกันพบในคลาส Gauntlet และ Ring, โดยเฉพาะในเมทอด applyEffect

สิ่งที่ควรปรับปรุง: แยกตรรกะที่คล้ายกันเป็นเมทอดช่วยหรือนำมาไว้ในคลาสที่ใช้ร่วม เพื่อหลีกเลี่ยงการทำซ้ำและทำให้การบำรุงรักษาเป็นไปได้ง่ายขึ้น

Rule #3: Comment Where Needed

Code: บางเมทอดขาดความคิดเห็นที่อธิบายวัตถุประสงค์หรือพฤติกรรม เช่น useSlashAttack, useDefend, castSpell, และ meditate

สิ่งที่ควรปรับปรุง: เพิ่มความคิดเห็นเพื่ออธิบายวัตถุประสงค์และพฤติกรรมของแต่ละเมทอด เพื่อให้เห็นภาพรวมได้ชัดเจน

Rule #4: Fail Fast

Code: ในเมทอด attack มีการตรวจสอบ this.getAttack() > 0 แต่ไม่มีการจัดการเมื่อเงื่อนไขเป็นเท็จ

สิ่งที่ควรปรับปรุง: พิจารณาการส่งข้อมูลหรือการดำเนินการที่เหมาะสมเมื่อไม่สามารถโจมตีได้ จะช่วยให้สามารถตรวจพบปัญหาได้ในระยะแรกของการพัฒนา

Rule #5: One Purpose for Each Variable

Code: บางตัวแปรเช่น newHP, newMana, newAttackSpeed, และ damageDealt ถูกใช้ในการคำนวณโดยไม่มีการใช้งานตัวกลางที่ชัดเจน

สิ่งที่ควรปรับปรุง: ให้แน่ใจว่าแต่ละตัวแปรมีวัตถุประสงค์เฉพาะและไม่ถูกใช้ซ้ำในงานต่าง ๆ เพื่อเพิ่มความอ่านง่ายของโค้ด

Rule #6: Use Good Names

Code: ชื่อตัวแปรเช่น hpBonus, manaBonus, และ attackSpeedBonus ถูกใช้ที่หลายที่ และความจุของมันอาจไม่ชัดเจน

สิ่งที่ควรปรับปรุง: ใช้ชื่อที่อธิบายได้ชัดเจนมากขึ้น เช่น strengthBonus, intelligenceBonus, gauntletHPBonus, ringManaBonus เป็นต้น