

| # | หัวข้อการให้คะแนน | Weight | | | | |
|----|--|--------|---|---------------------------------------|---|--------------------|
| 1 | ความสมบูรณ์ของโปรแกรม | 30 | ถูกต้องหมด (10) | bug <= 3 (8) | bug > 3 (5) | |
| 2 | ความยากในการพัฒนา | 10 | ยาก ใหญ่ ซับซ้อน (10) เช่น เกมต่อสู้, RPG | ปานกลาง (8) เช่น เกมประยุกต์ | ง่าย หรือ เกมที่ เคยมีมา(6) เช่น เกม ุงเปิดไฟ เกมรับของ ดีตุน หลบสิ่งกีด ขวาง | |
| 3 | ความสวยงาม | 10 | ดีมาก (10) มีทั้ง ภาพและเสียง | ดี (8) สวยงาม ไม่มี เสียง | ปานกลาง (6) | ต่ำกว่ามาตรฐาน (5) |
| 4 | หลักการ OOP ได้แก่ inheritance, interface, polymorphism, access modifier (private, public) & setter/getter | 10 | - inheritance (+2), interface (+2), polymorphism (+2), access modifier/setter/getter (+1), ความถูกต้องของการออกแบบ (+3) | | | |
| 5 | Thread (จะพิจารณาคะแนนให้ เมื่อ การใช้งานถูกต้องตามหลักการ และ เหตุผล) | 5 | มีการใช้ Runnable (5) | มี thread.sleep (3) | ไม่มีการใช้ (0) | |
| 6 | Exception (จะพิจารณาคะแนนให้ เมื่อการใช้งานถูกต้องตามหลักการ และเหตุผล) | 5 | มีการสร้างและ จัดการ Exception ที่สร้างขึ้นเอง (5) | ใช้เฉพาะ Exception ที่มีอยู่ แล้ว (2) | ไม่มีการใช้ Exception (0) | |
| 7 | Coding Sytle (อ้างอิงจาก code review check list) เช่น ชื่อตัวแปร ชื่อคลาส (name convention) | 3 | ถูกต้อง (1) | ไม่ถูกต้อง (0) | | |
| 8 | โปรแกรม run จาก jar ไฟล์ได้ | 5 | ได้ (1) | ไม่ได้ (0) | | |
| 9 | ไฟล์ resource โหลดด้วย | 5 | ได้ (1) | ไม่ได้ (0) | | |
| 10 | Report (หากส่งภายหลังจากนำเสนอไป แล้วจะได้คะแนน/2) | 10 | อธิบายวิธีการใช้ งาน (คู่มือ) หากไม่ ครอบคลุม หักจุดละ 1 คะแนน | | | |
| 11 | Code, Jar, Report, UML Diagram ส่งทาง github ถ้าส่งช้าจะไม่มี คะแนน | 2 | มีครบ (1) | ไม่ครบ (0) | | |

| | | | | | | |
|--|-------|----|--|--|--|--|
| | Total | 95 | | | | |
|--|-------|----|--|--|--|--|