



## Energy War

### เสนอ

ผศ.ดร. มิ่งมานัส ศิวรักษ์

### จัดทำโดย

นายวัชรศักดิ์ พรหมณี 6413110

นายศศิศ ศรีรัตน์ 6413112

นายกวิน เก่งเกตุ 6413210

นายรวิพล ฉายรักษา 6413223

รายงานนี้ เป็นส่วนหนึ่งของรายวิชาเทคนิคการเขียนโปรแกรม (EGCO112)  
หลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยมหิดล  
ประจำภาคการศึกษาที่ 1 ปีการศึกษา 2564

## สารบัญ

ที่มาและความสำคัญ	1
รายละเอียดเกม	1-2
Requirement	3
แผนผังการทำงานของเกม	3
โครงสร้างของเกม	4-5
วิธีการเล่น	5
ข้อจำกัด	5
สรุป	6
หน้าที่	6

## ที่มาและความสำคัญ

จากเรื่องราวในเกมจะกล่าวถึงโลกในยุคปัจจุบัน ซึ่งโลกกำลังตกอยู่ในความโกลาหลด้วยฝีมือที่เกิดขึ้นจากเหล่ามนุษย์ ทำให้เหล่าหุ่นยนต์จากปี 2099 ต้องการย้อนเวลากลับมาฆ่าล้างเผ่าพันธุ์มนุษย์ก่อนที่โลกจะพังหรือเลวร้ายไปมากกว่านี้ ซึ่งในที่สุดจึงทำให้เกิดเหตุการณ์หุ่นยนต์ต่อสู้กับมนุษย์บนพื้นโลก โดยจากที่กล่าวมา จึงเป็นสาเหตุและแนวคิดหลักของเกมที่ทำให้สมาชิกในกลุ่มตัดสินใจทำเกมแนว Turn-base ออกมาเนื่องจากมีรูปแบบของเกม que เล่นได้ง่าย และมีความน่าสนใจ อีกทั้งเนื่องจากแนวเกมดังกล่าว สามารถทำให้มีความหลากหลายของโครงสร้างการเขียนโปรแกรมได้ นี่จึงเป็นที่มาและความสำคัญของโครงงานดังกล่าว

## รายละเอียดเกม

- เกมต่อสู้ Turn-base ที่ต้องเล่นเป็นหุ่นยนต์ต่อสู้กับมนุษย์ การทำ Action ต่างๆ ต้องใช้พลังงาน ถ้าจัดการศัตรูได้หมดก็จะชนะ หากถูกศัตรูจัดการจนหมดก็จะแพ้
- มีด่านทั้งหมด 5 ด่าน แต่ละด่านก็จะมีจำนวนศัตรูและจำนวนระลอกแตกต่างกัน ศัตรูจะมาเป็นระลอก หากเอาชนะศัตรูทั้งหมดได้ก็จะเข้าสู่ระลอกถัดไป หากสำเร็จระลอกก็จะถือว่าเคลียร์ด่านและสามารถเล่นด่านต่อไปได้ ในแต่ละด่านจะมีจำนวนระลอกไม่เท่ากัน โดโนในด่าน1 จะมีแค่ระลอกเดียว ด่าน 2 และ 3 จะมี 2 ระลอก ด่าน 4 จะมี 3 ระลอก และด่าน 5 จะมี 4 ระลอก แต่ละระลอกสามารถเจอกับศัตรูได้สูงสุด 3 ตัว รวมทั้งหากเคลียร์ด่านได้พลังทั้งหมดของหุ่นยนต์จะถูกรีเซ็ตกลับมาเต็ม หากเคลียร์ด่านทั้งหมดจะถือว่าชนะเกม
- การต่อสู้ในแต่ละระลอกจะแบ่งเป็นเทิร์น หากฝ่ายไหนตาย(พลังชีวิตหมด)หมดทุกตัวละครก่อนก็จะเป็นฝ่ายแพ้ แต่ละเทิร์นจะแบ่งเป็น 2 ช่วง ช่วงแรกคือช่วงของหุ่นยนต์ ผู้เล่นจะสามารถเลือกการกระทำของหุ่นยนต์แต่ละตัวได้ในรอบนี้ โดยจะเรียงจาก C3P0 R3D2 และJUNKCAT เมื่อหุ่นยนต์ทั้งหมดได้สำเร็จการกระทำตามคำสั่งแล้ว และยังมีศัตรูเหลืออยู่ก็จะเข้าสู่ช่วงของศัตรู ช่วงนี้ผู้เล่นจะไม่สามารถทำอะไรได้ ศัตรูจะเรียงกันสุมโจมตีหุ่นยนต์ทีละคน หลังจากศัตรูทุกคนโจมตีเสร็จแล้ว ก็จะเข้าสู่เทิร์นถัดไปแล้วก็วนช่วงหุ่นยนต์ใหม่ไปเรื่อยๆ หากภายใน 10 เทิร์นไม่มีฝ่ายไหนเอาชนะอีกฝ่ายไปได้ ก็จะหมดเวลาและถือว่าฝ่ายหุ่นยนต์แพ้นั่น
- หุ่นยนต์มีทั้งหมด 3 ชนิด คือ C3P0 R3D2 และJUNKCAT แต่ละชนิดจะมีค่าสถานะที่แตกต่างกัน สามารถทำการโจมตีได้เหมือนกัน และหุ่นยนต์แต่ละชนิดจะมีทักษะพิเศษแตกต่างกันโดย C3P0 จะมีทักษะในการโจมตีจุดสำคัญทำให้โจมตีศัตรูได้รุนแรง R3D2 จะมีทักษะในการฟื้นฟูพลังชีวิตให้กับตัวเองหรือหุ่นยนต์ตัวอื่น

และJUNKCAT จะมีความสามารถในการโจมตีได้พร้อมกันหลายเป้าหมาย ในด่าน 1 จะมีแค่ C3P0 ในด่าน 2 จะมี R3D2 เพิ่มเข้ามาและในด่าน 4 จะมี JUNKCAT เพิ่มเข้ามาจนครบ 3 ตัว

- ศัตรูจะมีทั้งหมด 4 ชนิด แต่ละชนิดมีความแตกต่างกันที่รูปร่างหน้าตาและค่าสถานะพลัง ทุกชนิดสามารถโจมตีเป้าหมาย(หุ่นยนต์)ได้ที่ละตัวตามค่า atk

- ค่าสถานะมีทั้งหมด 4 อย่างคือ HP def atk และEN

- HP หมายถึงพลังชีวิต สามารถลดลงได้จากการได้รับความเสียหาย และสามารถเพิ่มขึ้นได้ไม่เกินhpตั้งต้นจากการได้รับการฟื้นฟู หากพลังชีวิตหมดตัวละครานั้นก็จะตายและไม่สามารถทำหรือถูกกระทำอะไรได้อีก

- def หมายถึงพลังป้องกัน หากตัวละครนั้นถูกกระทำ ความเสียหายให้พลังชีวิตลดลง ค่าความเสียหายนั้นจะถูกหักลบจาก def ก่อน ยกตัวอย่างเช่นหากA มี hp 3 หน่วย และ def 1 หน่วย หากBโจมตีใส่ A เป็นความเสียหาย 2 หน่วย ความเสียหายนั้นจะถูกลบด้วย 1 ก่อนที่จะเข้าไปลดพลังชีวิต ทำให้สุดท้ายแล้ว hp ของ A จะลดลง 1 หน่วย และเหลือ 2 หน่วย

- atk หมายถึงพลังโจมตี เมื่อตัวละครทำการโจมตีตัวละครฝ่ายตรงข้าม จะสร้างความเสียหายเท่ากับพลังโจมตีและสุดท้าย

- EN หมายถึงค่าพลังงาน จะเป็นค่าสถานะที่มีอยู่ในพวกหุ่นยนต์เท่านั้น หุ่นยนต์ทุกตัวจะมีค่าพลังงานตั้งต้นที่ 5 หน่วย การโจมตีจะใช้พลังงาน 1 หน่วย และการใช้ทักษะจะใช้พลังงาน 3 หน่วย หากพลังงานไม่พอก็ไม่สามารถใช้ทักษะหรือการโจมตีได้ พลังงานสามารถฟื้นฟูได้โดยแลกกับการเสียเทิร์นไป 1 เทิร์นโดยจะฟื้นฟูได้ที่ละ 3 หน่วยโดยจะไม่สามารถฟื้นฟูให้เกินพลังงานตั้งต้นได้ หากพลังงานหมดในเทิร์นนั้น หุ่นยนต์ตัวนั้นก็จะถูกบังคับให้ฟื้นฟูพลังงานทันที ถ้าหากไม่เหลือศัตรูแล้วแต่ยังมีเทิร์นของหุ่นยนต์ตัวนั้นอยู่ หุ่นยนต์ตัวนั้นก็จะฟื้นฟูพลังงานอัตโนมัติ

## Requirement

### 1. Use Class with constructor

- class character (มี พลังชีวิต พลังงาน พลังป้องกัน พลังโจมตี สามารถโจมตี ใช้ทักษะ)
- class LL และ NODE สำหรับสร้างลิ่งค์ลิส

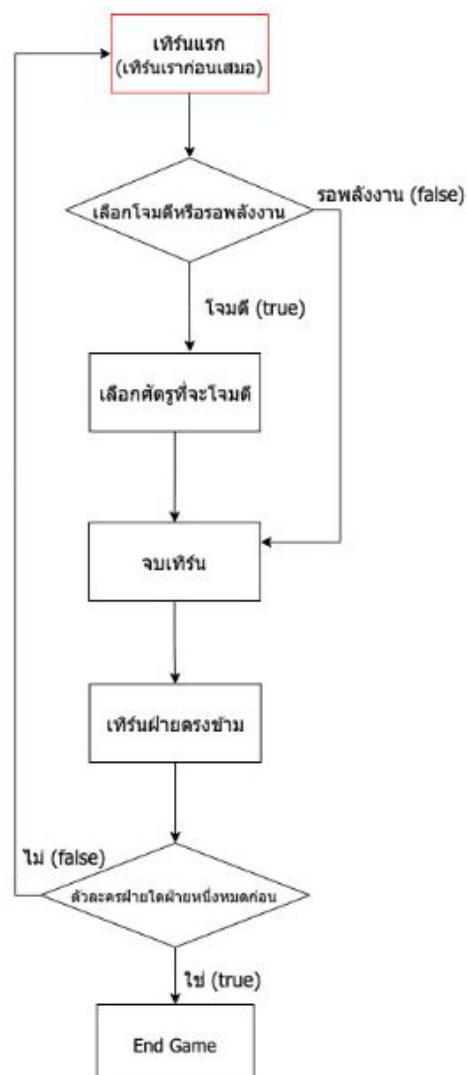
### 2. Use polymorphism

- class character แบ่งคลาสย่อยเป็นclass Robot และclass enemy และแบ่งเป็นคลาสย่อยตามประเภทลงไปอีก

### 3. Exception Handling

- ดักข้อผิดพลาดจากการรับค่า (cin) ทุกจุดในโปรแกรม

## แผนผังการทำงานของเกม



## โครงสร้างของเกม

### class character

class character เป็นคลาสแม่ของ Robot และ Enemy ซึ่งหมายถึงตัวละครทั้งหมดในเกมส์ ทำหน้าที่เก็บและแสดงผลค่าสถานะต่างๆ คือ Name, Hp(health points), Df(defense) และ Atk(attack)

### class Robot:public character

class Robot เป็นคลาสแม่ของ robot1 robot2 และrobot3 ซึ่งหมายถึงตัวละครที่ผู้เล่นสามารถควบคุมการกระทำได้ คลาสนี้ทำหน้าที่จัดเก็บและแสดงผล รวมถึงคำนวณค่าสถานะพิเศษที่มีเฉพาะในตัวละครหุ่นยนต์ซึ่งคือ En (Energy)

### class robot1:public Robot

class robot1 ทำหน้าที่ทำหน้าที่กำหนดค่าพลังของตัวเองส่งไปไว้ที่คลาสแม่ เก็บรูปร่างของตัวrobot1ไว้ใช้สำหรับแสดงผลบนหน้าจอ และแสดงผลตัวเลือกเฉพาะของตัวเอง

### class robot2:public Robot

class robot2 ทำหน้าที่ทำหน้าที่กำหนดค่าพลังของตัวเองส่งไปไว้ที่คลาสแม่ เก็บรูปร่างของตัวrobot2ไว้ใช้สำหรับแสดงผลบนหน้าจอ และแสดงผลตัวเลือกเฉพาะของตัวเอง

### class robot3:public Robot

class robot3 ทำหน้าที่ทำหน้าที่กำหนดค่าพลังของตัวเองส่งไปไว้ที่คลาสแม่ เก็บรูปร่างของตัวrobot3ไว้ใช้สำหรับแสดงผลบนหน้าจอ และแสดงผลตัวเลือกเฉพาะของตัวเอง

### class enemy:public character //ฝั่งศัตรู

class enemy ทำหน้าที่เก็บประเภทของศัตรู และใช้สำหรับแสดงผลการโจมตีของฝั่งศัตรู

### class human:public enemy{

class human ทำหน้าที่กำหนดค่าพลังของตัวเองส่งไปไว้ที่คลาสแม่ และเก็บรูปร่างของตัวhumanไว้ใช้สำหรับแสดงผลบนหน้าจอ

### class fat\_human:public enemy{

class fat\_human ทำหน้าที่กำหนดค่าพลังของตัวเองส่งไปไว้ที่คลาสแม่ และเก็บรูปร่างของตัวfat humanไว้ใช้สำหรับแสดงผลบนหน้าจอ

### class soldier\_human:public enemy{

class fat\_human ทำหน้าที่กำหนดค่าพลังของตัวเองส่งไปไว้ที่คลาสแม่ และเก็บรูปร่างของตัวfat humanไว้ใช้สำหรับแสดงผลบนหน้าจอ

```
class super_human:public enemy{
```

```
    class fat_human ทำหน้าที่กำหนดค่าพลังของตัวเองส่งไปไว้ที่คลาสแม่ และเก็บรูปร่างของตัวfat  
humanไว้ใช้สำหรับแสดงผลบนหน้าจอ
```

## วิธีการเล่น

- หลังจากเข้าไปในด่าน เมื่อเริ่มเทิร์นและเข้าสู่ช่วงของหุ่นยนต์แล้วก็จะสามารถเลือกการกระทำของหุ่นยนต์ได้ โดยการกรอก 0 เพื่อฟื้นฟูพลังงาน กรอก 1 เพื่อโจมตีและจะมีรายชื่อศัตรูขึ้นมาให้เลือก ก็ให้กรอกตัวเลขตามศัตรูที่เลือกไป และกรอก 2 เพื่อใช้ทักษะ โดยการโจมตีจะใช้พลังงาน 1 หน่วย และการใช้ทักษะจะใช้พลังงาน 2 หน่วย โดยลำดับของหุ่นยนต์จะเรียงดังนี้ คือ C3P0 R3D2 และJUNKCAT เมื่อเสร็จการกระทำของหุ่นยนต์แล้วก็จะเข้าสู่ช่วงของศัตรู ศัตรูจะสุ่มโจมตีหุ่นยนต์ทีละคน และเมื่อจบช่วงของศัตรูก็จะเข้าสู่เทิร์นถัดไป
- หากฝ่ายหุ่นยนต์สามารถกำจัดศัตรูได้ทั้งหมดก็จะถือว่าชนะระลอกนี้ และเข้าสู่ระลอกถัดไป หากชนะทุกระลอกแล้วก็จะถือว่าเคลียร์ด่านนี้ แต่ถ้าหากหุ่นยนต์พลังชีวิตหมดทุกตัวหรือไม่สามารถกำจัดศัตรูได้ทั้งหมดภายใน 10 เทิร์นก็จะถือว่า game over

## ข้อจำกัด

1. การเพิ่มจำนวนหุ่นยนต์ให้มากกว่า 3 ชนิด รวมถึงการแสดงผลหุ่นยนต์บนหน้าจอมากกว่า 3 ตัว จะต้องปรับแก้โปรแกรมหลายจุด
2. พวกศัตรูถูกออกแบบมาให้ทำได้แค่โจมตีเท่านั้น หากต้องการให้ศัตรูมีทักษะมากกว่าแค่การโจมตีจะต้องปรับแก้โปรแกรมหลายจุด
3. การทำงานของโปรแกรม จะต้องส่งผ่านการใส่ตัวเลขลงไปเท่านั้น
4. หากทำการส่งค่าเข้าไปก่อนที่โปรแกรมจะขอจะทำให้โปรแกรมทำงานผิดพลาดได้
5. ถ้ากดเริ่มเกมไปแล้วไม่สามารถออกกลางคันได้จนกว่าจะชนะเกมหรือแพ้

## สรุป

จากการทำเกม Energy war ทำให้ทางกลุ่มพบว่า การเขียนเกมเพื่อดำเนินกระบวนการต่างๆ สำหรับเกมแนว Turn-base นั้น ถือเป็นสิ่งที่มีความซับซ้อนเป็นอย่างยิ่ง โดยเฉพาะเมื่อต้องการที่จะเลือกโจมตีหรือกระทำการใดๆต่อตัวละครต่างๆโดยเฉพาะเจาะจงในเกม ซึ่งทางกลุ่มเลือกใช้ Link lists เข้ามาช่วย เพื่อช่วยให้การกระทำต่อตัวละครที่เป็นรายตัวมีความคล่องตัวและยืดหยุ่นมากขึ้น นอกจากนี้ ในส่วนของการแสดงผลของตัวละครก็ถือเป็นสิ่งที่ซับซ้อนเช่นกัน เนื่องจากต้องสร้าง Array ขึ้นมาหลายครั้ง เพื่อที่จะทำการแสดงผลให้เกิดตัวละครที่สามารถขยับตัวได้เมื่อถูกกระทำต่าง ๆ ดังนั้น การเขียนเกม Energy war จึงเป็นการฝึกทักษะของทางด้านความคิด การคิดวิเคราะห์ และการทำงานร่วมกันเป็นทีมได้เป็นอย่างดี

## หน้าที่ที่รับผิดชอบ

นายวัชรศัณย์ พรหมณี 6413110

- ดูแล function battle.h

นายศศิศ ศรียรัตน์ 6413112

- ดูแลส่วน class

นายกวิน เก่งเกตุ 6413210

- ดูแลส่วน link listed

นายวิพล ฉายรักษา 6413223

- ดูแล screen และ วาดรูป

หมายเหตุ : การแบ่งหน้าที่ในกลุ่มเราเป็นการแบ่งหน้าที่หลัก ๆ เท่านั้น ในการจะดำเนินการในแต่ละ function จำเป็นต้องมีการสื่อสารและช่วยกันเลือกใช้ function ที่ไม่ได้อยู่ในหน้าที่ของตนด้วย