Assignment 3: Further Design and Implementation

Learning Outcomes

In this assignment, you will design and implement some new game functionality. This assignment is intended to develop and assess the following unit learning outcomes:

- **LO1**. Iteratively apply object-oriented design principles to design small to medium-sized software systems using standard software engineering notations, namely UML class diagrams and UML interaction diagrams.
- **LO3**. Apply object-oriented programming constructs, such as abstraction, information hiding, inheritance, and polymorphism, to implement object-oriented designs using a programming language (namely, Java).
- **LO4.** Apply effective programming strategies to refactor and debug object-oriented implementations systematically and efficiently using available programming language tools.
- **LO5**. Apply principles of software engineering practice to create object-oriented systems with peers using tools, including integrated development environments (IDEs), UML drawing tools, and version control systems.

To demonstrate your ability, you will be expected to:

- design and implement further extensions to the system
- use an integrated development environment to do so
- update your UML class diagrams as required to ensure that they match your implementation
- use git to manage your team's files and documents

The marking scheme for this assignment reflects these expectations.

Project Requirements

In this assignment, you will design and implement **four (4)** fixed requirements and **one creative requirement**. Note that the CREATIVE requirement **IS NOT OPTIONAL**, so you **MUST** include it.

1. Fixed Requirements

You must **design and implement four (4) fixed requirements (for a group of 3) or three (3) fixed requirements (for a group of 2). These requirements include requirements 1 to 4.**

- Group of 3: REQ 1, REQ 2, REQ 3, REQ 4 + 1 Creative requirement
- Group of 2: REQ 1, REQ 2, REQ 3 + 1 Creative requirement
- Group of 1 [special case*]: REQ1, REQ2 + 1 Creative requirement

* Generally, you should not work alone on this assignment. A group of 1 will only be assigned based on the decision of the admin TA and the lecturer for very exceptional circumstances.

2. Creative Requirement

We hope that you enjoy working on the assignment so far! We also hope that you are comfortable working with the engine code. Since this unit is about design, we give you the opportunity to do a bit of creative work! In this part, you can be creative in designing and implementing your own requirements. However, **you must strictly follow the rules and standards** that we set so that we can grade your work fairly.

The following rules apply:

- To ensure that your creative requirement meets our standards, you must fill out the form below and answer the following questions:
    - What is the title of the requirement?
    - What is the summary of the feature?
    - How and why does your proposed requirement adhere to the SOLID principles?
    - The new requirement must use at least two classes from the engine package. Which classes are you going to use, and how will you use them in the design & implementation?
    - The new requirement must use/re-use at least one (1) existing feature (either from assignment 2 and/or fixed requirements from assignment 3). Which feature(s) are you going to use, and how will you use them in the design & implementation?
    - The new requirement must use existing or create new abstractions (e.g. abstract classes or interfaces, apart from the engine code). Which abstractions are you going to create/re-use, and how will you use them in the design & implementation?
    - The new requirement must use existing or create new capabilities. Which capabilities are you going to create/re-use, and how will you use them in the implementation?

- You must ensure new features do not affect the existing system (old features).

Google Form Link for Creative Requirement

You can fill out the form mentioned above here: https://forms.gle/WUxy9ydvrvfLCcuo6

> ⚠ Before filling out the form above, you may want to discuss your idea with your TAs to ensure that the proposed requirement meets our standard.

Examples

Here are several general examples to give you some ideas. You **may pick <u>one</u>** of the following ideas, or feel free to create your feature as long as you can complete the table above.

> ⚠ Some ideas below might require some modification to be considered sufficiently complex. Again, please discuss the idea with your TA before filling out and submitting the form!

- Boss battle against Margit, The Fell Omen, Malenia, Blade of Miquella, or any other bosses
- Two-phase boss battle against Radagon of the Golden Order and the Elden Beast
- New weapons with status effects (e.g. poison status effect will damage the enemy for several turns, sleep status effect will make the enemy stay still in a location, etc.)
- New map with grounds that have status effects (e.g. Lake of Rot that will damage the player over time if the player is standing on top of it)
- New consumable items with varying effects (e.g. Golden seeds that can be consumed to increase the number of uses the Flask of Crimson Tears has, boluses that can remove status effects mentioned above, etc.)
- Conversation with Trader or other NPCs
- Splitting the original map into North-East, North-West, South-East and South-West instead of just East and West and adding new types of enemies
- And more...

REQ1: Travelling between Maps

If you are a group of 3, this requirement is **mandatory**.

If you are a group of 2, this requirement is **mandatory**.

Scenario

In Elden Ring, The Lands Between consists of five vast areas, each ruled by a demigod.

In this assignment, we will only focus on the first area of Elden Ring, which is known as Limgrave. In fact, Limgrave is the area that we used back in assignments 1 and 2. Here, we will expand this area further.

A. New grounds

1. Cliff `+`

- It is represented as `+`. Limgrave and other areas of The Lands Between have many cliffs. To make the game even harder for the player, If the player steps on a cliff, they will instantly get killed.
- Note that when the player falls off a cliff, they should still be able to recover their runes since the runes will be dropped on the last location before the player dies (see Assignment 1 specification).

Note that other non-playable actors/enemies cannot fall off a cliff.

2. Golden Fog Door `D`

- The golden fog door is represented as `D`. It allows the player to travel to other parts of Limgrave.

Note that only the player is allowed to use the fog door to travel between maps.

The original map that we used back in assignments 1 and 2 is the general area of Limgrave, so for now, let's call it Limgrave.

The **Limgrave** map has 2 doors:

- One door goes to the Roundtable Hold map, which is a safe place with no enemy.
- One door goes to the Stormveil Castle map

The **Stormveil Castle** map has 2 doors:

- One door goes back to the general Limgrave area
- One door will bring the player to the room where the player will fight against the demigod ruling Limgrave.

The **Roundtable Hold** has 1 door:

- This door will bring the player back to the general Limgrave area

The **boss room** does not have any doors.

B. Fast Travel (OPTIONAL)

Each map described above will have at least one site of lost grace.

- One in Limgrave, which will be called "The First Step"
- One in the Roundtable Hold, which will be called "Table of Lost Grace"
- One in Stormveil Castle, which will be called "Stormveil Main Gate"
- One in the boss room, which will be called "Godrick the Grafted". This one is special, as it will only be added after the demigod is killed (see requirement 3)

The first time the player visits a site of lost grace, the player should be given the option to activate it. Otherwise, the player will not be able to rest on the site of lost grace.

> You can assume that "The First Step" site of lost grace has been activated **before** the game starts.

When the player is on a lost site of grace, the player should be able to travel to any of the sites of lost grace that the player has visited/activated (except for the current site of lost grace since the player is already there).

If the player dies, the player should be respawned/revived on the last site of grace that the player **rested on**. This means that if the player only visited/activated the site of lost grace without resting on it, then the player will not respawn there, if the player only fast travel to the site of lost grace without resting on it, then the player will not respawn there, and so on.

> You can assume that at the start of the game, the player has rested on "The First Step" site of lost grace. So, if the player has not rested on any other site of lost grace, you can always bring the player back to live on "The First Step" site of lost grace.

> You can also assume that only the player can use site of lost grace. Other actors cannot use site of lost grace (including for travelling and resting).

Implementation Expectations

- You might want to have the site of lost grace surrounded by floors to avoid enemies walking on top of the ground, which may crash the game if the player teleports to it without checking whether there is another actor on the ground.

- The engine will continue running the other maps when the player moves to a new map.
- Print something when the site of lost grace is discovered, e.g. LOST GRACE DISCOVERED

```
"\n" +
"
"
"`7MMF'        .g8\"\"8q.    .M\"\"\"bgd MMP\"\"MM\"\"YMM       .g8\"\"\"bgd `7MM\"\"\"Mq.        d
"  MM        .dP'    `YM. ,MI    \"Y P'   MM   `7     .dP'     `M   MM   `MM.      ;MM:    .dP'
"  MM        dM'      `MM `MMb.          MM          dM'       `   MM   ,M9      ,V^MM.   dM'
"  MM        MM        MM   `YMMNq.      MM          MM            MMmmdM9      ,M  `MM   MM
"  MM      , MM.      ,MP .     `MM      MM          MM.    `7MMF' MM  YM.      AbmmmqMA  MM.
"  MM     ,M `Mb.    ,dP' Mb     dM      MM          `Mb.     MM   MM   `Mb.   A'     VML `Mb.
".JMMmmmmMMM   `\"bmmd\"'   P\"Ybmmd\"     .JMML.          `\"bmmmdPY .JMML. .JMM..AMA.   .AMMA. `\
"
"
"\n";
```

- You may use the following string to create the maps listed above:
  - Limgrave (slight modification from assignments 1 & 2)

> You can add the grounds manually with the `setGround` method of the Location class, e.g. `gameMap.at(x, y).setGround(...)`, or add the Ground display character directly to the map below.

```
".....................#.............#...........................+++.........",
".....................#.............#.......................+++++..........",
".....................#..___....____#.........................+++++........",
".....................#...........__#............................++........",
".....................#_____........#..............................+++......",
".....................#............_#...............................+++.....",
".....................######...######.....................................",
"........................................................................",
"..........................=.............................................",
"........++++.....................###___###...............................",
"........+++++++..................________#...............................",
"..........+++....................#________...............................",
"............+++..................#_______#...............................",
"............+....................###___###...............................",
"............++.....................#___#...................................",
"..............+..................=.......................................",
"..............++.................................................=.........",
"..............................................++..........................",
"..................++++.....................+++...............######..##...",
"#####___######....++..........................+++............#....____....",
"_____________#.....++++.........................+..............__.....#...",
"_____________#.....+....++.......................++..........._.....__.#...",
"_____________#.........+..+.....................+++...........###..__###...",
"_____________#.............++.............................................."
```

  - Stormveil Castle

```
".....................................................................",
".................<.............<.....................................",
".....................................................................",
"#############################################..#########################",
".........................#.............#.......B.............B......",
".....B.............B.....#.............#.........................",
"............................<.........<..............................",
".....B.............B.....#.............#.......B.............B......",
".........................#.............#.........................",
"######################..#############...############.####..#########...#####",
"..............#++++++++++++#.............#++++++++++++#.............",
"..............#++++++++++++...<.........<...#++++++++++++#.............",
"..............#++++++++++++.................++++++++++++#.............",
"..............#++++++++++++#.............#++++++++++++#.............",
"#####...##########.....#############...#############..#############...#####",
"..._______.....................B......B.....................B.....B...",
"_____..._..____....&&........<.............<.........................",
".........____......&&.....................................................",
"...._______.................<.............<.................<.....<...",
"#####....##...###..#####...##########___###############......##.....####...",
"++++++++++++++++++++++++++#.................#++++++++++++++++++++++++++",
"++++++++++++++++++++++++++.................#++++++++++++++++++++++++++",
"++++++++++++++++++++++++++#.................++++++++++++++++++++++++++",
"++++++++++++++++++++++++++#.................#++++++++++++++++++++++++++"
```

- Roundtable Hold

```
"###################",
"#_________________#",
"#_________________#",
"#_________________#",
"#_________________#",
"#_________________#",
"#_________________#",
"#_________________#",
"#_________________#",
"########___#######"
```

- Boss room

```
"++++++++++++++++++++++++",
".......................",
"..=....................",
".......................",
".......................",
".......................",
".......................",
".......................",
"++++++++++++++++++++++++"
```

REQ2: Inhabitant of the Stormveil Castle

If you are a group of 3, this requirement is **mandatory**.

If you are a group of 2, this requirement is **mandatory**.

Scenario

The Stormveil Castle is a sub-region of the Limgrave area. It is known to be the stronghold of the demigod, known as Godrick the Grafted. Just like in the general Limgrave area, there are hordes of Godrick soldiers and dogs that will attempt to stop the player from progressing through the map and defeating the demigod ruling Limgrave.

A. Grounds

1. Cage `<`

- It is represented as `<` and it spawns Dogs with a 37% chance at each turn.

2. Barrack `B`

- It is represented as `B`. It spawns Godrick Soldiers with a 45% chance at each turn.

B. Enemies

1. Dog `a`

- A hostile creature, represented by `a`, that has 104 hit points and bites other creatures, dealing 101 damage with 93% attack accuracy.
- At each turn, dogs have a 10% chance of being despawned (removed from the map) unless they are following the player
- When the game is reset, dogs will be wiped from the map
- Generates 52 - 1390 runes when defeated by the player
- **Since these dogs are raised in the Stormveil Castle, they will not attack Godrick soldiers, but they will attack Giant Dogs and Lone Wolves if these creatures ever enter the castle.**

You can test this feature by adding a Gust of Wind near a Cage, and see whether the Dog would attack LoneWolf/GiantDog.

2. Godrick Soldier `p`

- The soldier of Godrick, represented by `p`, that has 198 hit points and carries around a ranged weapon, Heavy Crossbow (see below). The Heavy Crossbow will be dropped by the soldier of Godrick when they are defeated by the player.
- At each turn, the Godrick soldiers have a 10% chance of being despawned (removed from the map) unless they are following the player
- When the game is reset, Godrick soldiers will be wiped from the map
- Generate 38 - 70 runes when defeated by the player
- **Since these soldiers own the dogs raised in the Stormveil Castle, they will not attack Dogs, but they will attack Giant Dogs and Lone Wolves if these creatures ever enter the castle.**

> i You can test this feature by adding a Gust of Wind near a Barrack, and see whether the Godrick Soldier would attack LoneWolf/GiantDog

Similar to assignments 1 & 2,

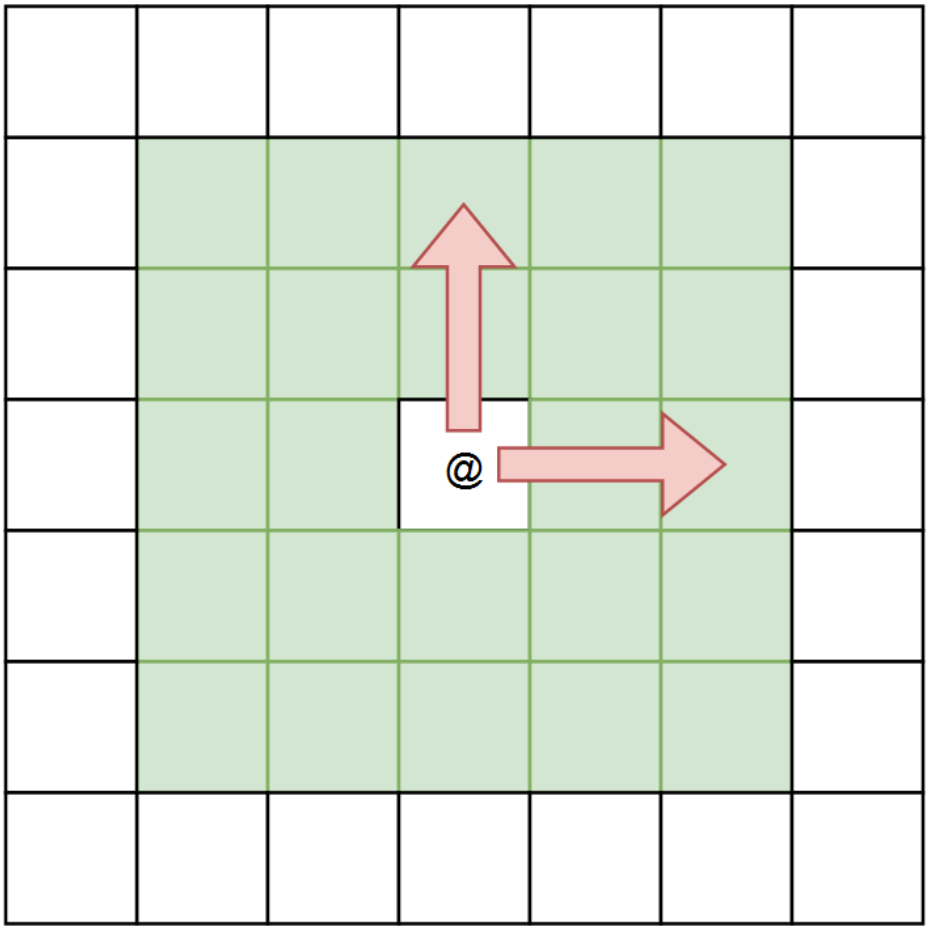
- If the weapon that the enemy holds has a special skill, the enemy has a 50% chance of using the weapon skill.
- If an enemy is close (one block away) to the player, they will follow the player. Otherwise, they will wander around the map

C. Weapons (OPTIONAL)

> i You can add any of the existing weapon for Godrick Soldier. Please mention the weapon that you add for the enemy in your design rationale.

1. Heavy Crossbow

- A crossbow, represented by `}`, carried around by the Godrick soldiers. It deals 64 damage with 57% attack accuracy.
- This weapon can be sold for 100 runes, and it can be purchased for 1500 runes.
- The weapon does not have a special skill, but it can shoot an enemy 2 blocks away from the holder (see the diagram below)

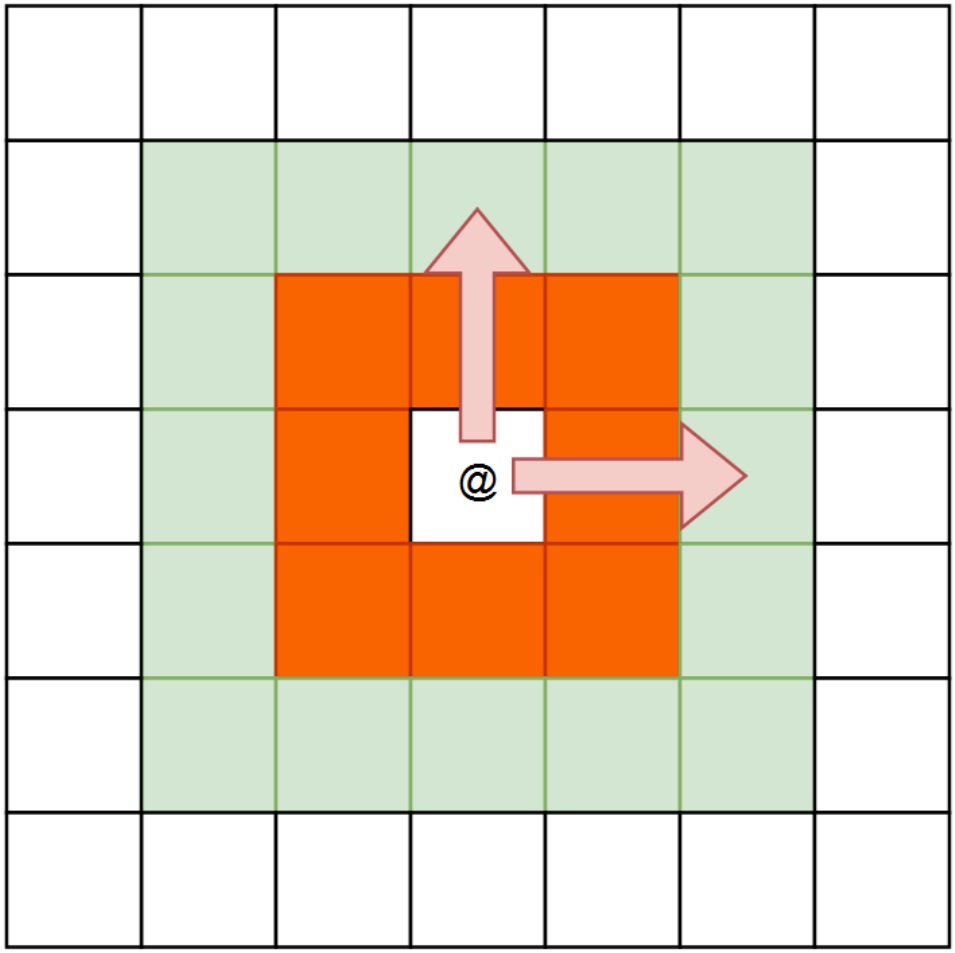


Anything in the green boxes can be attacked by the holder of the heavy crossbow.

Implementation Expectations

- Make sure the range attack does not go beyond the map
- Make sure other friendly actors cannot be targeted by this weapon
- (**OPTIONAL**) for range attack, you may only want to check target locations that are not within the exits of the user, e.g. target locations are anything within range 2 until x, since range 1 is within the user exits, and the exits can be handled by attack behaviour/action. This is only to avoid showing an option to attack an enemy within the surrounding of the player twice.

This is just for a better game experience. if you don't have enough time, you don't need to do this.



Anything in the green box can be handled by the ranged attack, while anything in the orange box can be handled by the normal attack action.

REQ3: Godrick the Grafted

If you are a group of 3, this requirement is **mandatory**.

If you are a group of 2, this requirement is **mandatory**.

Scenario

Here it is! The first boss battle in the game of Elden Ring. You will be facing off against the ruler of Limgrave, Godrick the Grafted.

A. Godrick the Grafted `Y` **(OPTIONAL)**

- A demigod, represented by `Y`, that has 6080 hit points.
- This enemy cannot be removed from the map unless defeated by the player, i.e. there is no despawn chance.
- When the game is reset, this enemy will go back to their initial position with full health if they have not been defeated.
- This boss has 2 phases:
    - The first phase is when their hit points are above (greater than) 50% of their max hit points
        - Godrick starts with the weapon: Axe of Godrick, which has a special skill that damages anything in the holder's surroundings. Similar to other weapons that have special skills, this weapon can do normal attacks as well.
        - It deals 142 damage with 84% attack accuracy. It is represented by `T`.
        - Not purchasable, but sellable for 100 runes.
    - The second phase is when their hit points are below or equal to 50% of their max hit points
        - When the second phase is reached, the Axe of Godrick weapon will be removed from Godrick's inventory. It will be swapped with Grafted Dragon.
        - This weapon is similar to the Axe of Godrick as it has a special skill that damages anything in the holder's surroundings. This weapon can also do normal attacks.
        - It deals 89 damage with 90% attack accuracy. It is represented by `N`
        - Not purchasable, but sellable for 200 runes.
        - (**OPTIONAL**) This weapon can also drop fire in the surroundings of the holder with a 75% chance at each turn. Only dirt can be burned. Fire, represented by `u`, stays on the ground for 3 turns and deals 57 damage to any other actor that is not the holder of the weapon. The holder themselves cannot be damaged by the fire. If an enemy is killed by the fire generated by the player, the runes will go to the player directly.

The "dropping fire" feature is only for improving the gameplay experience. Therefore, it is optional. Only implement this if you have the time.

- After their defeat,
  - The weapons will not be dropped. Instead, a Remembrance of the Grafted, represented by `0`, will be dropped. This item cannot be purchased but can either be sold to any trader for 20000 runes or exchanged with **a trader that can accept remembrances for either one of Godrick's weapons (not all traders can)**.
  - 20000 runes will be given to the player
  - The ground where Godrick last stood will change to a site of lost grace (see requirement 1)

B. More Selling and Purchasing

1. Golden Runes `*`

- An item, represented by `*`, that cannot be purchased. Instead, this item can be found scattered across the maps. Therefore, it can be picked up or dropped off by the player.
- It can be consumed. When consumed, this item can generate any amount of runes, ranging from 200-10000 runes.

2. Finger Reader Enia `E`

If you do not plan to implement REQ3A (Godrick the Grafted) above, please create the two weapons of Godrick the Grafted above **without any special skills** and the Remembrance of the Grafted to implement the exchange item feature below.

- A new trader, represented by `E`, which can accept the Remembrance of the Grafted from the player to be exchanged for **either** the Axe of Godrick or Grafted Dragon.
- The player cannot purchase weapons from this trader, but they can still sell anything that is sellable to this trader.

3. Site of Lost Grace [**OPTIONAL**]

- To make the game a bit easier to play, the player can upgrade their max hit points by purchasing it from the site of lost grace (only once the site of lost grace has been discovered).
- The purchase cost for upgrading the player's max hit points is 200 runes. One upgrade will increase the player's max hit points by 48.
- The price will increase by 100 every time the player purchases this (regardless of the site of lost grace, e.g. if the player upgrades their hit points upgrade in the site of lost grace A by paying 200 runes, then if the player goes to the site of lost grace B, there is an option to upgrade the maximum hit points by paying 300 runes)

You can assume that when the max hp increases, the player's health will go back to full.

The "increasing price" feature is only for improving the gameplay experience. Therefore, it is optional. Only

implement this if you have the time.

Implementation Expectations

- You can put Godrick the Grafted in the middle of the boss room map in Application.java
- You can scatter Golden Rune anywhere across Limgrave & the Stormveil Castle
- Since Kale and Enia are put inside floors, they won't be able to be burned by the special attack
- If the probability is met, the fire will be dropped within all of the exits of the holder, not independent chance, i.e. one fire might drop, another might not - all should drop at the same time
- Print something when Godrick is defeated, e.g. DEMIGOD FELLED

```
"\n" +
"
"
"`7MM\"\"\"Yb. `7MM\"\"\"YMM  `7MMM.     ,MMF'`7MMF' .g8\"\"\"bgd    .g8\"\"8q. `7MM\"\"\"Yb.
"  MM    `Yb. MM    `7    MMMb    dPMM    MM .dP'     `M  .dP'    `YM. MM    `Yb.       MM    `7
"  MM     `Mb MM   d      M YM   ,M MM    MM dM'       `  dM'      `MM MM     `Mb       MM   d
"  MM      MM MMmmMM      M  Mb  M' MM    MM MM           MM        MM MM      MM       MM\"\"MM
"  MM     ,MP MM   Y  ,   M  YM.P'  MM    MM MM.    `7MMF'MM.      ,MP MM     ,MP       MM   Y
"  MM    ,dP' MM     ,M   M  `YM'   MM    MM `Mb.     MM  `Mb.    ,dP' MM    ,dP'       MM
".JMMmmmdP' .JMMmmmmMMM .JML. `'  .JMML..JMML. `\"bmmmdPY    `\"bmmd\"' .JMMmmmdP'       .JMML.
"
"
"\n";
```

REQ4: A Guest from Another Realm

If you are a group of 3, this requirement is **mandatory**.

If you are a group of 2, this requirement is **optional**. You may decide to implement this requirement to develop a complete game. However, the marker will not assess the design and implementation of this requirement, i.e. no extra marks will be given, and no marks will be deducted if your design and implementation do not meet the marking criteria. If you, as a group of 2, work on this requirement and want some feedback on your design, please let your TA know (your mark will not be affected).

Scenario

In the game of Elden Ring, the player may ask for assistance from another tarnished to make progressing through the game less challenging. However, be careful when using this system, as you may accidentally invite an invader into your world!

A. New Role (Combat Archetype)

1. Astrologer

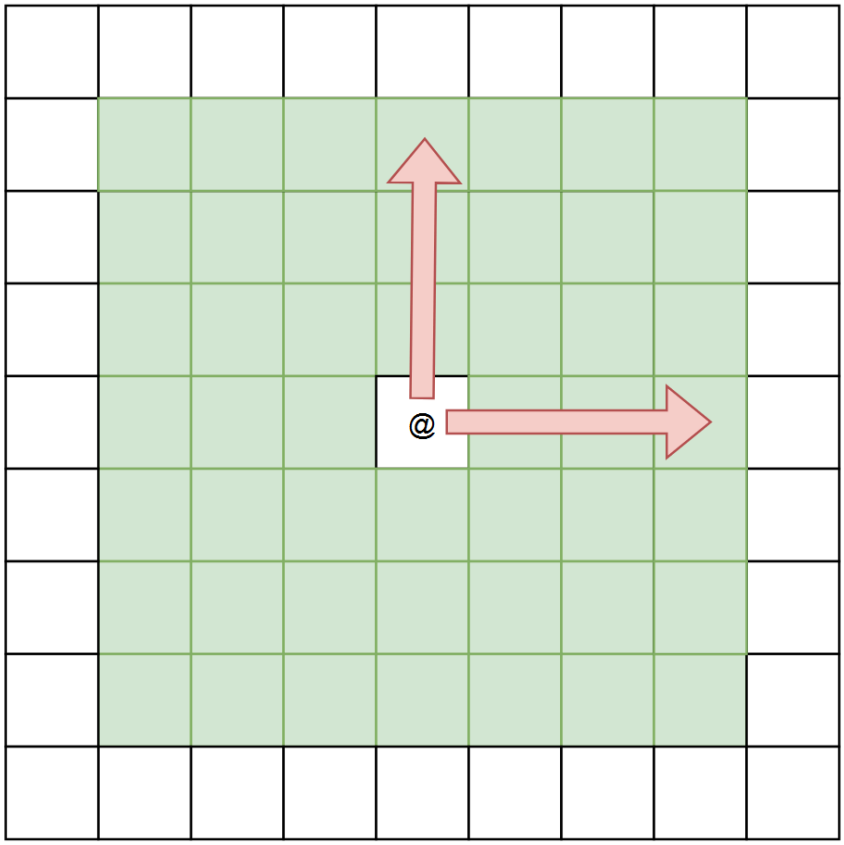
If the player chooses the Astrologer class, they will start the game with Astrologer's Staff as their starting weapon. Their starting hit point will be 396.

B. New Weapon **(OPTIONAL)**

You can add any of the existing weapon for the new Astrologer role. Please mention the weapon that you add for the role in your design rationale.

1. Astrologer's Staff `f`

- A staff, represented by `f`, that deals 274 damage with 50% attack accuracy.
- This weapon can be sold for 100 runes and can be purchased for 800 runes.
- Similar to the Heavy Crossbow in requirement 2, this weapon does not have a special skill, but it is able to shoot an enemy 3 blocks away from the holder (see the diagram below)



Anything in the green boxes can be attacked by the holder of the Astrologer's Staff.

C. Allies/Invaders

1. Summon Sign `=`

Allies or Invaders could be summoned from the Summon Sign, represented by `=`. There is a 50% chance that the player will spawn an Ally and a 50% chance that the player will spawn an Invader. The player will need to be on top of the summon sign (or within its surroundings) to interact with it (Allies/Invaders will not be spawned automatically into the game, the player must intentionally summon them).

You need to find an empty spot within the exits of the summon sign so that they can be safely spawned into the game.

When spawned/summoned into the world, an Ally or an Invader can start with any of the available roles (chosen randomly). For example, an Ally can be spawned as a Bandit, Samurai, Wretch or Astrologer. They will start with the starting weapon and hit points of the given role.

1. Ally `A`

When spawned/summoned into the world, an Ally can start with any of the available roles (chosen randomly).

Ally, represented by `A`, can help the player defeat enemies or Invaders. They should not attack the player or other friendly actors, e.g. Traders or other allies. Similarly, the Player should not attack Ally.

Ally, however, can attack other hostile creatures.

They can be removed from the map ONLY if the player dies. If the player rests, do not remove them from the map.

> Removing them only when the player dies, not when the player rests. Have we done this somewhere before?

Once they die, they cannot be respawned. The player, however, may decide to spawn/summon another Ally.

For simplicity, Ally cannot buy or sell items, cannot interact with the site of lost grace, cannot fast travel, and cannot retrieve runes. Ally can wander around, but they will not follow the player. When Ally dies, the weapon carried by them will be dropped.

2. Invader `▯`

When spawned/summoned into the world, an Invader can start with any of the available roles (chosen randomly).

Invader, represented by `▯`, is similar to Ally, except that they can attack the player. Invader, like Ally, can also attack other hostile creatures. Invader cannot attack another Invader.

They can be removed from the map ONLY if the player dies. If the player rests, do not remove them from the map.

> Removing them only when the player dies, not when the player rests. Have we done this somewhere before?

Once they die, they cannot be respawned. The player, however, may accidentally spawn/summon another Invader.

Invader will follow the player if the player is within the exits of the invader. If there is no player within their surroundings, the invader will wander around. Note that, like other enemies, Invader cannot enter floor.

Defeating an invader grants you 1358 - 5578 runes.

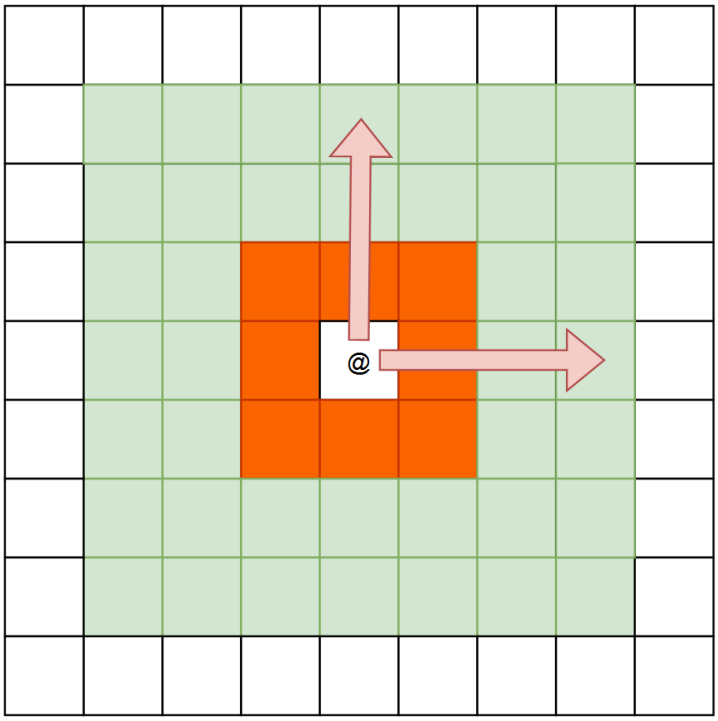
For simplicity, Invader cannot buy or sell items, cannot interact with the site of lost grace, cannot fast travel, and cannot retrieve runes. When Invader dies, the weapon carried by them will be dropped.

Implementation Expectations

- Don't forget to add Astrologer as a new option in the console menu for role selection at the start of the game
- Make sure the range attack does not go beyond the map
- Make sure other friendly actors cannot be targeted by this weapon
- (**OPTIONAL**) for range attack, you may only want to check target locations that are not within the exits of the user, e.g. target locations are anything within range 2 until x, since range 1 is

within the user exits, and the exits can be handled by attack behaviour/action. This is only to avoid showing an option to attack an enemy within the surrounding of the player twice.

This is just for a better game experience. if you don't have enough time, you don't need to do this.



Anything in the green box can be handled by the ranged attack, while anything in the orange box can be handled by the normal attack action.

REQ5: Creative requirement

Please refer to this page for an explanation about this feature:
https://edstem.org/au/courses/10331/lessons/30073/slides/211789

Google Form Link for Creative Requirement

You can fill out the form mentioned above here: https://forms.gle/WUxy9ydvrvfLCcuo6

Assignment Expectations

Design is important

One of the primary aims of this unit is for you to learn the fundamentals of object-oriented design. In order to get a high mark on this assignment, it will not be sufficient for your code to "work". It must also adhere to the design principles covered in lectures and in the required readings on Ed Lesson.

If you would like informal feedback on your design at any time, please consult your TA in a Applied Session or attend any consultation session.

You might find that some aspects of your existing design need to change to support the new functionality. You might also think of a better approach to some of the requirements you have already implemented — your understanding of the requirements and codebase will have improved as you have progressed. You might also spot places where your existing code (and thus design) can benefit from refactoring.

If you want to update your design, you may do so; if you decide to do this, update your design documents to match the code and write a brief explanation of your changes and the reasons behind them. This will help your marker understand the thinking behind your code.

Coding and commenting standards

You must adhere to the Google Java coding standards (covered in Ed Lesson).

Write Javadoc comments for *at least* all public methods and attributes in your classes.

You will be marked on your adherence to the standards, Javadoc, and general commenting guidelines (covered in Ed Lesson).

To ensure that your work adheres to good coding practices in this unit, we encourage you to minimise the use of `instanceof` and/or downcasting, as they are considered code smells. It's important to note that if there are any instances where you need to use them, please provide appropriate justifications in code comments or design rationale. We believe learning how to properly utilise polymorphism is crucial in addressing this code smell, and we are committed to teaching you how to do so effectively.

Submission Instructions

You are expected to produce the following artefacts in Assignment 3:

- New class diagrams or updated class diagrams with new features: distinguish between existing classes and new classes with different colours
- Design rationale: why this approach is good? what's the alternative?
- Updated Design Report in the `.md` or `.pdf` format (including the two items above).
- Implementation (the code) of the extension of the system
- Contribution Logs

NOTE: SEQUENCE DIAGRAMS ARE OPTIONAL (NOT REQUIRED)!

Additionally, you will be asked to complete the following forms:

- Peer Assessment Form (mandatory, useful for markers and for you to reflect on your group dynamics)
- Feedback Request Form (to be completed in order to upload your files to Moodle, see instructions here).

We will mark your Assignment 3 on the state of the `main` branch of your Monash GitLab repository at that time. If you have done any work in other branches, make sure you merge it into `main` before the due date. Additionally, please, make sure at least one team member submits the assignment to Moodle. This step is compulsory.

> ⚠ **Do not create a new copy of your work for Assignment 3.** Continue working on the same files, in the same directory structure (you might like to add a tag to your final Assignment 2 commit before starting on Assignment 3, so you can find that version easily). *https://git-scm.com/book/en/v2/Git-Basics-Tagging*

As we said above, you may update your design and implementation if you find that your Assignment 2 solution is not suitable for extension or if you think of a better approach during Assignment 3.

It is highly recommended that you explicitly Indicate in the **design rationale** what changed in the design between Assignment 2 and Assignment 3. Explaining the changes would help you organise your ideas to explain how the design changed during your handover interview. A quick summary of changes with some explanation regarding why you applied such changes would be really valuable.

Contribution Logs

We require you to have a document to know how you plan to divide the work between team members and simultaneously capture individual contributions. We have prepared a template you can copy to your university's Google Drive and share with your team. Please follow the detailed instructions **here.**

Since your markers need to access this information, make sure that you create a share link and put it in the README of your repository. Failure to do so will result in 0 mark.

In this matter, everyone must contribute a **FAIR amount of code AND documentation**. It means you cannot work only on the code or only on documentation. In other words, your team **MUST NOT** split the work by artefacts: a person working on a "class diagram", another on a diagram alone, and another working on design rationale alone. We will give a heavy penalty to your team as a whole.

You will meet up with your team ideally once a week for about 1 hour for team discussion and another 1 or 2 hours for integration and resolving conflicts (if any).

Peer Assessment Form

You should also complete the peer assessment form, which you can find here: https://forms.gle/mTK8WM6LZ345y8aQ7, to help us understand and evaluate your (and other team members') contributions to the assignment better.

About Late Submissions

Unless a team member has applied for and received special consideration according to the Monash Special Consideration Policy, late submissions will be penalised at 10% per day late (including weekends). Special considerations are granted individually.

It is all team members' responsibility to ensure that the correct versions of the documentation and code are present in the repository by the due date and time.

Marking Protocol & Rubric

Marking Protocol & Rubric

For this assignment, you have been asked to extend the functionality beyond those you designed and implemented in Assignments 1 and 2. You are free to change your design, of course. Here are some important instructions for completing the submission.

- You can update your design documents directly on GitLab.
- Make sure the implemented code is in the main branch on GitLab.
- Make sure you haven't modified the engine.
- Make sure you submit class diagrams for each of the new requirements.
- Make sure that the link to your contribution log is in the README file in your repository.

⚠ Before the final deadline, download a copy of your repository and upload it to Moodle (each team member should do this). If you forget to do this it is ok, don't panic, just upload it as soon as possible.

When using git:

- All team members **must** commit their work (not relying on one member to commit everything)
- Add commit comments that are effective in describing the changes (not just the GitLab defaults)

ℹ You should also complete the peer assessment form, which you can find here: https://forms.gle/mTK8WM6LZ345y8aQ7, to help us understand and evaluate your (and other team members') contributions to the assignment better.

Rubric (100 marks)

⚠ A **hurdle** for Assignment 3: design documents (**class diagrams for each requirement**) should be submitted to get marks for the implementation

Each requirement is worth 20 marks (REQ1-5: 20 marks each, and scaled depending on your group size). It can be broken down as follows:

Feature completeness (3 marks)

3 marks - Meeting all expectations as per the requirement. (e.g., all important classes and relationships are included in the diagram)

2 marks - Some classes or relationships are missing, but the requirement can still be implemented (e.g. Behaviour is not included in the UML class diagram even though it is a part of the requirement)

1 mark - Most of the classes or relationships are missing

0 marks - The UML class diagram or the implementation is missing

Implementation quality: design principles (8 marks)

8 marks - The implementation follows good design principles (if some principles are violated, the trade-off is explained in the rationale, e.g. using singleton to implement a feature), the design is easy to extend

7 marks - The implementation follows good design principles, and the design is easy to extend. However, trade-offs are not mentioned or explained in the rationale

6 marks - The implementation involves a minor violation of design principles (could be easily fixed), (e.g. some attributes are not set to private without any justification)

5 marks - The implementation involves a minor violation of design principles in multiple places (could still be easily fixed), (e.g. some attributes are not set to private without any justification)

4 marks - The implementation involves a non-severe violation of design principles that could be implemented in a better way (not a trade-off), (e.g. some code repetitions are found in the implementation)

3 marks - The implementation involves a non-severe violation of design principles that could be implemented in a better way in multiple places (not a trade-off), e.g. some code repetitions are found in the implementation (DRY)

2 marks - the implementation involves a severe violation of design principles that could be implemented in a better way, e.g. violating the SOLID principles

1 mark - The design/implementation can be considered hacky. For example, the implementation uses downcasting and instanceof in various cases. It can also be the case that abstraction is not used, making the design difficult to extend.

0 marks - The UML class diagram or the implementation is missing

Integration with the existing system (3 marks)

3 marks - The implementation effectively uses the engine classes (e.g. the submitted work demonstrates that the students understand the difference between actions and behaviours)

2 marks - The implementation does not use some engine classes as intended (e.g. behaviours are created for some actions that do not need it, such as actions performed by the player)

1 mark - The implementation includes custom classes for functionality that could be implemented with the engine class, e.g. created a custom ground class instead of using the ground class given in the engine package

0 marks - The engine has been modified in a minor or significant way OR the UML class diagram OR the implementation is missing

UML syntax and clarity (1 mark)

1 mark - if the diagram is perfect (no missing multiplicities, correct arrowhead for all relationships, correct relationship, e.g. realisation for classes implementing interfaces, generalisation for classes inheriting other classes, generalisation for interfaces inheriting other interfaces, etc.). The TA can make sense of the diagram.

0.5 marks - there are syntax errors in the diagram (no multiplicities for most associations, generalisation for classes implementing interfaces, realisation for classes extending other classes, classes extending multiple classes, etc.). Still, with a little work, the design can be understood by the TA.

0 marks - the diagram for the requirement is missing or does not resemble a UML diagram as required.

Alignment with design (4 marks) - Improving from previous A2 feedback incl. Design rationale and UML (class) diagrams

4 marks - The design rationale and the UML class diagram perfectly match the implementation AND The design rationale should contain a brief description of what has been done and why it has been done that way (NOT in terms of game design, e.g. I designed it that way to make the game easier to play). The explanation relates to principles taught in the unit, e.g. DRY, SOLID, etc. The rationale also includes some discussion of the pros and cons of the design, why the current design is chosen, and how it can be easily extended (e.g. my design achieves OCP because if a new character is added in the future, ...).

3 marks - The design rationale and the UML class diagram perfectly match the implementation OR the rationale contains a brief description of what has been done and why it has been done that way (NOT in terms of game design, e.g. I designed it that way to make the game easier to play). The explanation relates to principles taught in the unit, e.g. DRY, SOLID, etc. The rationale also includes some discussion of the pros and cons of the design, and why the current design is chosen, but it does not have any examples of how it can be extended in the future

2 marks - The design documents have some mismatches with the implementation OR the rationale contains a brief description of what has been done and why it has been done that way (NOT in terms of game design, e.g. I designed it that way to make the game easier to play). The explanation relates to principles taught in the unit, e.g. DRY, SOLID, etc. However, it does not have any discussion on the pros and cons of the current design & no examples of how it can be extended in the future

1.5 marks - The design documents have some mismatches with the implementation OR the rationale describes what has been done with some mentions of principles (e.g. my design follows SRP) with insufficient explanation

1 mark - Most of the design documents do not match with the implementation OR The rationale describes what has been done without in-depth explanation (class A has an association with class B because A will have an attribute of type B)

0.5 marks - Most of the design documents do not match with the implementation OR the rationale describes what has been done with justification in terms of game design or the rationale only mentions principles without any explanation

0 marks - The design rationale or the UML class diagram is missing for the requirement

Style & Javadoc (1 mark)

1 mark - The code is properly documented with Javadoc, including class-level and method-level documentation. The Google Java Style guide is followed properly (e.g. package names are written in lowercase, attributes and variables names are written in lowerCamelCase, class names are written in UpperCamelCase, etc.)

0.5 marks - Some classes and methods are missing Javadoc documentation OR some classes do not follow the Google Java style guide

0 marks - Most classes and methods are missing Javadoc documentation OR The Google Java style guide is not followed.

Individual Contribution & Handover Interview (100%)

This individual contribution is used to weigh the marks awarded at a group level (see above) for cases where individual students did not contribute to the group task fairly (see details under Individual Contribution, below) or did not show evidence that they understand their own work during the handover interview (see details under Handover Interview, below). If you contributed to the group task and demonstrate that you understand your own work you should get (100%) the full marks awarded at a group level as per the rubric above.

Individual Contribution (80%)

80% - The student did sufficient contributions to the team, i.e., multiple **commits** (+5) with helpful commit comments (note: default comments from the web UI don't count.), ideally, one commit per week previous to the submission deadline; and multiple **contribution log entries** (more than 3 entries) with meaningful explanations that can be understandable by a new team member (e.g. the TA). The student shows a reasonable quality of contributions, such as a fair understanding of the OOP knowledge in programming or designing UML diagrams; If applicable, the student contributes to BOTH design and code. There are positive reviews or zero complaints from other team members (via **peer assessment**);

50% - Peer assessment, contributions logs, and git logs show there are insufficient contributions from an individual; the student received a negative review from other team members which usually happened due to last-minute work, difficult communication outside of labs, or any other kind of

difficulties (evidence are collective from peer assessment, emails, contribution logs, and git logs). The student committed to git, with at least one commit and comment OR a single commit found without any comment or various commits that are not meaningful and that do not explain the purpose. Some contribution log entries (at most 2 entries) with meaningful explanations that can be understandable by a new team member (e.g. the TA).

0% - There is no evidence of contribution to the team from Git logs or contribution logs. Submitting files as attachments to the release notes would be marked as zero. If applicable, the student only writes comments (inline or Javadoc) without any appropriate programming contribution. Clear evidence of academic misconduct of an individual would be marked as zero.

Handover Interview (20%)

20% If the student can answer all questions during the handover interview satisfactorily. The responses demonstrate that the student understands the various parts of the assignment both parts created by the student and those created by other team members.

10% If all the questions but one is responded to adequately and sensibly. The responses still demonstrate some knowledge about the student's own work but may miss understanding about parts of the work created by other team members.

0% If two or more questions are not responded to adequately and sensibly. The remaining question(s) is/are partly responded to, but it is unclear whether the student understands their own work.

IMPORTANT: Failing to have meaningful commits (i.e. showing that the task was progressively completed) and/or failing the handover interview would automatically flag this as a potential case of plagiarism, it will be further investigated using a similarity check software, and **zero marks would be awarded.**

Contribution

We will assume all team members equally contributed to the assignment (i.e. 50-50% for a team of two or 33.33% each, for a team of three).

IMPORTANT: Any inquiry (e.g. potential conflict within a team) should be submitted via the emails below (not your AdminTA, Lecturer nor CE). Emails sent in other ways will not be processed in time.

FIT2099.Clayton-x@monash.edu if you are based in Clayton
FIT2099.Malaysia-x@monash.edu if you are based in Malaysia