# AO 2.2 Analysis:

## Computational Methods:

I want to make an offline top town zombie shooter. I will use a lot of abstraction in this project to remove details that are not needed, for example, the graphics will be 2D as since it is top down, the sprites do not need to be 3D as the camera is locked above so there will be no difference between a 2D sprites animation and a 3D one so there is no point using the extra processing power. The sprites will also be a lot less detailed than real life as since the camera will not zoom in, there is no point adding the unnecessary details that the user will not notice. The player will not be able to jump up as there will be nowhere to jump to, this will mean that the movement will not be as complicated and more instinctive to the user

I will use a lot of problem reduction in the project as there are already pathfinding algorithms that exist so there is no point me trying to create my own for the zombies. Additionally, code like taking damage will be rewritten a lot of times throughout the code so it will be better if the code is changed to a subroutine and called whenever it is needed.

I will use decomposition in the project, as the game can be split into smaller problems that are easily solvable. For example, the graphics can be split into background, foreground and HUD, the foreground can be split into enemies, the player, items, walls etc… and so on.

I will also use algorithms in my project, for example, the pathfinding of the enemies will have to be an algorithm so it can follow the same process every frame to know where to move every frame, the item system and inventory system will also have to use algorithms for things such as picking up items, dropping them and moving them around in the user’s inventory.

## Stakeholders:

The project was requested by siblings, Joel Green and Esther Green who are ages 14 and 12 respectfully. The problem they have is that they don’t have a game that they both enjoy as they are both competitive with each other but don’t both like the same game to have a competition between each other based on. Joel likes zombie / hoard shooter games like the Call of Duty zombies game modes whereas Esther likes roguelikes such as Enter the Gungeon. I asked if there were any other features that they wanted in the game and they said that they want it to be offline as they have issues with their internet that cause online games to run verry slowly, additionally they want it to have a high score and leader board system so they can be competitive with each other on the game. Esther said that the features that they want the most are chests that give random items and for it to be top-down and have smaller levels or floors rather than one big map. The features that Joel wants are for the game to have hordes of zombies and for them to drop currency that can be picked up and used to purchase items or unlock new areas of the map.

My wider target audience will consist of people that like to play 2d games and possibly can’t play online.

## Research:

Enter The Gungeon

Enter the Gungeon is a Roguelike Dungeon crawler made in 2016 by Dodge Roll. The features that I like and will be including in my project are:

The UI / HUD as it is simple and works well. I will use it in my project beacause it is simple and intuitive, so I won’t need to integrate a tutorial to explain how it works.

The simple, pixelated style. I will use a similar style in my project as it is a lot less detailed than photoreal, therefore it will be less intensive on the computer running it.

The random chest system. I will use this mechanic in my project as it is one of the features that my client, Esther, likes the most about games of this genre and it will make it different from the typical zombie shooter games with fixed weapons placed around the map

The different characters with different loadouts that you can pick from at the start of every run. I will include this in my project accept, I will make them closer to classes that you can customize and upgrade as this is one of the features that my client, Joel, wanted the most.

Image: <https://store.steampowered.com/app/311690/Enter_the_Gungeon/>

Call of Duty (“Zombies” game mode)

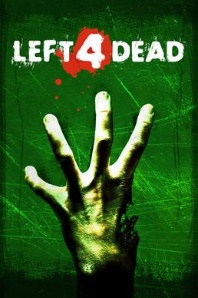
The Call of Duty series has a zombies or similar horde type game mode where you try to survive increasingly difficult waves until an objective has been achieved or the player is overwhelmed. The features in this game that I like the most are:

The increasingly difficult waves of zombies. I will use this feature beacause it is one of the features that my client, Joel, likes that most and it will make it easier to make harder levels beacause, instead of making many different types of zombies, I can just make a few and make the waves contain more and later wave ones have more health and do more damage.

The currency that you get by killing zombies. I will include this as it will be a good way to make the players progress at the proper rate as in the earlier waves, they will not be able to get enough currency to get weapons that they should not have or areas that they should not get to yet.

The fact that the missions have a goal that the player/s are trying to aim for, for example, survive until and beat a “final boss” or unlock and get to a final area or exit to escape. I will use this feature as it will make it possible to make a campaign out of smaller levels and maps in which the objectives get harder to achieve and this is a feature that my client, Esther, said that she wanted.

Image: <https://www.callofduty.com/blog/page?id=Call-of-Duty-Black-Ops-Cold-War-Zombies-A-New-Beginning&src=agb>

Left 4 Dead

Left 4 dead is a first-person zombie shooter, similar to the call of duty “zombies” mode. However, instead of there being endless escalating waves of zombies, the zombies are already positioned on the map, waiting for the players. There are also events and enemies that can “alert the hoard” which will send a massive wave of zombies at the player. The features that I like the most are:

The different “special infected” that are rarer than normal enemies but are much harder to deal with. I will use this feature as the “special infected” make encounters much more difficult and more interesting than just more and more infected running at the player being gunned down.

The small “campaigns” that are made up of small “acts”. This is one of the features that my client, Esther, wanted and the overarching campaign will give more meaning to the individual missions which will make them more interesting to play through them for the player as their choices in earlier levels can affect the final outcome.

The actions and enemies that can “alert the hoard” and send a massive wave of zombies at the player/s. I will include this as it will force that player to be a lot more careful and not just brute force through the whole campaign all guns blazing as doing so can send extra hordes of infected after them during waves which can make situations much harder to deal with.

Image: <https://en.wikipedia.org/wiki/Left_4_Dead>

## Minimum requirements:

* The game has a main menu with a start, high scores and quit buttons
* The start button starts the first level so that the player has a way to start the game
* The high scores button takes the user to a screen showing the top 10 scores achieved and who achieved them so that the player can see what their high score is or who has the high score
* The quit button closes the game so that the player can close the game easily
* The game itself will be played over at least 3 levels so that the game isn’t just the same level and has some variety
* Each level will be one of a random selection of pre-set environments so that the campaigns are not always the same
* At the end of the last level the user will be shown their score and where they are on the scoreboard so that the player knows if they have beaten their best or not
* The player will be able to move with the ‘w’, ‘a’, ‘s’ and ‘d’ keys so that the player can interact with the game
* The player will be able to interact with doors and chests to pay to open them and items on the floor to pick them up using the ‘e’ key so that the player can interact with the environment
* There must be a HUD that shows the user things like their health and currency so the player knows if they are about to die or if they can afford a door or chest
* The player will be able to aim their weapon using the mouse and shoot their weapon using left click so that the player can attempt to defeat the enemies
* The weapons the player uses will fire projectiles that will either stop if the hit a wall or, if it hits an enemy, it will stop and deal damage to them
* During the game the player, environment and the enemies will be correctly drawn to the screen so that the player can actually see what’s going on
* The enemies will come in waves of increasing size spawned from set locations in the environment and the next wave will start set time after the last started so that the player has enemies to try and stop them and an increasing difficulty to add time pressure

## Limitations:

There are some limitations to the solution to this problem. Firstly, I have never coded a project like this which will mean that many of the mechanics that I want to implement will take much longer beacause I won’t immediately know how to code it.

I only have a year to develop the project which will limit the number of aspects that I can add to the game which will mean that some mechanics that my clients want might get cut from the final product to get the project done in time.

I am not going to make it cross-compatible with other devices which means that the target audience is limited to people that have computers.

I have never done pixel art which will mean that it will take a very long time to make the assts needed for the game.

The budget for the game in terms of money and time are a lot lower than normal games from big companies such as Valve which means that the quality of the product will be lower than, for example, triple A games.

I am the only programmer on this project which will dramatically decrease the amount of code I can get done in the time. That and the limited time mean that the project will be a lot smaller than normal games on the market today.

I am using Pygame for this project which is open source which means that it isn’t often updated so the library can be left behind the code itself which can create errors in the code that didn’t exist before.

## Hardware and software Requirements:

The game will be fairly simple which will mean that there will not be a lot of limitations to the hardware that is needed, the only issue is that it will not be cross-play so the people that don’t have a computer will not be able to play the game beacause it will not be supported on their device. They will need, at a minimum, an intel core I3 GT 2.00 GHz or an equivalent CPU, 1GB of RAM and 130 MB of storage space

In terms of software, they will need windows 7, 8 or 10 or equivalent for other operating systems

## Success criteria

### Menu:

* Start button takes the player to a menu that allows the player to customise their character’s starting loadout so that the player can customize their character and make it feel more like their build rather than a pre-set one
* In this menu the player can also pick one of a set of pre-set character appearances to add more customization options to the player
* The player can name their character and this name will be the name that is shown on the scoreboard so that the player can see who scored what scores on the leader board
* The quit button makes a popup that asks the player if they are sure they want to close the game and has a ‘yes’ and a ‘no’ button. If the player picks ‘no’, the popup will close. If the player picks ‘yes’, the game will close so that the player cannot miss click and accidentily close the game
* The game will have a pause menu that the player will be able to open with the ‘Esc’ button. While the menu is open, the game pauses. This is so that the player has a way to pause the game mid-run if they need to
* The pause menu will have a resume button on it that, if pressed, will close the menu and resume the game
* The pause menu will have a quit button that the player can use to close the game mid-run. This is so that the player has an easy way to close the game while they are playing it

### Game:

* The game will consist of a set of levels ending with a boss level so that the player gets a sense of progression as they complete levels
* The levels will be a random one of a set of environments to add variety to the levels
* The objective will also be random and not based on what the environment is to add more variety between the levels and runs
* The environments and objectives cannot repeat in the same run so that the runs do not get repetitive
* The environments will, in random places, contain things to interact with such as chests or items/locations linked to the current objective so that the player has things to interact with
* The environment will be split up into multiple sections divided by doors that the player can open by using currency that they have gained through the level so that the player has a clear way to progress
* The chests will contain an item, such as ammo, or a weapon that the player can pick up so that the player, if they are getting bored of a certain weapon or are running out of ammo, can pick up another one
* The player, like with the doors, will have to spend currency to open the chests so that the player will have to “earn” the item by defeating enemies
* The cost of the chests and doors will be based on the level that the player is on and how far it is away from the player’s spawn so that as the waves of enemies get bigger, the doors and chests don’t get easier to open
* The last level will be a boss level where the player will have to beat one massive enemy that, when defeated, opens the final exit so that the game has a big final challenge that the player has to beat to finish the level
* Certain objects or enemies can “alert the hoard” and send an additional group of enemies at the player so that the player has to think about what they are doing and if it is worth risking more enemies

### Player:

* The player will be able to move the player in 8 directions using the ‘w’, ‘a’, ‘s’ and ‘d’ keys so that the player can move around the environment
* The player will be able to interact with various objects around the map using the ‘e’ key so that the player can pick up objects and open the doors and chests around the environment
* The player will be able to aim their weapon using the mouse and fire it using left click. This I so that the player can use the items they have picked up and attempt to kill the enemies
* The character in the game will always look towards the mouse. This is so that the player has an indication of where they are aiming when they are not concentrating on where the mouse is
* The player will have a total health which will be presented as a health bar and if the health goes below 0, the player will die, and the game will end. This is so that there is a way for the player to lose

### The GUI:

* During the game the player’s health and currency will be displayed in the top left corner of the screen. This is so that the player has a clear indication of their health and currency while not being intrusive on the game
* During the game the player’s current weapon, total ammo left, and ammo left in the weapon will be displayed in the bottom right corner of the screen. This is so that the player has a clear indication of what weapon they are using, how much ammo they have left and when they will have to reload while not distracting from the game itself

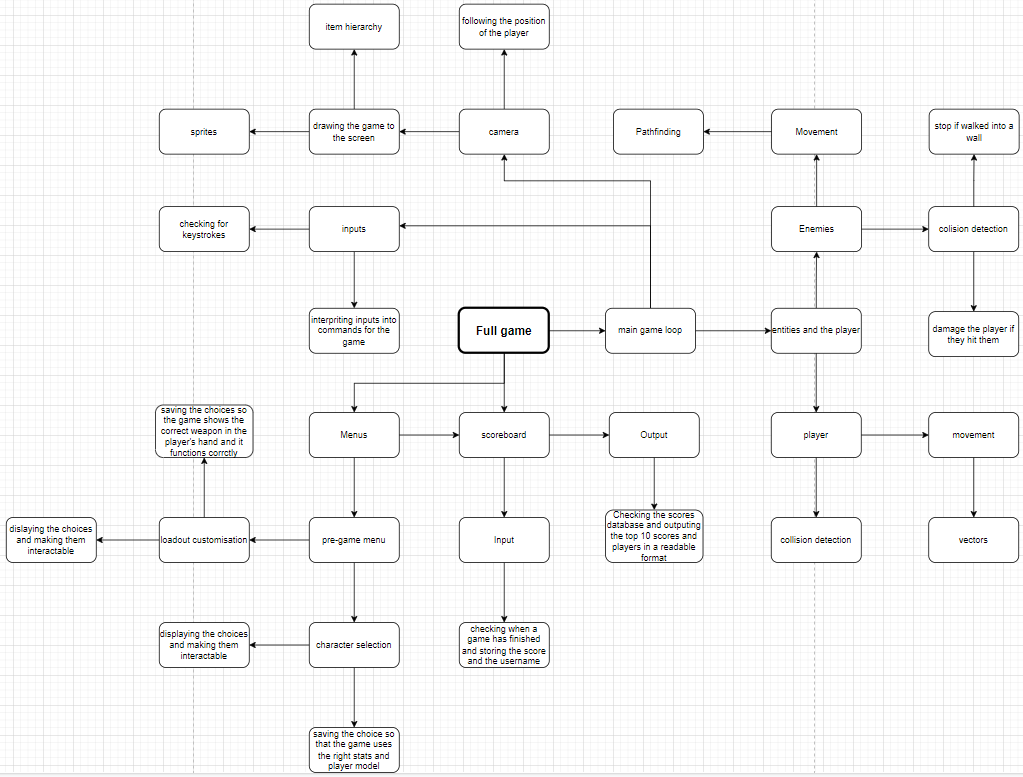
### The enemies:

* The enemies will be spawned in waves from set locations on the map that gradually increase in size as waves progress. This is so that the player has a sense of progress as the waves get harder
* The enemies will have random attributes within certain limits. This is so that the enemies aren’t all exactly the same which could get boring to the player
* The enemies will get gradually more and more health as the waves progress, this is so that the waves get more difficult in more ways than one
* Certain enemies will be elite versions that will be rarer but more dangerous to the player. This is, again, so that all of the enemies aren’t the same and get boring for the player
* There will be different types of elite enemies which will be different from each other. This is to add more variety to the game
* The enemies will have a basic pathfinding algorithm and always walk towards the player in an attempt to damage them. This is so that the enemies move and are actually a threat to the player
* Whenever an enemy is killed, the player will receive an amount of currency that will be determined by if the enemy was an elite or not, what type of elite it was and what wave it is. This is so that there is a way for the player to obtain currency and so that there is a need to kill the enemies beyond survival

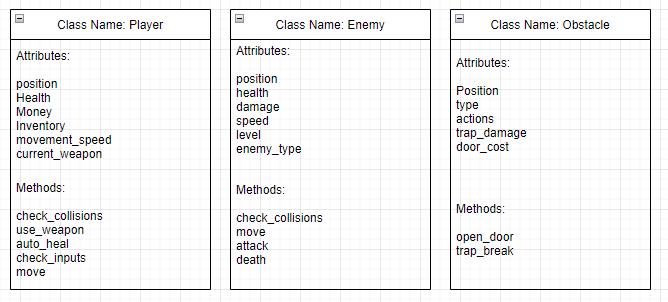
# AO 3.1 Design:

## Structure of the solution:

### Decomposing the problem:



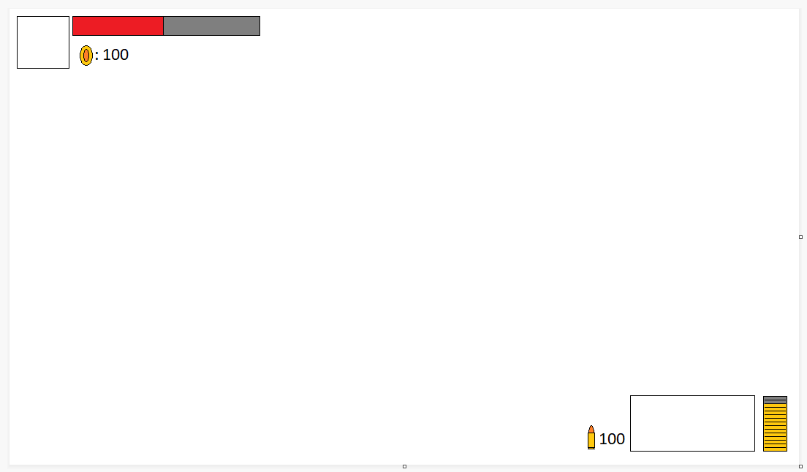
### Class diagrams:



### Visualization diagram:

There will be a picture of the player here

The player’s health will be displayed here



The player’s currency will be displayed here

The amount of ammo left in the player’s clip will be displayed here

There will be a picture of the player’s current weapon here

The amount of ammo that the player has left in their inventory will be displayed here

The UI will look like this beacause it is simple and easy to understand without being intrusive and restricting the player’s view of the game

## Algorithms:

Player movement:

Inputs = GetInputs()

X = 0

Y = 0

For key in inputs:

If key == “w”:

X += 1

Endif

If key == “s”:

X -= 1

Endif

If key == “d”:

Y += 1

Endif

If key == “a”:

Y -= 1

Endif

Endfor

If not Player.DetectCollision(x, y):  
 player.x += x

Player.y += y

Endif

This algorithm will handle the controls for the movement of the player. This will allow the user to use the “wasd” keys to move the charter around the map

## Usability features:

There will be a button to close the game mid game so that the player can quit the game easily without being trapped in the level

There will be a currency count underneath their health bar so that the player has a good indication if they can afford to open a door or chest yet

The player will have an ammo count as well as how much ammo they have left in the clip, this is so that they know how close they are to having to reload or running out of ammo

The player will have an image of the weapon they are using in the corner next to their ammo count so that they know which weapon they are using without having to fire their weapon to find out what it is.

The sprites for the elites and the default enemies will be clearly different so that the player can clearly see where the elites are.

Items on the ground will be highlighted so it is more likely that the player will see them and not miss them

There will be a high score system and a leader board so that the player can see what their highest scores were an if multiple people play on the same game, they can be competitive with each other.

The player will be able to pause the game so that they can quit midway through a run or pause it temporarily with ease.

The players will be able to change the button controls to what they want so that if the player doesn’t like the default setup, they will be able to change it to something they are used to or find more intuitive.

## Key Variables and Data structures.

### Constants:

There will be a lot of constants in the code as there will be a lot of variable features that will stay the same and be used in multiple places. Some of the constants will include:

|  |  |  |
| --- | --- | --- |
| Constant name: | Constant type: | Use(s): |
| PLAYER\_SPEED | Integer | Used in the movement function to handle how fast the player can move |
| PLAYER\_HITBOX | Pygame – rect | A rectangle object that will be used when checking collisions |
| PLAYER\_MAX\_HEALTH | Integer | The maximum amount of health the player can have. Healing items and effects cannot make the player’s health go over this |
| ENEMY\_BASE\_HEALTHS | List | A list of the health values for each type of enemy when they are level 1 |
| ENEMY\_BASE\_DAMAGES | List | A list of the damage values for each type of enemy when they are level 1 |
| ENEMY\_BASE\_SPEEDS | List | A list of the speed values for each type of enemy when they are level 1 |
| ENEMY\_LEVEL\_MULTIPLIERS | List | A list of the percentage that the enemies’ stats increase by each level |
| ENEMY\_TYPES | List | A list of the different types that an enemy can be |

### Variables:

|  |  |  |
| --- | --- | --- |
| Variable name: | Variable type: | Use(s): |
| level | Integer | Keeps track of what wave the player is on and how much times the enemies’ modifiers are to be applied |
| player\_group | Pygame – sprite group | Holds the player sprite object and keeps track of its place in the draw hierarchy |
| enemy\_group | Pygame – sprite group | Holds the enemies’ sprite objects and keeps track of its place in the draw hierarchy |

### Functions:

|  |  |
| --- | --- |
| Function name: | Use(s): |
| detect\_collisions(object\_a) | Used to check when a given object have collided |
| draw\_health\_bar(health, max\_health) | Will use the player’s current health and their max health to draw a health bar on the screen |
| draw\_hud() | Will use the player’s attributes and the draw\_health\_bar function to draw the player’s hud |

### Classes:

Player – attributes:

|  |  |  |
| --- | --- | --- |
| Attribute name: | Attribute type: | Use(s): |
| health | Float | Holds the player’s current health |
| currency | Integer | Holds the amount of currency that the player is holding |
| xcord | Float | Holds the position that the player is in the x-axis |
| ycord | Float | Holds the position that the player is in the y-axis |
| weapons | List | It will be a list of Weapon objects that the player is holding |
| ammo | List | Holds the amount of ammo the player holds o each ammo type |

Player – methods:

|  |  |
| --- | --- |
| Method name: | Use(s): |
| movement | used to handle the player inputs and move the character in response to these |
| take\_damage | Used when anything damages the player and reduces the player’s health attribute |
| check\_for\_death | Checks if the player’s health total has dropped below 0 and if it has, it will stop the game |
| heal | Increases the player’s health by a given value |
| check\_over\_heal | Used in the heal method to check if the health will go above the player’s maximum health value and will prevent it doing so |
| shoot | Handles the controls for the player’s weapons and will call methods in the weapons themselves to fire them |

Enemy – Attributes

|  |  |  |
| --- | --- | --- |
| Attribute name: | Attribute type: | Use(s): |
| health | Integer | Holds the enemy’s health value |
| xcord | Float | Holds the position that the enemy is in the x-axis |
| ycord | Float | Holds the position that the enemy is in the y-axis |
| damage | Integer | This will hold the enemy’s average damage and when they hit the player, they will do a random amount of damage between 5% + and 5% - this value |
| speed | Integer | This will be how fast the enemy can move |
| type | Integer | The index value of the type the enemy is |
| cooldown | Integer | If the enemy is an elite and has a special ability that they can activate, it will have a cooldown so they cannot permanently use it |
| drop\_money | Integer | This will be the amount of money that the enemy will drop and give the player when they die |

Enemy – Methods:

|  |  |
| --- | --- |
| Method name: | Use(s): |
| move | This will move the enemy towards the player |
| attack | this will make the enemy try to hit the player if they are within range |
| use\_special | checks if the time between the last use of the special is above the cooldown and if it is they use their special |
| check\_death | Checks if the enemy’s health value is below 0 if it is the enemy dies and the player receives the currency for killing them. |
| take\_damage | If the enemy is hit by the player, this method is called, and it will decrease the enemy’s health value |
| check\_over\_heal | If the enemy is healed, it makes sure that their health total does not go above their maximum health value |

Item – Attributes:

|  |  |  |
| --- | --- | --- |
| Attribute name: | Attribute type: | Use(s): |
| xcord | Float | Holds the position that the item is in the x-axis |
| ycord | Float | Holds the position that the item is in the y-axis |
| type | String | Contains what type of item it is and will be used in the methods to govern how the item acts |
| name | String | Contains the name of the item and will be displayed to the player |

Item – Methods:

|  |  |
| --- | --- |
| Method name: | Use(s): |
| Interact | Used when the player interacts with the item and the code will use polymorphism to interact with different objects in different ways |
| Use | Used when the player uses the item, for example, fires a gun. Polymorphism will be used in the same way as the interact method |

## Iterative test data:

For the first prototype of the program, I will implement and test the following:

* During the game the player, environment and the enemies will be correctly drawn to the screen so that the player can actually see what’s going on
* The player will be able to move the player in 8 directions using the ‘w’, ‘a’, ‘s’ and ‘d’ keys so that the player can move around the environment
* The character in the game will always look towards the mouse. This is so that the player has an indication of where they are aiming when they are not concentrating on where the mouse is
* The player will have a total health which will be presented as a health bar and if the health goes below 0, the player will die, and the game will end. This is so that there is a way for the player to lose
* The enemies will have random attributes within certain limits. This is so that the enemies aren’t all exactly the same which could get boring to the player
* The enemies will have a basic pathfinding algorithm and always walk towards the player in an attempt to damage them. This is so that the enemies move and are actually a threat to the player
* Whenever an enemy is killed, the player will receive an amount of currency that will be determined by if the enemy was an elite or not, what type of elite it was and what wave it is. This is so that there is a way for the player to obtain currency and so that there is a need to kill the enemies beyond survival
* During the game the player’s health and currency will be displayed in the top left corner of the screen. This is so that the player has a clear indication of their health and currency while not being intrusive on the game

For the second prototype, I will implement and test the following:

* The game has a main menu with a start, high scores and quit buttons
* The start button starts the first level so that the player has a way to start the game
* The high scores button takes the user to a screen showing the top 10 scores achieved and who achieved them so that the player can see what their high score is or who has the high score
* The quit button makes a popup that asks the player if they are sure they want to close the game and has a ‘yes’ and a ‘no’ button. If the player picks ‘no’, the popup will close. If the player picks ‘yes’, the game will close so that the player cannot miss click and accidentily close the game
* The game itself will be played over at least 3 levels so that the game isn’t just the same level and has some variety
* Each level will be one of a random selection of pre-set environments so that the campaigns are not always the same
* At the end of the last level the user will be shown their score and where they are on the scoreboard so that the player knows if they have beaten their best or not
* The player will be able to interact with doors and chests to pay to open them and items on the floor to pick them up using the ‘e’ key so that the player can interact with the environment
* The enemies will come in waves of increasing size spawned from set locations in the environment and the next wave will start set time after the last started so that the player has enemies to try and stop them and an increasing difficulty to add time pressure
* The environments will, in random places, contain things to interact with such as chests or items/locations linked to the current objective so that the player has things to interact with
* The environment will be split up into multiple sections divided by doors that the player can open by using currency that they have gained through the level so that the player has a clear way to progress
* The chests will contain an item, such as ammo, or a weapon that the player can pick up so that the player, if they are getting bored of a certain weapon or are running out of ammo, can pick up another one
* The player, like with the doors, will have to spend currency to open the chests so that the player will have to “earn” the item by defeating enemies
* The cost of the chests and doors will be based on the level that the player is on and how far it is away from the player’s spawn so that as the waves of enemies get bigger, the doors and chests don’t get easier to open

For the third prototype, I will implement and test the following:

* Start button takes the player to a menu that allows the player to customise their character’s starting loadout so that the player can customize their character and make it feel more like their build rather than a pre-set one
* In this menu the player can also pick one of a set of pre-set character appearances to add more customization options to the player
* The game will have a pause menu that the player will be able to open with the ‘Esc’ button. While the menu is open, the game pauses. This is so that the player has a way to pause the game mid-run if they need to
* The pause menu will have a resume button on it that, if pressed, will close the menu and resume the game
* The pause menu will have a quit button that the player can use to close the game mid-run. This is so that the player has an easy way to close the game while they are playing it
* The objective will also be random and not based on what the environment is to add more variety between the levels and runs
* The environments and objectives cannot repeat in the same run so that the runs do not get repetitive
* The last level will be a boss level where the player will have to beat one massive enemy that, when defeated, opens the final exit so that the game has a big final challenge that the player has to beat to finish the level
* Certain objects or enemies can “alert the hoard” and send an additional group of enemies at the player so that the player has to think about what they are doing and if it is worth risking more enemies
* During the game the player’s current weapon, total ammo left, and ammo left in the weapon will be displayed in the bottom right corner of the screen. This is so that the player has a clear indication of what weapon they are using, how much ammo they have left and when they will have to reload while not distracting from the game itself
* The enemies will be spawned in waves from set locations on the map that gradually increase in size as waves progress. This is so that the player has a sense of progress as the waves get harder
* The enemies will get gradually more and more health as the waves progress, this is so that the waves get more difficult in more ways than one
* Certain enemies will be elite versions that will be rarer but more dangerous to the player. This is, again, so that all of the enemies aren’t the same and get boring for the player
* There will be different types of elite enemies which will be different from each other. This is to add more variety to the game

## Post iterative test data:

There are certain things that I can only test after the code has been fully finished. After the iterative development phase I will test the following things:

* The game will consist of a set of levels ending with a boss level so that the player gets a sense of progression as they complete levels
* The player can name their character and this name will be the name that is shown on the scoreboard so that the player can see who scored what scores on the leader board
* Once you complete the game, win or lose, the player will be taken to the leader board screen

# AO 3.2 Developing the coded solution:

### Making a window: Text Description automatically generated

The first thing I made was the game class which will be what most other classes will be associated with as it is where the main game loop (events, updates, draw) will be handled.

Then I made the window an attribute of the game class so that when the game object is made, the window is opened.

In the draw method I made the window constantly be filled with black.

Finally, I made the events method check if the x in the corner of the window has bin clicked and will close the window and end the program if it has.

Text

Description automatically generated

Calendar

Description automatically generated

Error:  
when trying to make the player move, they started moving on a key press, but they did not stop when the key was released

Text

Description automatically generated

This was beacause, pressing the key changed the player’s dx or dy attributes but they never changed back so the player would keep moving, I fixed this by changing the player’s update method to reset the player’s dx and dy attributes to 0 every game cycle

Graphical user interface, text

Description automatically generated

# AO 3.3 Evaluation: