Project Analysis:  
Dead by Daylight Calculator

A screenshot of a game

Description automatically generated with low confidence

By Pepijn Muskens

# Project summary

## General:

In the game Dead by Daylight 4 people playing as survivors go up against 1 person playing as the killer. To get an advantage in a match survivors can bring items into the game which can help them in different ways. There are 5 different types of items and each item can have up to 2 modifiers called addons, there are around 60 addons in total in the game. These addons can make the effects of the items even stronger. For this project I want to make a sort of calculator where you can combine items with addons to see how strong their effect will be in a game.

## In depth:

### Goal of a game:

The goal of the survivors is to escape the game before the killer can kill them all. In the game there are 7 broken generators, when 5 of the generators are repaired the 2 exit gates will be powered. And the survivors can leave the game. Repairing the generators and opening the gates takes time so the killer will chase survivors away from these objectives. If a killer catches up with a survivor and hits them they are injured and get a speed boost to get away again. Survivors can be healed if they can find each other in the game. When a injured survivor gets hit by the killer they will fall on the ground and the killer can pick them up and hang them on a hook. During the time a survivor is on the hook they are unable to do anything and have to wait for another survivor to come rescue them. When a survivor is hooked a third time they will die and the rest of the survivors can keep trying to escape.

### Getting items:

In the menu screen survivors can spend point they earn during matches on items, addons and perks. Items have different rarities generally based on how good they. The more rare a item is the more it will cost. Players can a random selection of items which resets every time they have bought around 70% of that selection.

### Perks:

Survivors and killers can bring up to 4 perks into a game. These perks can have a variety of effects like temporarily increasing your movement speed to being able to see the killer in certain situations. Some of these perks can have effect on items by increasing the time they can be used or increasing their speed. These perks can be unlocked by playing the game so not all players have access to all of the perks.

### Item Types:

#### Flashlight:

A Flashlight is used to blind the killer and to counter some killer’s powers. Addons can modify the duration the flashlight can be used, the width and range of the beam and the duration the killer is blinded.

#### Key:

The key has multiple uses. When a player is the last survivor standing a key allows the user to open a hatch to escape the game. Addons can give a key extra uses like seeing the killer through walls and seeing where the other survivors are. The second addon can be used to increase the range where you can see trough walls and the duration the key can be used.

#### Map:

There are 2 different maps in the game. When using a map, you can see objectives through walls where you have been within 8 meters of. The first map can only track generators while the second map can see all objectives like exit gates, chests, and the hatch when they are the last survivor left. The addons can increase the range in which the map marks objectives and the time the map can be used.

#### Med-Kit:

Med-Kits enable the user to heal themselves. The addons increase the number of times you can use the Item and how long it takes to heal yourself.

#### Toolbox:

A toolbox can be used to speed up the repair of a generator and to sabotage hooks, when a hook is sabotaged, the killer can’t use it for some amount of time. The addons can increase the time the toolbox can be used, the speed at which they sabotage hooks or repair generators and the time it takes for a hook to be usable after it has been sabotaged.

# Requirements

* Must have:
* A user can select an item and 2 addons and see important statistics on how the combination would perform.
* For each item type there is a way to get the best preforming combination in different categories.
* Should have:
* Users can select which items they want to use when calculating the strongest combinations.
* Users can select the total cost of the item and addons when calculating the strongest combinations.
* Could have:
* Users can select perks to modify the result of the calculator.
* The calculator has an option to select how many perks to use when looking for the best combination.
* A user can select which perks they have available for the calculator to find the best combination for them.
* Would have:
* The application can have a second calculator to see how addons effect killer powers.

# Screens

A screenshot of a video game

Description automatically generated with medium confidenceI want to base the layout of my screens on the menus of the game where the players can select their items and addons so it will be intuitive to use. In this image you can see the UI in game. When clicking on the item slot at the top left of the screen. The inventory space at the bottom will show all the items you have available. If you select an item, it will automatically switch to the first addon slot and show all the addons you have available which can be used with the selected item.

In my application this selection will fill the right side of the screen while the left displays a list of statistics and gives you the options to let the website calculate the best combinations for you.

# Architecture

## C4 diagram

### Full diagram

This model describes how a user will communicate with the application and how the application communicates within itself. The user can only see the frontend of the application and interacts with the whole application from there.

Diagram, schematic

Description automatically generated

### Level 2

Diagram

Description automatically generated

The application consists out of 3 separate containers 1 for the frontend 1 for the backend, and 1 for the database. The frontend is responsible for showing the statistics and give the user an easy way to select the combinations they want. It is the backends task to serve that data to the frontend with the information it gets from the database and make the calculations for the best combinations. The database will store all the data and can link users to the items and addons they want to personalize the experience.

Diagram

Description automatically generated

### Level 3

This level contains the full backend architecture. The frontend communicates with the controller in the backend. In the logic layer all the data it gets from the data layer is formatted and used to calculate whatever the controller requests.   
  
The Interface layer ensures the abstraction of the data layer, and the factory layer makes sure both logic and data are dependent on an abstract layer.

### Level 4

Diagram

Description automatically generated

The fourth level zooms in on the code itself. The main functions happen in the calculator which has an item- and an addon container who save everything available to a user. It can put these in a loadout format to calculate the best combinations and send these out of the application.

# Solid

The architecture of the backend is based on the solid principles. This means the high-level modules and the low-level modules are dependent on abstract layers. Each layer should only be responsible for a single task.