UNREAL SURVIVAL 2.0

GAME DESIGN DOCUMENT

Table of content:

Back to the table of contents

Back to the table of contents

0. Readme

This document is written using hyperlinks and special mannerisms; these are used to more clearly showcase and/or explain how the game is to feel and function.

In this introductory *readme* section I wanted to explain shortly how this design document is structured:

In the table of contents and during some sections you'll find some *hyperlinks* that look like this: *example*.
These hyperlinks are typically suited for *pdf-files,* but they can also be used in the Google Documents-editor. To use a hyperlink, simply click the **blue hyperlink**, and then on the **link** that appears in the *pop-up*.

When explaining mechanics and functions, you'll probably come across this type of text: **[variable_example]**.

The example above is a *variable that is to be tweakable by the designers.* In most cases this will be some kind of *value* directly linked to the mechanic that currently is being detailed.

An example of this scenario would be:

> *...and the turn ratio is to be **[turn_speed]**. When the turn ratio...*

In this example, **[turn_speed]** is an arbitrary number that has to be exposed or in some way tweakable by the designers, or any other group member.

Back to the table of contents

1. Overview

1.1. High Concept

Unreal Survival is a top down horror game for one player. The premise of this game is to escape from a decrepit city that has been overrun by horrific creatures that hunt humans.

The player, only equipped with a flashlight and a handful of items will have to sneak past the creatures, and use his/her environments to their benefit in order to escape the desolate city.

1.2. Gameplay

The goal of this game is to get to an end point in the other end of the city. Along the way the player will have to bypass a number of creatures that lurks in the dark. These creatures are hard to kill, very fast and highly lethal. They do not seem too keen on *li**ght** **so**urces **though**.*

The **play**er w**ill** have **to** use his/her flashlight to ***fend off these creatures***. There are also other items in the city that the player can store in their *inventory*: some items include *flares, walkie-talkies* or *med-kits.*

1.3. **Winning/Losing Condition**

The player **wins** the game by reaching the final destination in one piece. There is no other means of winning this game.

Losing state will be reached when the player has depleted all of their health. When all of the player's health has been depleted they are to be spawned at a recent *spawn point.*

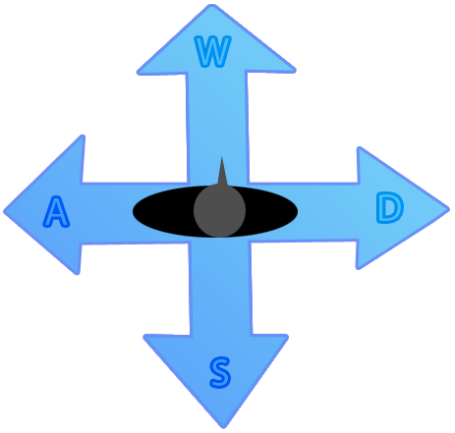
Back to the table of contents

1.4. Controls

Controls yooo.

2. Player mechanics

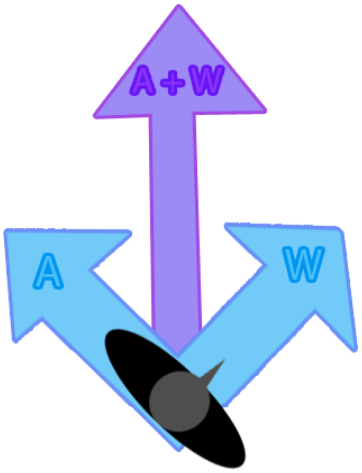
2.1. Movement



The player moves the avatar with the **W, A, S** and **D** keys; each key represents a specific direction in the world's space. The directions do not correlate to the player's *avatar*.

When the player presses the **W**-key the player's avatar will **rotate** and **move** *up* in relation to the worldspace. This direction is solely based on the world and not on either the *camera* or *the characters transform*.

This rule applies to the other keys as well; if the player presses the **A**-key the avatar will move straight down in relation to the the *world.* The same goes for the **S**- and **D**-key, which will move the player to the right and to the left respectively.



The player can in total move in **eight directions** by pressing down several keys at once.

If the player holds down *several keys at once*, the direction will be decided based on these rules:

- If the two keys can be combined into an *eight direction,* the player will walk in that direction (see the exemplary image to the right)
- If the player presses two keys in the *opposite direction* eg. **W** and **S**, the key that was pushed down *last* will have priority and the avatar will

Back to the table of contents

move in that direction.

For example, if the player presses **W** first, and then the **S** key, the player will move in the direction of the **S** key.

Back to the table of contents

2.2. Camera Movement

The camera is turned toward the *player* at a fixed distance. When the player's avatar moves around the gamespace, the camera will *translate in accordance*: as if it is a ***child*** *to the player*.

Back to the table of contents

2.3. Aiming and using items

Aiming stuff goes here yo.

Back to the table of contents

2.4. Flashlight

2.4.1. Flashlight overview

The flashlight will continuously *illuminate* a concentrated area in front of the player's avatar. This patch of light will have varying effects on the creatures or objects that it hits, besides lighting them up.

2.4.2. Turning the flashlight on/off

The player can turn off the flashlight by pressing the designated *flashlight-button.* The flashlight will be turned off once the **[flashlight_key]** has been *released*.

Similarly, when the flashlight is turned off the player can turn it on again by simply pressing and releasing the **[flashlight_key]**.

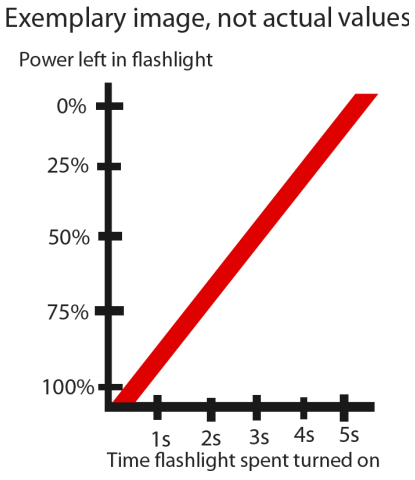
2.4.3. Power-meter of the flashlight

The flashlight runs on *power:* this power is slowly depleted as the player is having the flashlight turned on.

The flashlight power is to be depleted at a **static rate**: the rate at which the flashlight depletes is decided by a **[flashlight_deplete]** value.

As stated, the flashlight power will only be depleted as long as it is *turned on.*
Once the flashlight power has been fully depleted it will turn off *automatically,* and start recharging.

The player cannot start the flashlight until it has recharged *completely.*

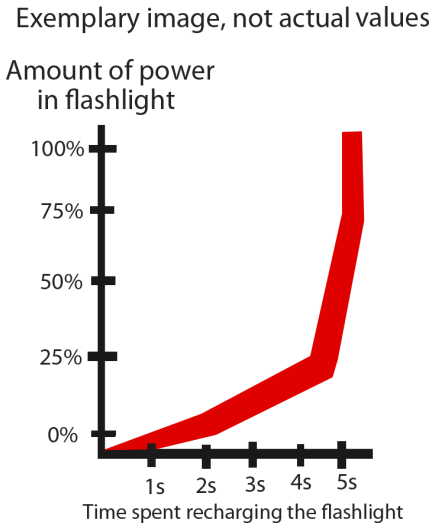


Back to the table of contents

The maximum amount of power in the flashlight is to be decided by the **[flashlight_max_power]** value. This value decides for how long the player can have their *flashlight turned on*.

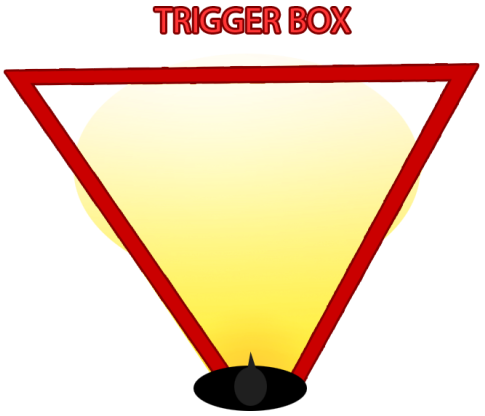
When the flashlight is turned off the *power-metre* will recharge at an **exponential rate**: the longer the player keeps the flashlight turned off, the faster the power will refill.

The rate at which the flashlight will recharge is to be decided by the **[flashlight_recharge]** value.
The potency of the exponential recharge should be tweakable by the designers through a **[flashlight_recharge_rate]** variable.



Once the power has recharged to its fullest (aka **[flashlight_max_power]** value, having the flashlight turned off will yield *no effect*.

2.4.4. Implementation



Rather than using the light itself of the flashlight, mechanically, a *trigger box* can be rendered in the illuminated area. When a creature collides/enters this trigger box, a function can take place: like for example fending off the creatures.

The **shape** of the trigger box is to resemble the flashlights illuminated patch. If the flashlights reach is extended by *Aiming* the trigger box is to extend alongside the extended illuminated area, to encompass a larger area of effect.

Back to the table of contents

2.5. Revolver

2.5.1. Revolver overview

The *revolver* is the player's only tool for *eliminating* enemies. The revolver is obtained immediately from the start of the game, and is usable throughout the game; the only thing refraining the player from using the revolver is if they have any spare *bullets.*

2.5.2. Ammunition overview & management

The revolver uses a resource in the means of *bullets* when used; firing **once** will consume **one bullet**.

Bullets are found in the world as *objects to be picked up with the space bar;* the number of bullets picked per location is not defined, but rather arbitrary, and to be decided by *designers.*

The revolver itself can contain a **maximum of six bullets**; it is from this source bullets are consumed when *firing the gun.* In order to fill this container, the player has to *reload their gun,* using a designated key (namely the **R-key**). When reloading, if there are *any bullets left in the revolver,* they won't be removed; the revolver will be filled with bullets until it has hit its maximum amount of bullets.

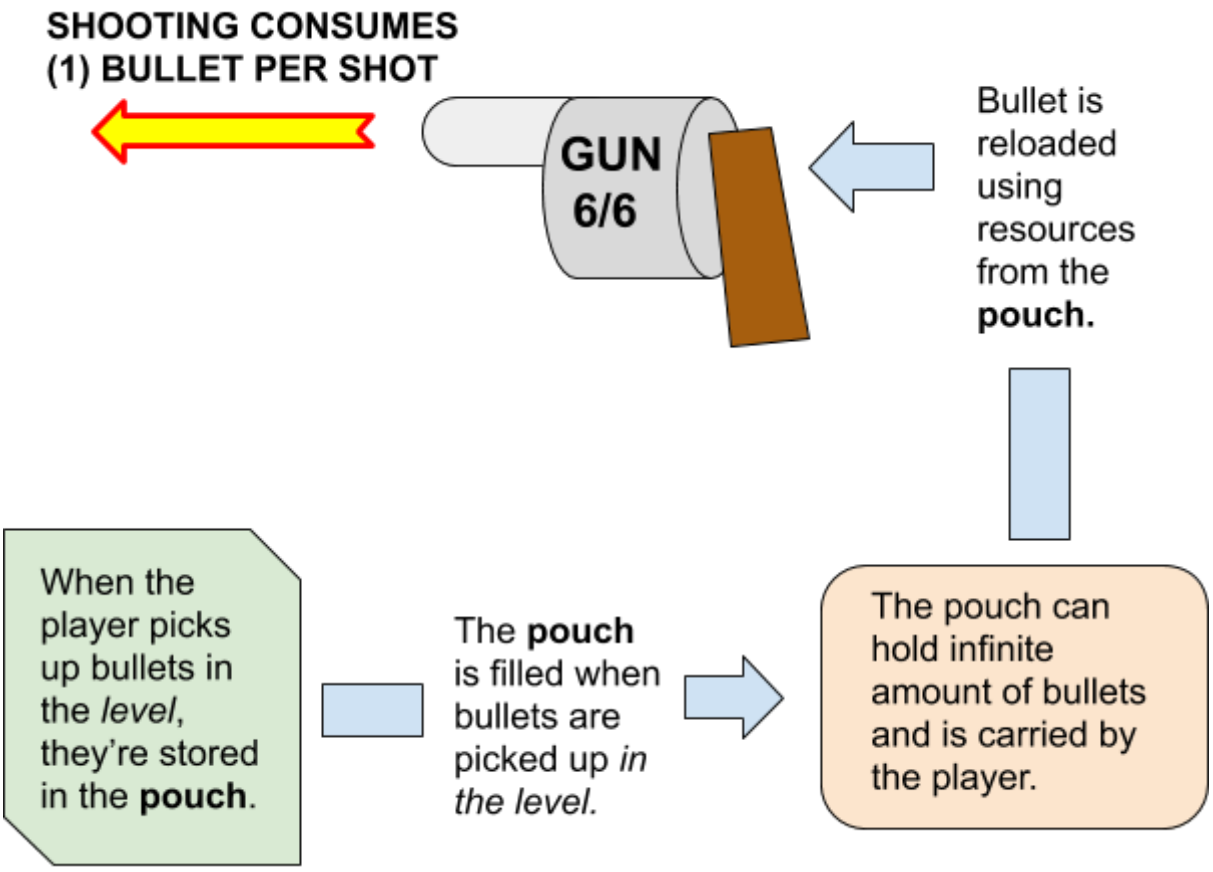
Eg. if the revolver has **0 bullets**, it will be filled with **6 bullets.**

If the revolver has **2 bullets**, it will be filled with **4 bullets**.

When filling the revolver, bullets are retrieved from the players *pouch:* bullets retrieved from the world will be stored *directly into the pouch.* There is **no limit to the amount of bullets within the pouch**.

Back to the table of contents

The exemplary picture belows shows the different states bullets can reside in:



2.5.3. Effects on creatures

Variables such as *type* of creature, and during which *circumstance* determines the effect that will occur when firing at enemies.

A regular *Jerry* that isn't lit by the flashlight will require **2-3 bullets to kill it**: if the Jerry is *completely engulfed in light*, i.e reached a threshold in amount of light, the Jerry will **only require 1 bullet to kill**.

When shooting *Hugo,* it will simply be slowed/stunned, rather than killed: this effect occurs when shot *anywhere but it's back.*
If the Hugo is shot in the back instead, **the stun will remain for a longer period of time**.

Back to the table of contents

2.6. Inventory

2.6.1. Inventory space

The player has an inventory that they can manage. In this game the inventory-system is based of **slots**: when an item is added to the inventory, no matter what type of item it is, it will take up a *slot*.

The total amount of slots is to be based on the **[inventory_max_slots]**; the inventory cannot contain more items than there are slots.

If the player attempts to pick up another item while already fully stocked, the other item will *remain in the world*. The player will have to drop an item to create *an empty slot*.

Two or more identical items will **not stack in the inventory**; they'll take up a slot for each item.

2.6.2. Inventory management

In order to manage their inventory the player will have to open up a separate *inventory menu*. The inventory menu is opened by *holding down the designated* **[inventory_key]**. The inventory menu will only be open as long as the **[inventory_key]** *is being held down;* once the designated key is released the inventory menu will fade out.

In order to **drop** or **equip/use** an item the player uses their *mouse* as a tool for interacting with items in the menu.

When the player hovers over an item's icon in the inventory with their mouse, *options for dropping* and an option for *using/equipping* items are to show up by the item's icon.

Back to the table of contents

The player simply *clicks* on either one of the options with the *left mouse-button*. When an option has been selected, the menu will fade and the action will be executed.

Dropping an item will simply leave it as an object in the gameworld. The player can pick up the dropped item again, if the player has an empty *slot in their inventory*. Dropping an item will as stated *close the inventory menu*.

Using or **equipping** an item will be detailed further in the *Items-section*.

Back to the table of contents

2.7. Items

2.7.1. Usable items

An item that falls under the *usable category* is an item that can be used directly from the inventory. Items like these can include for example:

- First-aid kits
- Spare batteries

When an item is used from the menu, it will be instantly *consumed* and *removed* from the inventory. Depending on what type of item it is, it will have different effects.

Back to the table of contents

2.7.2. Usable items list

Name	Function
First-aid kit	Using a **first-aid kit** from the inventory will restore the player's health to completely restored. When the first-aid kit has been used, it is instantly consumed and removed from the inventory.
Spare battery	Using a **spare battery** will add *some extra consumable time* to the flashlight's **[flashlight_max_power]** value. The time that is to be added is decided by the **[flashlight_battery_time]** value. As stated, this time is *not permanent* - once it has been consumed it is expended permanently and cannot be replaced, except by adding another **spare battery**. Adding several spare batteries *will not stack the effect* - only a maximum of **[flashlight_battery_time]** can be added to the flashlight's power-meter. *Reference: Binding of Isaac - **Soul Hearts**.*

Back to the table of contents

2.7.3. Equippable items

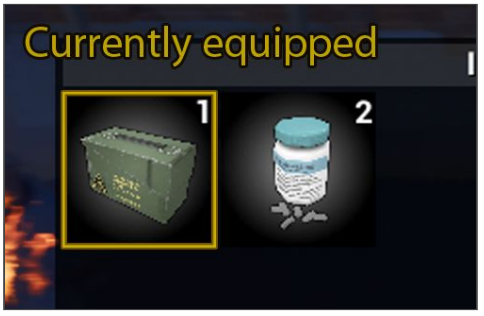
Items that are to be **equippable** by the player are items that are to be used in the *Aiming-state*. Items like these include:

- Pistol
- Flare
- Empty can

In general these are items that can be used to **interact** with the gamespace and the creatures within it.

In order to **equip** an item the player follows the procedure of hovering the mouse cursor above the item in the *inventory-menu*, and then selecting the **use/equip** option;
if it is an equippable item the option should state "*Equip*".

If the player hovers the mouse-cursor over an already equipped item, the options will state **unequip** and **drop** instead: the *unequip* option will remove the item from the players active item, but retain it in the inventory.



The player can only have *one item equipped* at all time; if the player equips another item from their inventory, that item will be equipped while the former equipped item will be *unequipped automatically.*

Back to the table of contents

2.7.4. Equippable items list

Back to the table of contents

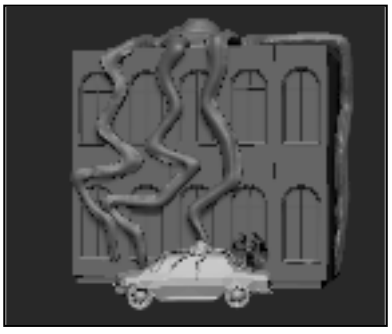
3. Enemy behaviour

3.1. Retreating & enemy safe zone

When an enemy that is hit by light retreats, they are to flee to a nearby **safe zone**: areas that function as enemy encampments.

3.1.1. Communal safe zones

Communal safe zones are spots that are distinguished by *creature-like structures* or *themes.* These spots are to be **patrolled** and guarded by creatures.



These safe zones are to be *static areas* in the game that the enemies can retreat to if startled by the player. Once they've retreated to the *communal safe zone* and calmed down, the creature will return to its patrolling route.

The player *can enter these zones*, but there is no inherent effect to it. The player cannot *disband* or *remove* a *communal safe zone*.

3.1.2. Environmental safe zones

In more distant or obscure parts of the city, the creatures can escape into *environmental safe zones.* Environmental safe zones can be both *static areas in the world* such as the *communal safe zone* or *areas in which the creature can disappear/run off.*

These safe zones function similarly to the *communal safe zones* in that the creatures will escape to these zones *when they've been hit by light.*

Back to the table of contents

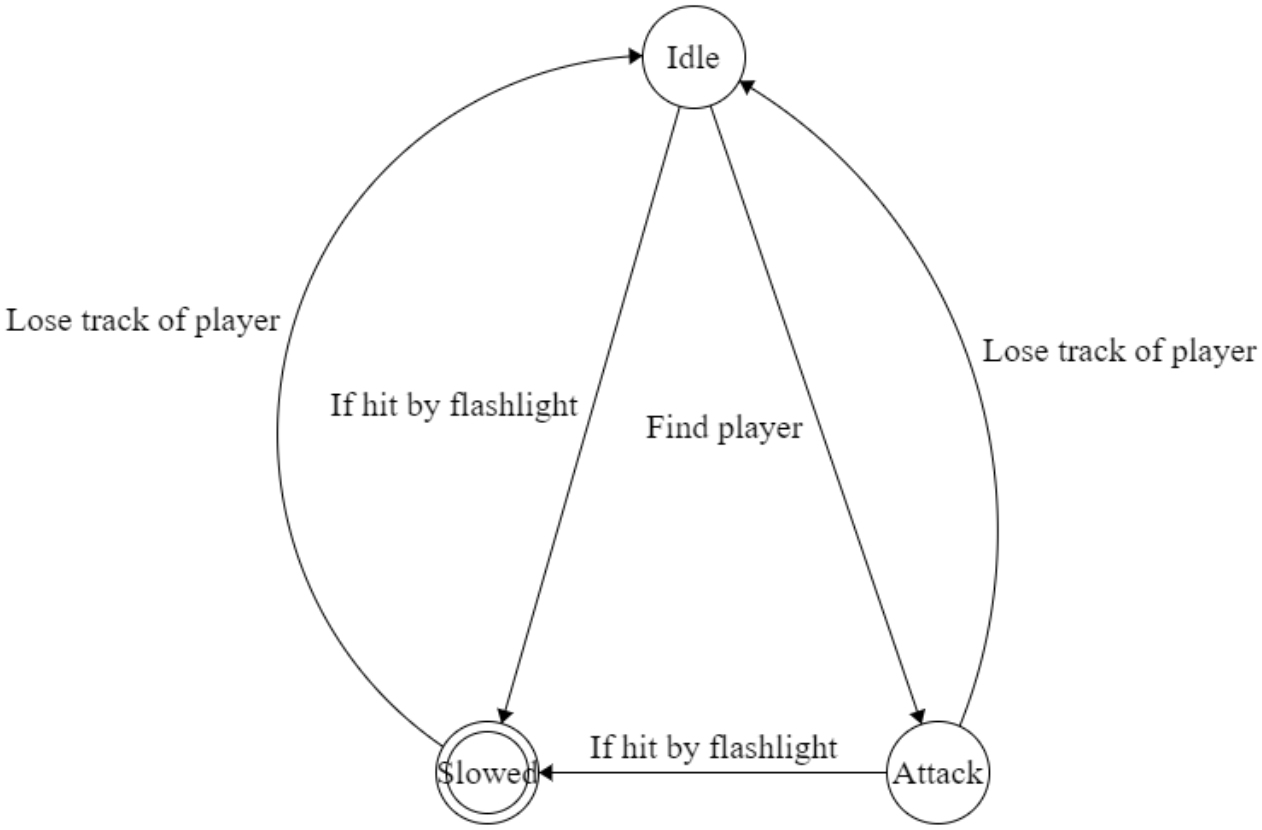
3.2. Enemy types

3.2.1. Jerry

This enemy will lurk in the dark and patrol. They have sensitive hearing and will seek out any living being that isn't one of their own in order to either breed or feed.

They patrol areas by idly walking about, either in a specific location looking for prey or around blobby building tentacles. If they pick up a noise, no matter what caused it, they will investigate it.

These creatures are extremely sensitive to *bright lights,* and when shone on by a lightsource, their bodies will start to malfunction, rendering them *slow-moving*. They'll continue pursuing the player, although at a diminished rate.



Back to the table of contents

3.2.1.1. Jerry group behaviour

-

Back to the table of contents

3.2.2. Critter

Smaller variants of the *Standard Enemy* - although these buggers are much *smaller* and a lot less *dangerous.* They will always idle around, seamlessly without aim. These enemies will not *seek out the player;* they will simply wander around aimlessly in a certain area.

If the player flashes a *Critter* with the flashlight, they will run away in the same fashion as the *Standard Enemy*.

3.2.3. Screamer

Stationary enemies that have clambered onto wall. Although they are immobile, they can still harm or disrupt the player.

There can exist both dangerous *wall huggers* that can either spit projectiles at the player or attack, albeit with a limited reach.

3.2.4. Hugo

Large, lumbering enemy that is *provoked* by light, rather than bamboozled. Will attack the player more ferociously when struck by light.

4. Level design

4.1. Tutorialization

As the player starts the game there is to be some sort of **tutorial** for the player, in order to explain the core mechanics of this game.
The philosophy of these tutorials is to integrate them into gameplay and refraining from *taking control from the player*.
Which method that is to be used as tutorial has to be tested properly first, but the decision stands between **tutorial integrated into the game-space** or a **tutorial based on UI-elements**: both methods will be detailed in this section.

Back to the table of contents

4.2. Option 1: Tutorial integrated into gamespace

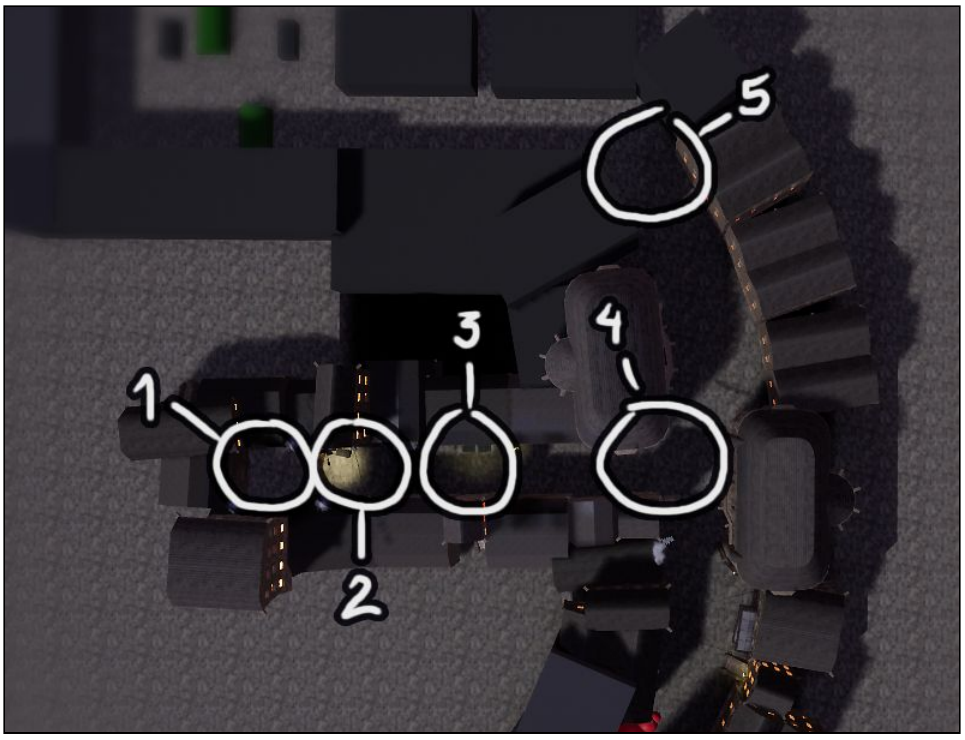
This type of tutorial is to be integrated directly into the game's *world,* rather than as some sort of extrinsic UI. This means that the player will run into messages that are a part of the game's settings: such as through *graffiti, signs etc.*

These messages are to appear by and by as the player moves along the game. In which order the messages appear is determined by what the player needs to know in advance *eg. aiming to use items.*

Back to the table of contents

4.2.1. Tutorial-layout for diegetic tutorial

The order in which these messages are to appear in, and in which locations can be seen in the *exemplary image below:*



1. How the player **moves** in the game:
   The message will simply state *WASD MOVE*; the aesthetics of this message is up for interpretation.

2. How does the player use the **flashlight**:
   The player starts of with their flashlight *turned off.* Firstly this message will display *F for FLASHLIGHT* or something similar: which will lead to the player turning their flashlight *on.*
   After the flashlight prompt, the message will state *RMB to AIM:* once the flashlight is on the player will be prompted to *aim their torch.*
3. How do the player **pick up** and **use items**:
   The player will at *location 3* find an *item* that is readily visible on some dumpsters: on the message above the dumpsters it will say:

Back to the table of contents

SPACE to PICK UP or something akin.
There is also the added HUD -element that informs the player to pick up an item.

After the *pick up prompt,* the message will state:
LMB while AIMING TO USE. This prompt explains how the player uses items. It is of importance that the very first item the player will find is *not so vital it is detrimental for progression if it is used.*

4. How do the player manage **several items**:

At *location 4* the player is to find another item that will be relevant for this section; *the inventory.* The message at this spot would state: *TAB for INVENTORY,* which prompts the player to open their inventory.

Any instructions regarding how to interact with the inventory menu itself is not vital for a graffiti message, as the player already has learnt to explore the game space using *their mouse cursor.*

5. How to introduce **sneaking** to the player:

When the player reaches *location 5* they will stumble upon a message that prompts them to *sneak;* the message itself should state something akin to: *CTRL to SNEAK*. The reasoning for this message's location is to prepare the player for the first *Jerry,* as well as the first *challenging area.*

Back to the table of contents

4.3. Option 2: Tutorial based on UI-elements

This method of tutorial, rather than being a part of game world, is instead an *extrinsic element of the game*. When the player reaches a relevant point in the introductory section of the level, they will be prompted with actions to perform, such as *walking* or *aiming;* upon completing said actions, *the prompts will fade out*.

When, where and how the prompts are to show up will be detailed in the section below, but it is important to notice that the idea of these prompts are that they'll only appear *once;* they'll only disappear once the player has completed the task, or once the player proceeded through the level.

4.3.1. Tutorial-layout for UI tutorial

The order in which these messages are to appear in, and in which locations can be seen in the *exemplary image below:*

Back to the table of contents

The exemplary picture above describes where each of the tutorial prompts will take place - in this part the *function* of each prompt will be detailed:

1. How the player **moves** in the game:
   In the first area of the game, the player will be introduced to **moving**. This will happen through a prompt that displays the **WASD** key-layout and the inclination to *move*.

   This prompt will disappear once the player has inputted each and every one of the W, A, S and D-keys. Otherwise, see the tutorial-prompt functions regarding when the prompt will disappear otherwise.

2. How does the player use the **flashlight**:
   In this area the first alley will be a bit darker; a tutorial-prompt will appear to tell the player to push ***F*** *to toggle your flashlight.* This prompt is to disappear once the player actually toggles the *flashlight.* It is important to note that the player starts the game with their ***flashlight turned off***.

   Once the player has turned on their flashlight, they will be introduced to a prompt that inclines the player to **aim their flashlight**. The prompt will state: *"Hold RMB (right mouse button) to aim your flashlight".* This prompt will disappear once the player has successfully *aimed with their flashlight.*

   Similarly to the movement prompt;
   if the player *doesn't* follow to prompt-instructions, the prompts will disappear according to the tutorial-prompt functions section.

Back to the table of contents

3. How to **pick up items**, and **how to use them:**

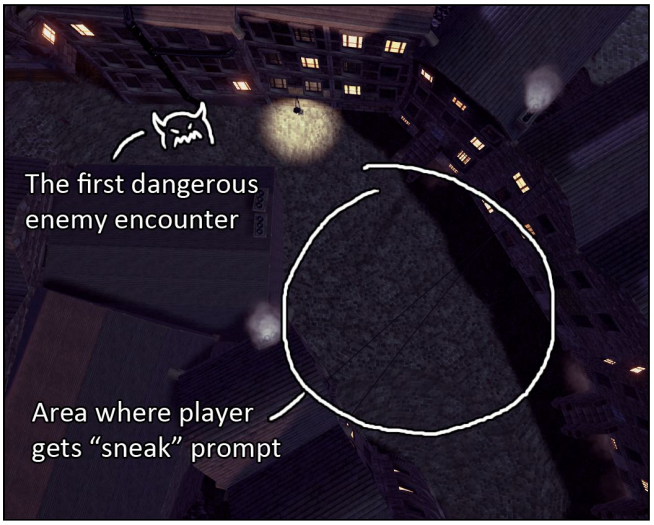
After the section with the flashlight, the player will stumble upon a set of dumpsters with an item on top of it. When the player is in the items vicinity, a prompt to pick it up will appear, stating: *“F to pick up items”*.
This prompt is to disappear *after* the player picks up the item.

If the player picks up the item, the first prompt will disappear and cause another prompt to show up, stating: *“LMB while aiming to us hold item”*. This prompt is to incentivise the player to *use the first item they’ve found*.

It is of importance that the first item *isn’t vital for progression*.
The player should be free to *experiment* with the first item they find.

4. How to introduce **sneaking** to the player:

At *location 4* in the game, the player will get a prompt to *“Hold SHIFT to sneak around monsters”*. This message will appear in close proximity to the very first Jerry, where the message will be relevant in gameplay immediately.



Back to the table of contents

5. How to introduce **item management** for the player:
   The player's **inventory** will not be introduced until they've actually picked up their **second item in the game**. It does not matter which item it is; once the **second** item has been picked up, a prompt will appear that states: *"Hold TAB to manage your inventory"*.
   This prompt will disappear once the player successfully presses the tab to open up their inventory.

   There will be *no instructions regarding interacting* with the inventory using the mouse-cursor. Until this point, the player has already explored the gamespace using their cursor; in theory the player will use their cursor to also explore the *menu-space*.

4.3.2. Tutorial UI-prompts functionality

This section will specifically detail how the prompts will behave in case of the player not removing prompts by *fulfilling the instructions.*

Once a *prompt* has appeared for the player, it will remain until one of these conditions has been met:

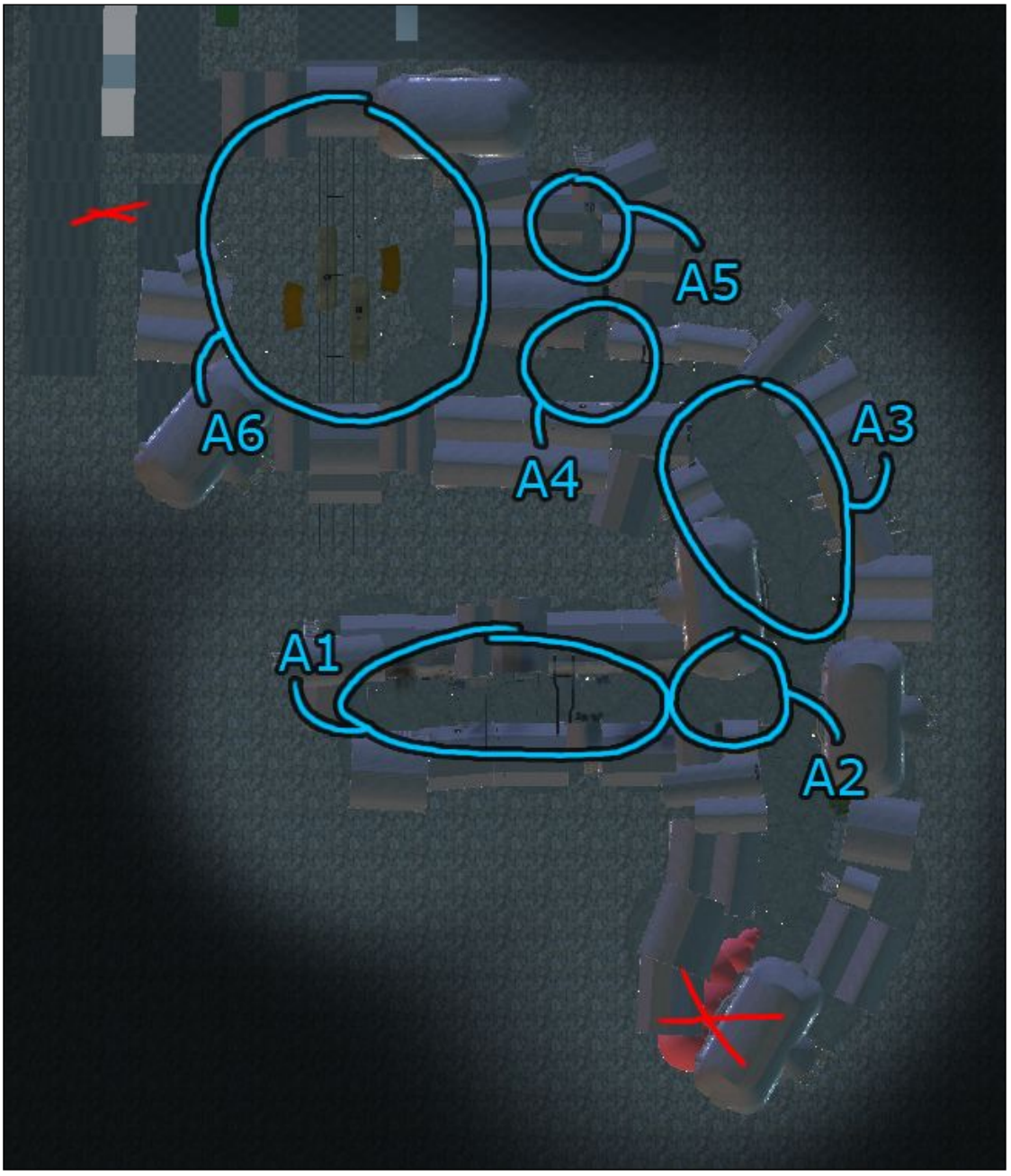
- If the player *completes the prompts instructions* eg. moving when inclined to move.
- If the player moves onto another location or trigger that would trigger another prompt, the *current* prompt would be replaced with *whatever prompt the player trigger* in the game.
- And lastly, if the player ignores the prompt, it will disappear on its own after a set amount of time that is dictated by the **[prompt_remain_timer]** value.

Back to the table of contents

4.4. Level design overview: Section 1 / Intro

This section of the game design document is to go into greater detail regarding the different sections of the level; this part will go through the *introductory stage of the level.*

The image below exemplifies *points of interest* in the level, which is detailed further in the section *after the image:*



Back to the table of contents

A1. Tutorial alley

The player starts out initially in the **tutorial alley**: once gameplay starts, the player is to be met by *graffiti-covered walls that displays the game's controls.*

The graffiti is to introduce some of the *core mechanics of the game*: once the player leaves the alley, they should know how to:

- How to **walk** and **sprint**
- How to **turn on/off** and **aim** the flashlight
- How to **fire** and **reload** the revolver

Before leaving this section, the player will have to pick up a *revolver* that has **[bullets_starting_amount]** already loaded - this way the player will be fully equipped to explore the rest of the game.

Another noteworthy thing is to use *lights* to highlight tutorials;

When starting out, the player should *immediately be inside a lightsource:* the same lightsource that highlights the first *movement-graffiti.*

The lightsource that highlights the *flashlight graffiti* should be from a separate source than a lamp; i.e from a *window instead.* This is so that the flashlight will be distinctly visible from another lightsource.

Back to the table of contents

For the final graffiti piece, it can be laid out on the ground in some color, *but not be highlighted by light:* let the player put their newly found flashlight *to use.*

Finally, by the flashlight tutorial the player should also encounter some *Critters.* This can be a good way for the player to learn that their flashlight also *affects the creatures.*

A2. Crossroad

At this point the player has learnt how to play the game; now the player will set off into the game itself.

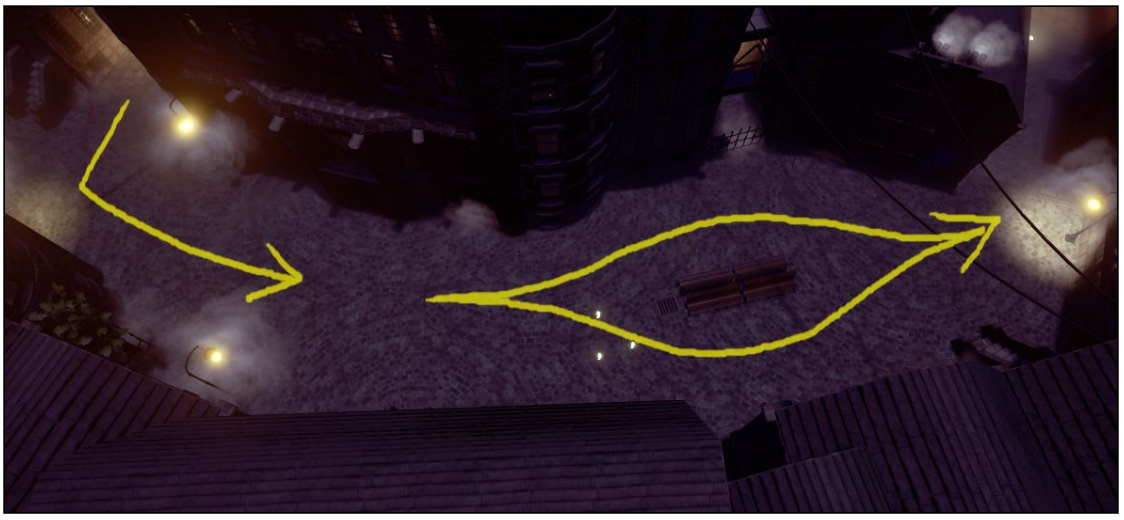
When entering the crossroads, the player can be prompted to use their flashlight on some sort of creature:
as a test to see if they've understood the core mechanics of the game. This obstacle can also be placed *right before entering the crossroads.*

During this segment, there can be an **event** where a *Jerry* runs past the player. This *Jerry* will be the same enemy that the player will run into later.

Back to the table of contents

A3. Bench street

This street directly follows the *crossroad*.



In this street the player will come across their *first Jerry* - the same Jerry that runs past the crossroad. When the player successfully flashes the Jerry, it will *escape into an alley to disappear.*
There is to be no *streetlights to assist the player here:* teach the basic functionality of the enemies firstly.

Otherwise, this alley can be utilised to build suspense up until the second encounter with a Jerry.

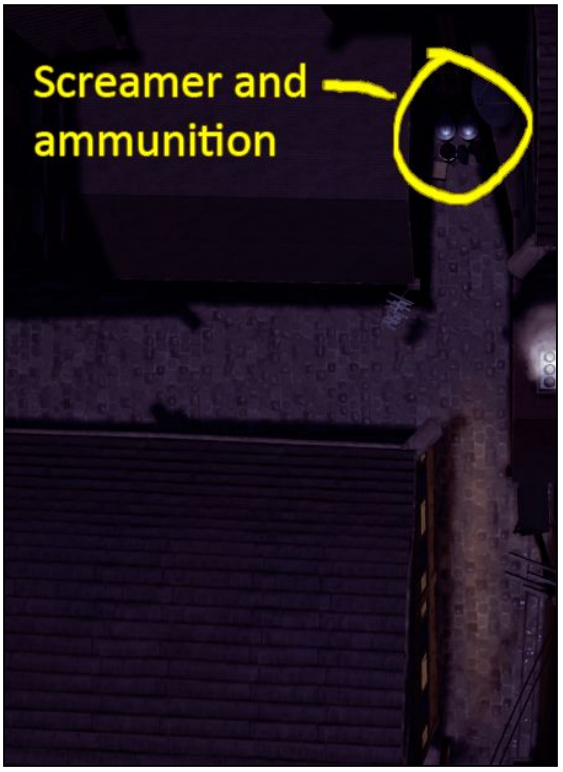
A4. Streetlight path

This alley, specifically, is to introduce *streetlamps,* and their *function in gameplay.* The player is to face another *Jerry* in this alley, that can be set to be more aggressive or likewise - so that it might *enter the streetlight.*

The path leading to the train-station is to be more inviting and lit, while the *thin alley should be more dark and uninviting.*

Back to the table of contents

A5. Dark alley



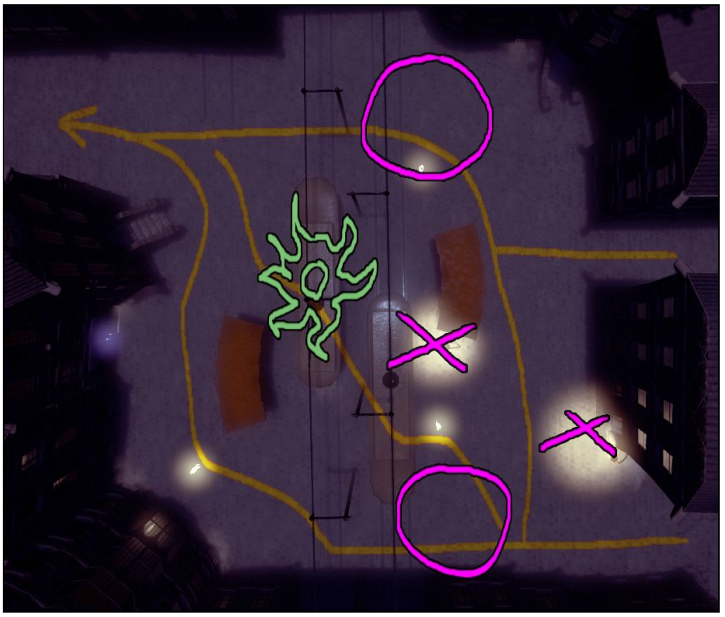
This alley is an option to the streetlight path: in the end of this alley, in the dark, unlit corner there is a *stationary* ***Screamer***.

When this Screamer is shone with light, it will shriek loudly. Right by the *Screamer,* the player can also find some ammunition for their revolver, as a reward for getting spooked.

Also, the alley leading up to the Screamer can be used to amplify the tension, using ambient tracks and so on. This can only play while in the alley though.

A6. Train station

The *train station* will primarily introduce *several enemies at once, larger areas* and the *safe spot blobs*.



There are also to be some *streetlights in this area,* to lead the player, and to give them some leeway and breathing space when passing through.

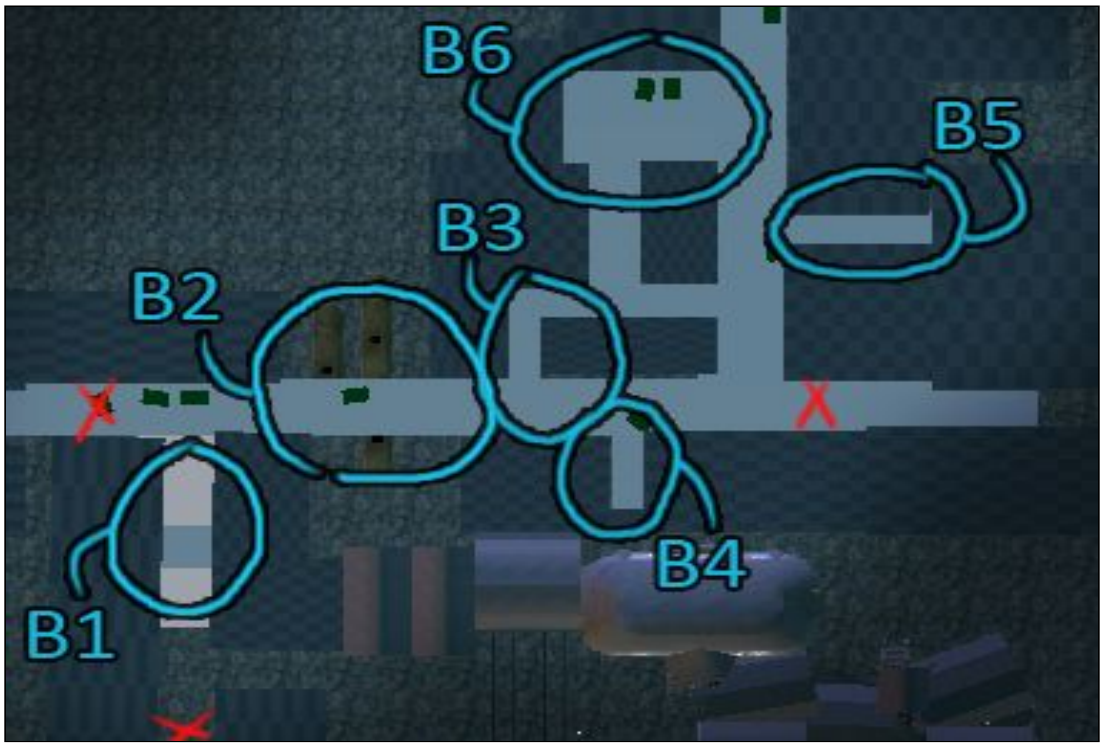
In the middle of the station, there is also a large blob that the enemies are *attracted to.*

Back to the table of contents

4.5. Level design overview: Section 2 / Middle

After overcoming the train station, the player will enter the *second section of the game,* which will be referenced to as **the mid section**.

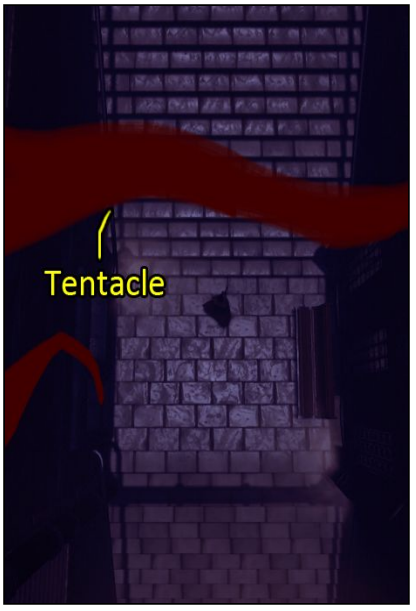
Below is an exemplary layout of the map, and details about *points of interest:*



B1. Staircase alley

This alley is designated to be a bit *calmer*; after the hectic *train station.* This does not however mean that the alley will be *contentless.*

When walking up the alley, there should be *parallaxing tentacles* arching overhead the alley, to indicate that the player is entering bad territory.



Back to the table of contents

B2. Bridge

The bridge is a scene that is dedicated to induce

Back to the table of contents

4.6. Level design overview: Section 3 / Finale

This section will specifically detail how the prompts will behave in case of the player not removing prompts by *fulfilling the instructions*.

Back to the table of contents

5. HUD & UI

5.1. Battery

The batteries power supply will be represented by a small white/grey bar in the lower right corner of the screen. When the battery is drained of its power the bar will shrink from the right to left. There will be an outline on the bar so that the player is able to see how much och the battery is depleted.

5.2. Cursor

The cursor will be represented by a decal in the shape circle with a fully filled in middle and fades towards the edges.

5.3. Health

There will not be a health bar in the game instead when the player is hit one time the a red tinge will appear on the screen accompanied by a raised heartbeat and heavy breathing sounds. Get hit twice and you die. If you die the screen will darken and the red tinge will still be there. The user will also get an *“Quit to main menu”* and a *“Continue”* option. These options will either quit the current game and take the player to the main menu or will restart the player at the last checkpoint.

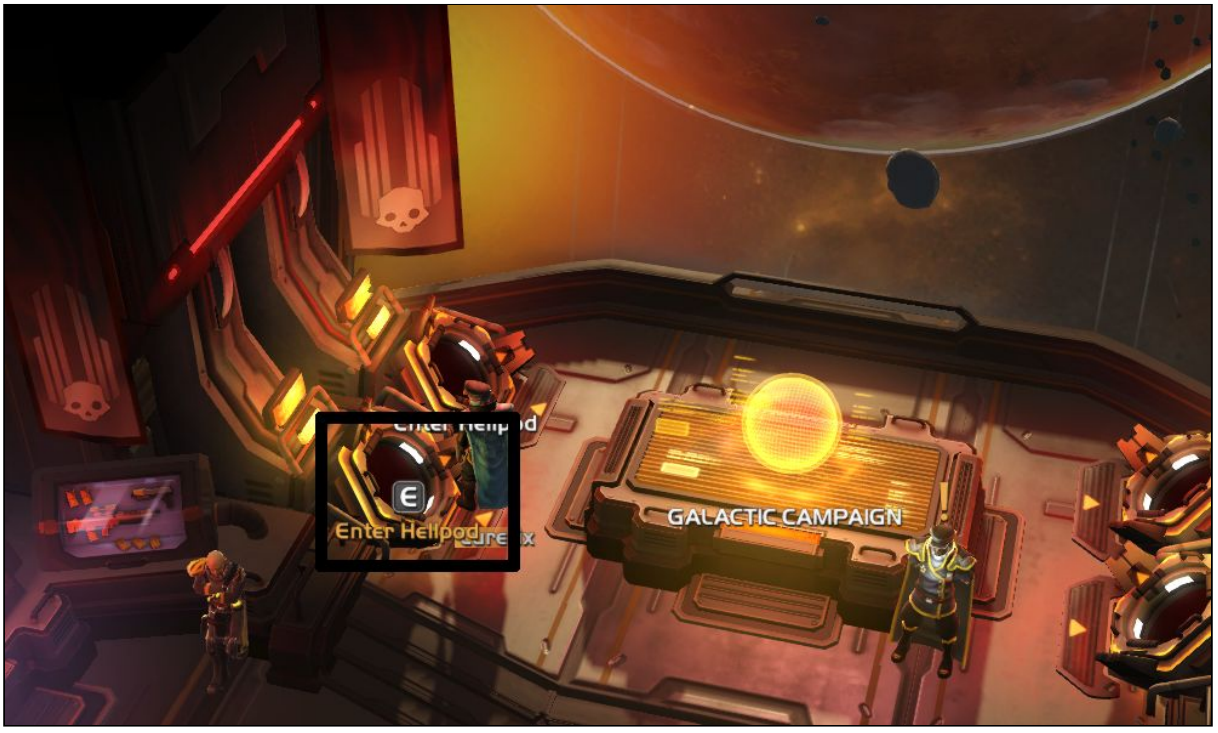
Back to the table of contents

5.4. Inventory

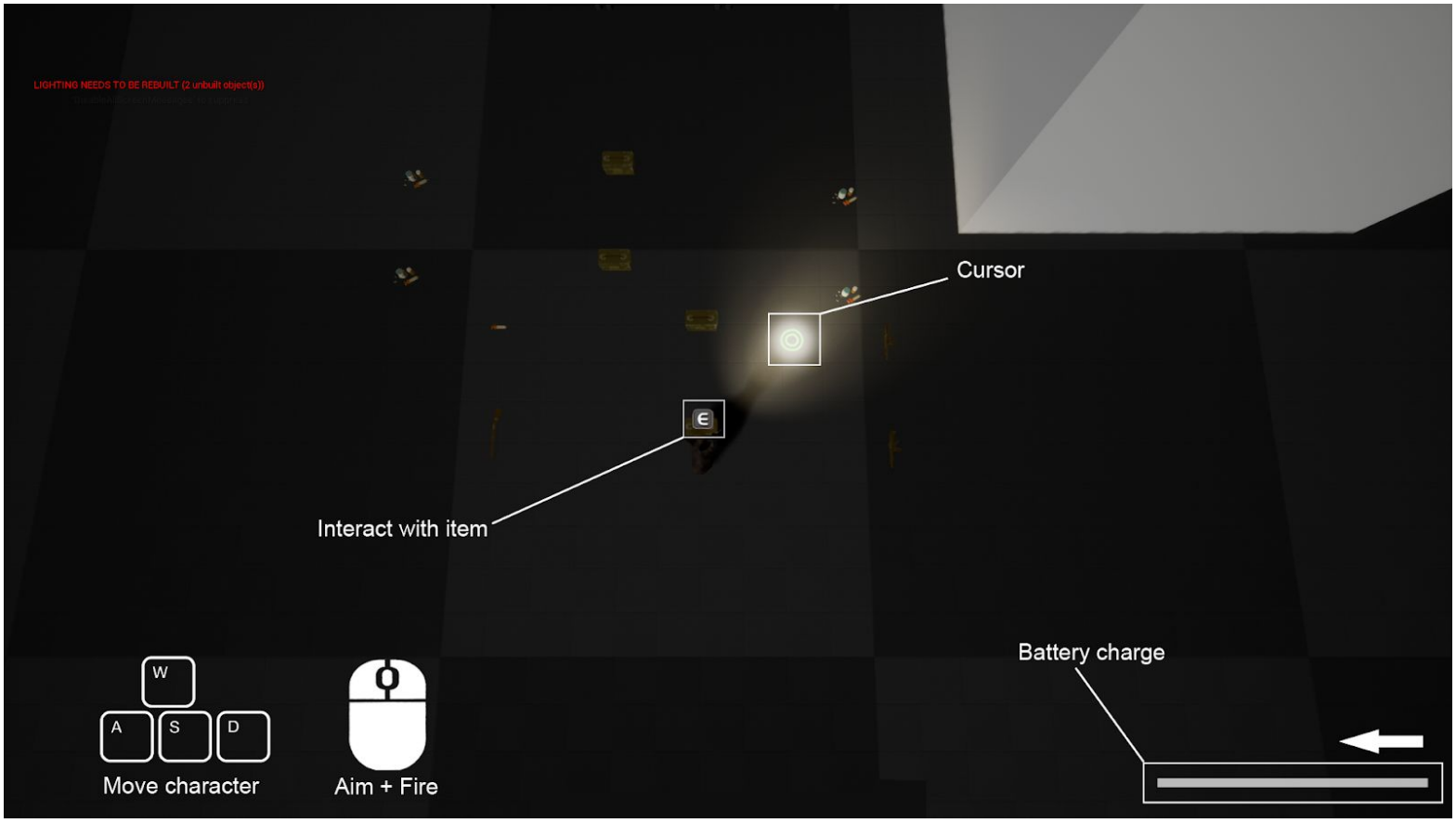
The inventory will show up as long as the player hold down the **[inventory button]** There will be 3 icon slots in the inventory. The player can hover the cursor over icons of items and when they does that the item's options will show up over the item. By clicking on either of the of the option buttons the player will do that specific command with the accompanied item.

5.5. Pick ups

When the player stands near an object that is abled to be picked up by the player, a icon for the **[interact with item button]** will show up. This button will pulsate, informing the player that it is clickable.



[Back to the table of contents]

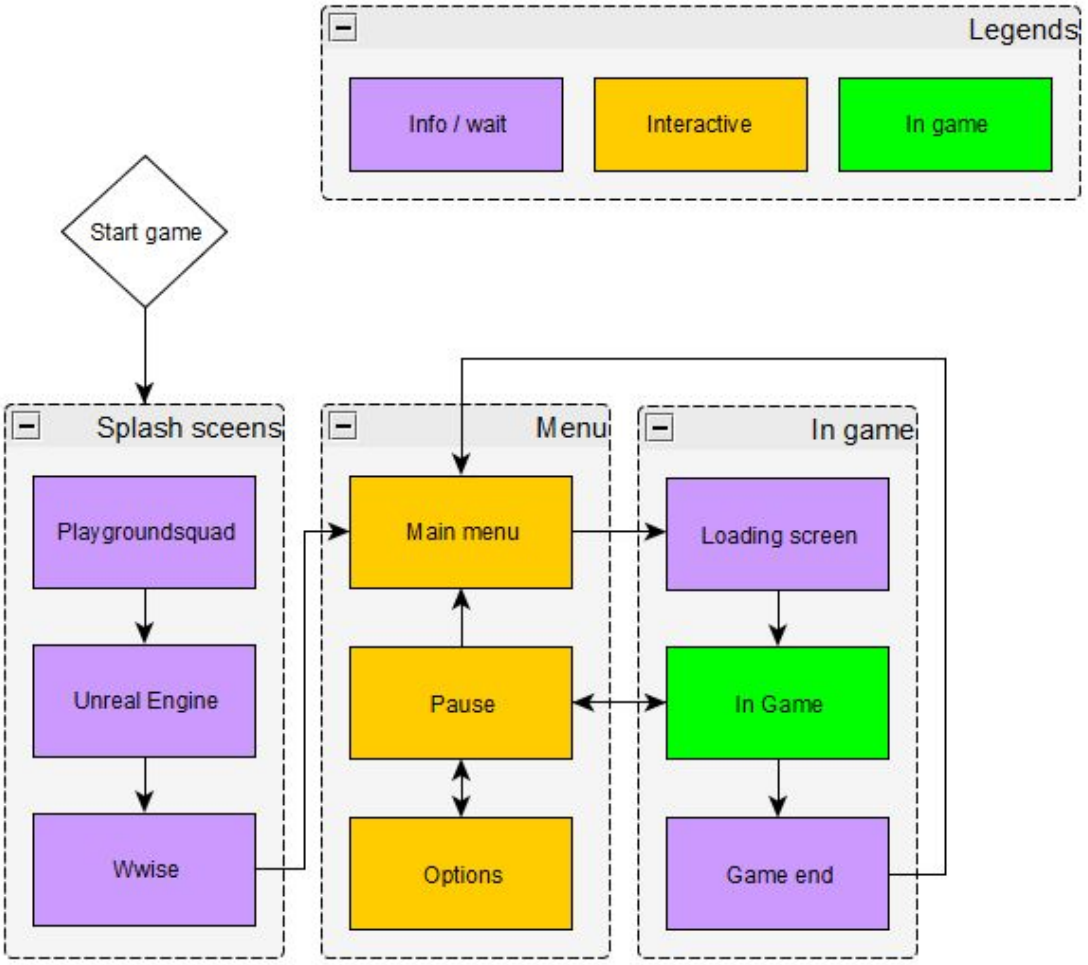


Back to the table of contents

6. Setting

Back to the table of contents

7. Flow chart / Wireframes



Wireframes summary

Start screen	**A**
Pause screen	**B**
Quit confirm screen	**B.3**
Options screen	**C**
Death screen	**D**
Credits screen	**E**

Back to the table of contents

Start screen

A.

Text: None

Art asset: Yes, some kind och view of the city or something else

Clickable: No

A.1

Text: “name of the game”

Art asset: Yes

Clickable:No

A.2

Text: “Press any key to start”

Art asset: Yes

Clickable:Yes

Function: Starts the game. Goes to in game

Click feedback: Yes

Back to the table of contents

A.3

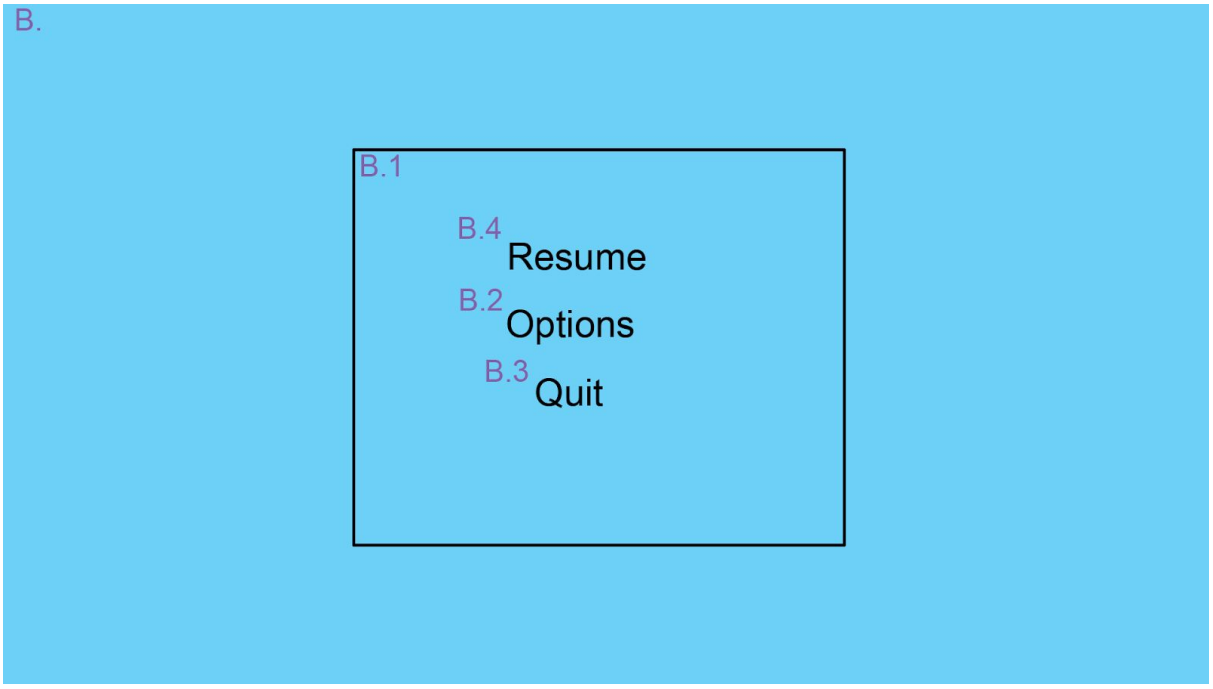
Text: “Quit game”

Art asset: Yes

Clickable:Yes

Function: Quits the game. Goes back to desktop.

Click feedback: Yes



B.

Text: None

Art asset: No, this is in game paused.

Clickable: No

B.1

Text:None

Art asset: Yes

Clickable:No

Function: Box that pops up over the paused in game

Back to the table of contents

B.2

Text: “Options”

Art asset: Yes

Clickable: Yes

Function: Goes to screen **C** “options”

Click feedback: Yes

B.3

Text: “Quit”

Art asset: Yes

Clickable: Yes

Function: Goes to screen **B.3** “Quit confirm”

Click feedback: Yes

B.4

Text: “Resume”

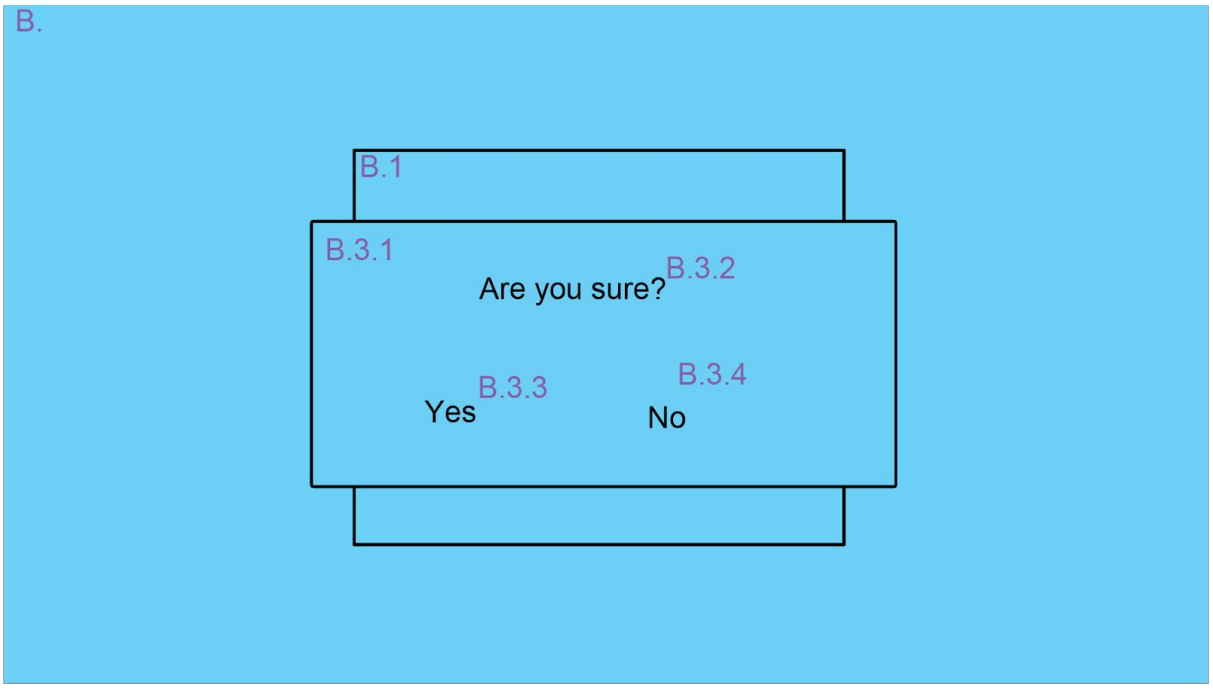
Art asset: Yes

Clickable: Yes

Function: Goes to back to in game

Click feedback: Yes

Back to the table of contents



B.3.1

Text: None

Art asset: Yes

Clickable: No

Function: Box that pops up over the in game and the pause box

B.3.2

Text: “Are you sure?”

Art asset: Yes

Clickable: No

B.3.3

Text: “Yes”

Art asset: Yes

Clickable: Yes

Function: Confirms that you want to quit the game, the game will shut down.

Click feedback: Yes

B.3.4

Back to the table of contents

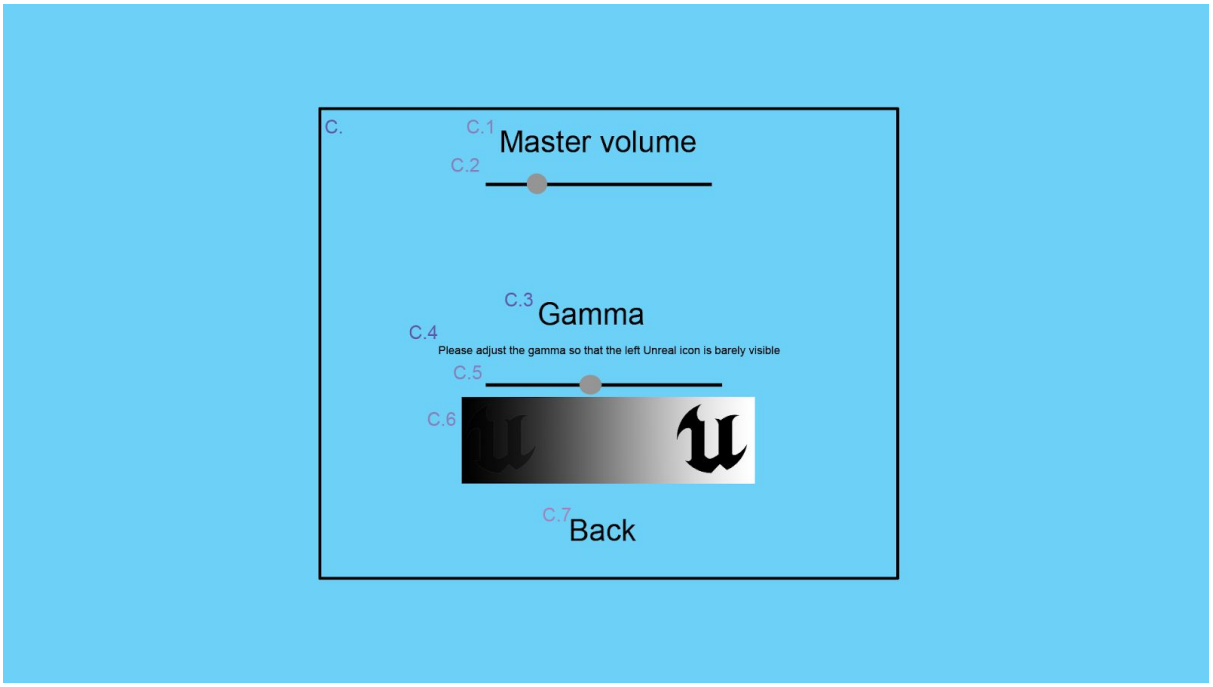
Text: “No”

Art asset: Yes

Clickable: Yes

Function:Confirm that you don't want to quit. Goes back to **B.** “pause screen”

Click feedback: Yes



C.

Text: None

Art asset: yes

Clickable:No

Function:Box that pops up over in game and the pause box

C.1

Text: “Master Volume”

Art asset: Yes

Clickable: No

Back to the table of contents

C.2

Text: None

Art asset: Yes

Clickable: Yes

Function:Slider that lets the player choose how loud the game will be.

Click feedback: Yes

C.3

Text: “Gamma”

Art asset: Yes

Clickable: no

C.4

Text: “Please adjust the gamma so that the left unreal icon is barely visible”

Art asset: Yes

Clickable: No

C.5

Text: None

Art asset: Yes

Clickable: Yes

Function:Slider that lets the player choose how dark or bright the game will be

Click feedback: Yes

C.6

Text: None

Art asset: Yes

Clickable: No

Function:An indicator that shows the player how bright the game currently is.

C.7

Text: “Back”

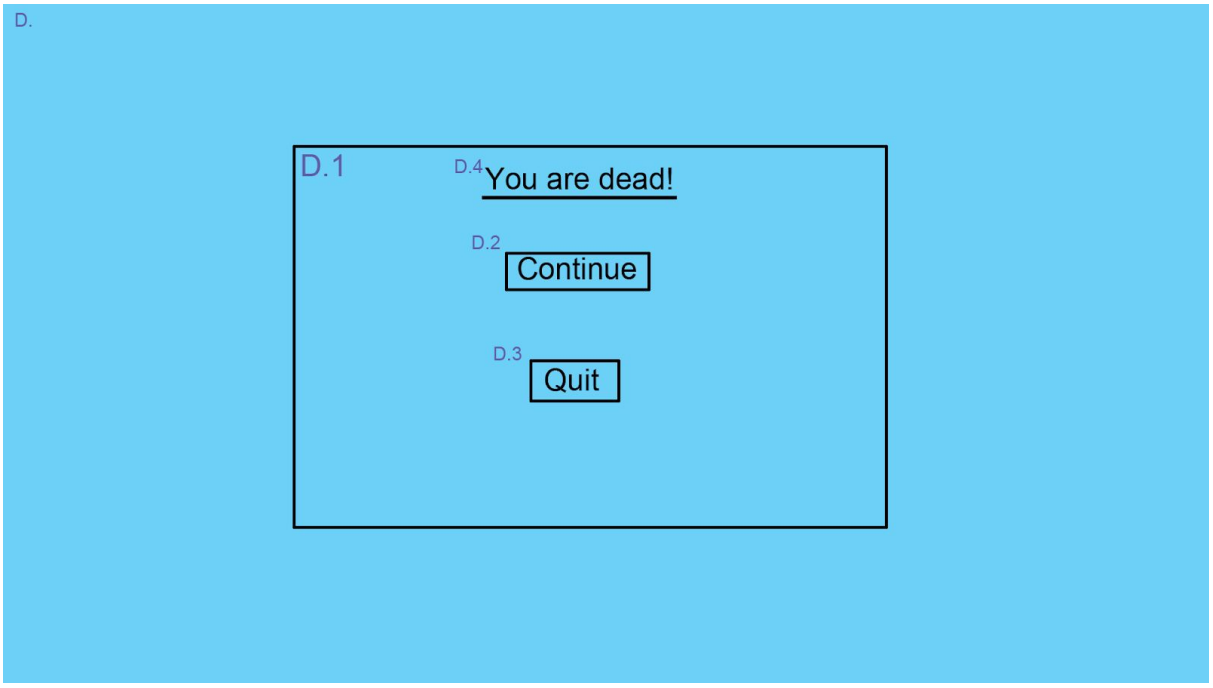
Art asset: Yes

Clickable: Yes

Function:Goes back to the **B** pause screen.

Click feedback: Yes

Back to the table of contents



D.

Text: None

Art asset: no

Clickable: No

Function: In game screen

D.1

Text: None

Art asset: Yes

Clickable: no

Function:Box that pop up over the in game screen when the player dies.

D.2

Text: “You are dead!”

Art asset: Yes

Clickable: No

Back to the table of contents

D.3

Text: “Continue”

Art asset: Yes

Clickable: Yes

Function:Goes back to last checkpoint in game

Click feedback: Yes

D.4

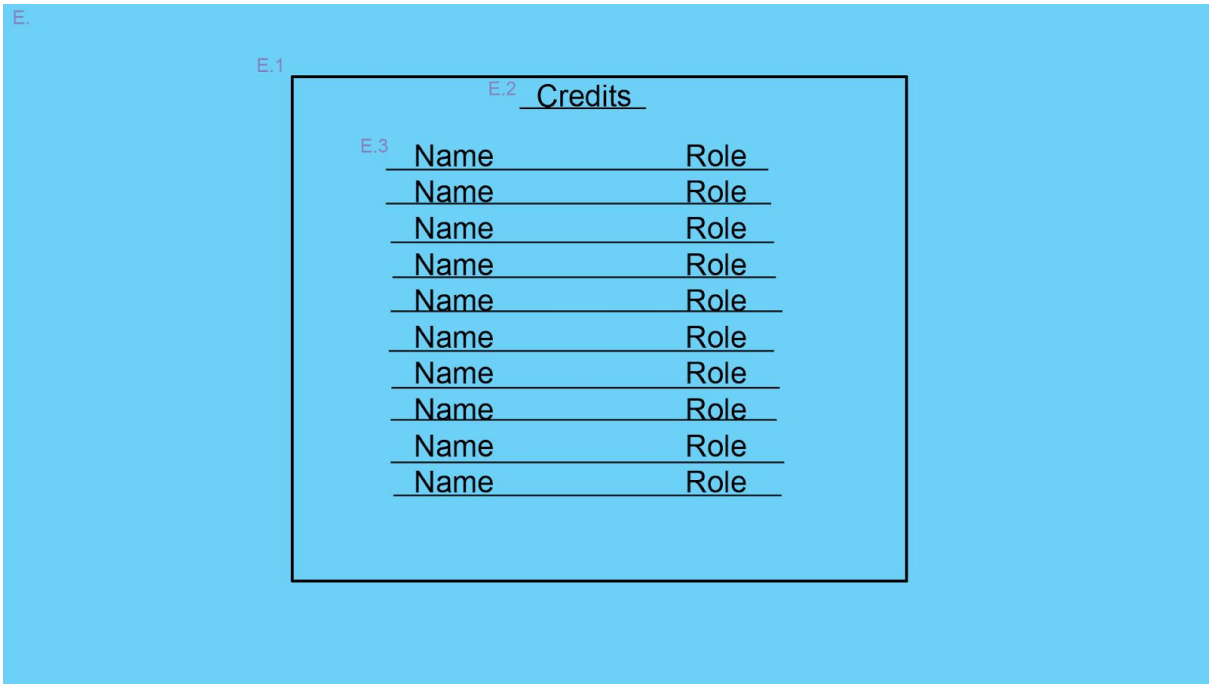
Text: “Quit”

Art asset: Yes

Clickable: Yes

Function:Goes back to the **A** Start screen.

Click feedback: Yes



E.

Text: None

Art asset: Yes

Clickable: No

Function:Background the when the credits roll. TBD what it should be.

Back to the table of contents

E.1

Text: None

Art asset: Yes

Clickable: No

Function: Box that pops up in front of the TBD background.

E.2

Text: “Credits”

Art asset: Yes

Clickable: No

E.3

Text: The names and roles of the people in the group

Art asset: Yes

Clickable: no