Escape Fort Thunder

# Overview

A dungeon crawl on a mobile device made to look like it is all done in Ascii. The few controllers will be superimposed over the top, breaking the illusion a little, but if it is done right then it should work just fine.

Text

Description automatically generated with medium confidence

# Story

Fort Thunder was the logical place for locals to flock to when the infection started. It was an old military structure built in the late 1800s and was repurposed as a fallback shelter as the infection spread throughout the land. Fear that the disease would breach the fort’s walls led to drastic measures being taken, many people were turned away at the gate for suspicion of contracting the virus, and all entry points were barricaded except for the main gate which was locked up tight.

A few sentences will be displayed as a new game is created. “When the infection hit, Fort Thunder was a haven for survivors. Now it is a place of gore and death. You must escape…”

More details about the fort’s history will be available to the player through an item called ‘literature’.

# Start Scene

The background should be interesting Ascii art with a simple player selection. The Player should be able to select New Game, Load Game, or Quit.

Text

Description automatically generated

# New Game Scene

This is where the player will select the World’s and Unit’s attributes.

|  |  |  |  |
| --- | --- | --- | --- |
| **Fort Attributes** | | | |
| **Name** | **Description** | **Type** | **Acceptable Values** |
| fortSize | The fort will be made off Ascii characters. For example, if this size is set to 100 then 100 characters will make up the width and 100 will make up the height, making the entire map have 10,000 tiles. (Small) | Int | 500 - 5000 |
| fortHostility | The higher this value is the more baddies will spawn. | Int | 1 - 10 |
| lootPercentage | The percentage of game tiles that start off with loot on them. Loot cannot spawn on a ground tile that is not passable, like a ground tile containing a wall. Therefore, this percentage can never be 100% | float | 0 - 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Ground Tile (Game Object) (2d Sprite Square)** | | | |
| **Name** | **Description** | **Type** | **Acceptable Values** |
| groundTileSprite | This holds the sprite from the tile set. | Sprite | Sprite will be hold in a dictionary. |
| playerHasSeen | This will be used when displaying the map. If a player has seen the tile it will show up on the map. | Bool | T, F |
| itemsOnFloor | This is an array of Items of length 3. Only three items will fit in on the same ground tile. If the tile is empty. | Item [3] | Any item |
| floorMaterial | This is so the correct sprite and description get shown. | FloorMaterial | See Class |
| tileIsEmpty | If the ground tile can hold items or people, it is considered empty. If it has a wall, or closed door it is not. |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Floor Material Class** | | | |
| **Name** | **Description** | **Type** | **Acceptable Values** |
| nameOfMaterial | [‘vertical wall’, ‘horizontal wall’, ‘grass’, …] | string | TBA |
| materialDescription | Used when the player wants a description of the ground tile. | string | ?? |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit Attributes** | | | |
| **Name** | **Description** | **Type** | **Acceptable Values** |
| Name | Name of unit | String | 15 chars |
| Class | The player will pick the unit’s class and it will change the hidden variables and dictate what items the unit spawns with. [Civilian, Farmer, Soldier, Need More…] | Enum | TBA |
| **Hidden Unit Attributes** | | | |
| health | This holds the unit’s current health. | int | 0 - MaxHealth |
| maxHealth | This holds max health. A soldier will have a high max. | int | 80 - 120 |
| visionDistance | The radius of the circle of vision the unit can see. | Int | 5 - 20 |
| speed | This will determine how much time passes when the unit takes an action. It will be reduced if the unit is carrying too many items, crouching, or injured. | float | 0.1 - 1 |
| hunger | Unit needs to consume food to keep this up, or starve. | int | 0 - 100 |
| thirst | Unit needs to consume food/ drink to keep this up, of die. | int | 0 - 100 |
| bit | If the player gets bit, they will turn into a zombie. This will take some time though and then can still escape before that. | bool | T, F |
| infected | This means the unit is now a zombie. Getting bit will lead to this. | bool | T, F |
| head | Body Part. | BodyPart | See Class |
| torso | Body Part. | BodyPart | See Class |
| armRight | Body Part. | BodyPart | See Class |
| armLeft | Body Part. | BodyPart | See Class |
| legRight | Body Part. | BodyPart | See Class |
| legLeft | Body Part. | BodyPart | See Class |

|  |  |  |  |
| --- | --- | --- | --- |
| **Class: Body Part** | | | |
| **Name** | **Description** | **Type** | **Acceptable Values** |
| Name | Name of body part. | String | (Head, Torso, ArmRight, ArmLeft, LegRight, LegLeft) |
| clothing | A ‘slot’ to hold the clothing item. | Game Object (Unless Items are a separate class) | Item: Clothing |
| backpack (Torso Only) | A ‘slot’ to hold the backpack. | Backpack |  |
| Item (Arms Only) | A ‘slot’ to hold any item. | Game Object (Unless Items are a separate class) | Item: Any |

|  |  |  |  |
| --- | --- | --- | --- |
| **Items** | | | |
| **Name** | **Description** | **Type** | **Acceptable Values** |
| Name | Name of Item | String | 15 chars |
| Condition | The condition of the item. | Float | 0 - 1 |
| Description | Short Description of the item. | String | ?? |
| **Sub-Class: Weapon** | | | |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| **Sub-Class: Clothing** | | | |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| **Sub-Class: Food** | | | |
|  |  |  |  |
|  |  |  |  |
| **Sub – Class: Door** | | | |
| doorIsOpen | This will help when deciding where to display and open door or closed door and whether the ground tile is empty. | bool | T, F |
|  |  |  |  |
| **Sub-Class: Misc** | | | |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Items** | | | |
| **Weapons** | Rifle |  |  |
| Pocket Knife |  |  |  |
| Chef’s Knife |  |  |  |
| Hunting Knife |  |  |  |
| Pistol |  |  |  |
| Shotgun |  |  |  |

# Main Scene

The main game play will consist of the camera staying right on top of the player and looking down as they move around. Every time the player completes an action, like moves, or shoots, or even drinks time will advance. If the player stands still and does nothing time is frozen. The player’s main motivation will be to find an exit out of the dungeon. The movement will be a lot like Adventure Mode in DF.

Text

Description automatically generated

## General Movement Keys:

The player should be able move using arrow keys, interact with things around him, and bring up the main menu.

Text

Description automatically generated with medium confidence

|  |  |  |
| --- | --- | --- |
| **Basic Movement** | | |
| Movement | Description | Time |
| Up/ Down/ Left/ Right | Arrow keys to move one tile. | 1 |
| Crouch/ Stand | Does | 0 |
|  |  |  |

## Basic Controls:

During the main game play the player can move the unit using arrows keys, bring the Interact Menu up the A and Main Menu up with B. With the interact button up the player will choose a selection using the arrow keys and A or use B to go back. Once a selection is made the interaction will either happen, of the player will be in ‘Select’ mode and must choose the item he wished to interact with using the arrow keys and A. Select mode will be identified because an ‘x’ will appear where the unit is, the indicates to the player that the arrow keys now move the x and not the unit. For example, a player might push A and the interact menu pops up, then select ‘Look At’ and the ‘x’ pops up indicating we are now in select mode, the player might then move the cursor using the arrow keys until it is over the item, they are curious about. Once the player hits A and makes the selection the feedback panel will write to screen the description of the item.

## Main Menu:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Inventory | Map | Legend | Settings/ Save/ Quit | Log |
| * Inventory Screen: Will display the items the player has on his body and inside his backpack. They will also be able to equip/ unequip items and see the item’s name, description, weight, and condition. * Map: As the player moves around in the game, they will see tiles. Once a tile has been seen a bool on the tile will be ticked. Then the map the player sees will only consist of the previously seen tiles. * Legend: Breaks down the tile set so the player can quickly see what a symbol means. * Settings/ Save/ Quit: Self-explanatory. * Log: During the game the player will receive feedback based on their interactions. For example, they may grab and item and the feedback may be, ‘You picked up an apple’. This will be entered in the log so the player can see this at any point. | | | | |

## Interactions:

The player can bring up the interaction menu by pressing ‘A’ during the main game play.

|  |  |  |
| --- | --- | --- |
| **Basic Interactions** | | |
| **Interaction** | **Description** | **Time** |
| Look At | If the tile is within the unit vision a description should be shown. | 0 |
| Crouch/ Stand | Changes movement speed. | 0 |
| Grab | Grab Items next to the unit and place in backpack. | 0 |
| Grab Item | When the player is standing on a tile all the items will have a button appear. This will encourage the player to stand on top of the item before grabbing it. | 0 |
| Attack | Attack range will depend on weapon. | Time should be taken from weapon. |
|  |  |  |
|  |  |  |
|  |  |  |