Will Liu

# RescueGame (Markus) Tech Spec Write-up

| Item/Task | Possible Issues | Design | Methods |
|---|---|---|---|
| Levels | The levels vary a lot in difficulty, such as the example comparing a cat rescue vs. saving multiple people from a hospital. This will require many different maps and a ton of hours to implement. | Each level is an object that contains two things: an array of sprites and serialized data of the image. The level object will be initialized on runtime. All objects will be stored under an overarching Difficulty class, which will have 3 arrays of levels for each (3 total) types of players. | Difficulty class:<br>whichLevel(xp_points, player)<br>-player is a string from a global array of types of players<br>loadLevel(level)<br><br>Level class (gets initiated by loadLevel())<br>deserializeMap()<br>loadSpritesOnScreen() |
| Players | The ambulance driver and rescue helicopter options might become difficult to implement because they are vehicles. This will greatly increase implementation time. | There will be a general super class for any type of players. This class will construct the subclasses for player type as specified and initializes the other traits such as speed and health based on experience points, which is a global variable set to 0 when the game launches. This superclass will also handle all movement, as detailed in the original descriptions. Movements will occur based on certain keypresses. Specific interactions besides talking will be specifically implemented in each subclass. | calculateTraits(xp_points)<br>- this sets the character traits such as health, stamina, and respect points<br><br>loadPlayer(player)<br>-constructs the specific type of player based on the string passed to it, which will be chosen from a global constant array of playable characters<br>-pass over the traits to the constructor<br><br>moveRight(), moveLeft(), moveUp(), moveDown(), jump(), speakToCivilian()<br>-for the last method, if you reputation points are high enough, you will earn some rescue points (based on some linear equation) or get some health back |

|  |  |  |  |
|---|---|---|---|
| Communication Banner | Updating real time might encounter speed issues. | Have a banner and have all the messages to display in a singly linked list as a queue. Each message if displayed remains on screen for 5 seconds. Show an enlarged sprite if a specific sprite is speaking. | displayMessage()<br>-dequeues a message and displays on screen for 5 seconds with the correct sprite |
| Fireman | Animating the water flow might become difficult. | Subclass of the player class. Will cause the fire class to construct. All animations will just be a list of movements. | -InteractionOne()<br>-sprays water, must program the movement of the water<br>-InteractionTwo()<br>-chops down breakable objects with an ax |
| Fire | Making fire appear dynamic might be difficult. | When touched, take away health from the player. Will have its own health variable that changes based on whether water touches it. | isTouched(), damage()<br>-If isTouched returns true, damage will take away from player health and stamina.<br><br>isWaterComing()<br>-for each second of water flow, lose health, if health is at 0, fire deconstructs and disappears<br>-considered true if any part of water sprite touches the fire |
| EMT | Animation with saving civilians might become difficult to implement. | Subclass of the player class. Will cause the injured_person class to construct. All animations will just be a list of movements. | -InteractionOne()<br>-Perform CPR. Program the movement accordingly. Takes away half of the player's stamina<br>-InteractionTwo()<br>-Some bandage and revival magic. Takes away half of the player's stamina and health. |

| | | | |
|---|---|---|---|
| Injured Person | All the animations showing the injured person's current state might become difficult to implement. | Have different states based on its own health variable from 1 to 100; by default, all members of this class start from a health variable of 1. | isHelped(type_of_help)<br>-if CPR, get plus 20 for health, else plus 50<br><br>currentState()<br>-state 1 if under 20, lying on the ground<br>-state 2 if under 70, sitting looking pale<br>-state 3 = fully revived, if over 90 |
| Police | Bullet mechanics might become difficult. Hostage rescue might become too similar to the EMT missions and would disrupt organization of the game. | Subclass of the player class. Will cause the criminal class to construct. All animations will just be a list of movements. | InteractionOne()<br>-arrest if criminal is less than 5 pixels away, else stumble and lose 10 stamina points<br><br>InteractionTwo()<br>-fire, lose 5 stamina points |
| Criminal | Animations for actual criminal activities might become difficult. | Have different states based on its own health variable from 1 to 100; by default, all members of this class start from a health variable of 100. | isShot()<br>-if bullet touches, loses 30 health points<br><br>isArrested()<br>-animate the criminal into a sitting position |
| Movable Sprites | No foreseeable difficulties. | Object that gets placed based on x and y coordinates. Players cannot walk through it unless they break it. | initialize(x, y)<br><br>isHit()<br>Returns True if touched by weapon like ax or bullet. Will disappear afterwards. |
| Static Sprites | No foreseeable difficulties. | Simple static object that gets placed based on x and y coordinates. Players cannot walk through it. | initialize(x, y) |

| | | | |
|---|---|---|---|
| Shop | Selection panel for buying. | A group of images. If a cursor clicks a certain part, upgrade that part if enough points. | canUpgrade(parts, points)<br>-upgrades the specific part to perform 5 more points of damage if enough reputation points are available, else let the player know they need more reputation points via a text box<br>-prices stored in a global array for this Shop class, which gets constructed on initialization |
| Town with HQs | Unfeasible for a canvas game. Will require too much time, unless a really limited town is created. | Perhaps it can be treated like the shop where events occur when you click on part of the picture. | |
| In-App Purchases | Will require out of Canvas libraries. Out of scope of the game in my opinion. | Perhaps some javascript on the same page will get this implemented, or redirect to another webpage. | redirect()<br>-redirects players via a new tab to the purchases web page |