Specification

Artificially Intelligent Game

The game when complete should be able to accurately measure inputs to the system in reaction to a stimuli provided within the game and using this information edit the fitness weighting of the stimuli and provide the information to the system so that it can better select a stimuli that will receive a reactional input from the user. As such it will be able to detect whether it is a good reaction or bad reaction and update the fitness accordingly.

Technical Structure Overview:

1. Artificial Intelligence Director:
    1.1. Setting Time before Fear Occurrence
        1.1.1. Allocate a fitness function to previously used timings.
        1.1.2. Multiply the fitness functions by the timings to create an average timing.
        1.1.3. Create a normal distribution around the average timing and choose a time based on the normal distribution.
        1.1.4. Send how much time has been selected to Fear Deployment
        1.1.5. Make sure timing between each fear does not happen quicker than 30 seconds or slower than 5 minutes.
    1.2. Fear Selection
        1.2.1. Allocate a fitness function to each fear created by the team.
        1.2.2. Have the fear selected by a metropolis algorithm.
        1.2.3. Send which fear has been selected to Fear Deployment
        1.2.4. Make sure timing between each fear is not selected 3 times in a row.
    1.3. Fear Deployment
        1.3.1. Make sure auditory fears are heard.
        1.3.2. Make sure visual fears spawn at different spawn points.
        1.3.3. Make sure enemies spawn at different spawn points and have their own set of behaviour. (For Example: How the enemy moves towards the player)
    1.4. Fear Potency Detection
        1.4.1. Take the distance at two intervals from the character mover and calculate a speed away from the stimulus.
        1.4.2. Take the rotation of view at two intervals and calculate the speed at which the player looks away from the stimulus.
    1.5. Fitness Recalculation
        1.5.1. Calculate the change in fitness based on the two speeds sent by the Fear Potency Detector and send it to the Time and Fear controllers to recalculate their fitness values.
2. Visual Asset Creation:
    2.1. Textures
        2.1.1. Textures must be fitting of the environment meaning it they must have the following qualities:
            2.1.1.1. Gloomy
            2.1.1.2. Desaturated
            2.1.1.3. High Resolution

2.2. Models
    2.2.1. Models must be used to make a realistic looking area to match the user's idea of a hospital and what is within.
    2.2.2. They must be less than 20,000 polygons.
    2.2.3. They must adhere to the texture specification.
3. Auditory Asset Creation
    3.1. Sound Effects
        3.1.1. Sound effects must be gained from a royalty free source.
        3.1.2. Sound effects must be fitting and edited to sound like they came from a hospital.
        3.1.3. Recorded voices and sound effects must be high quality.
        3.1.4. Acting quality must have at least some emotion put in it.
    3.2. Music
        3.2.1. Music must be gained from a royalty free source.
        3.2.2. Music must be fitting for a horror game.
        3.2.3. Recorded music must have a high audio quality.
        3.2.4. Music must be used to provide an emotive response and nothing else.
4. Complimentary Game Mechanics
    4.1. Character Movement
        4.1.1. A Character should be able to move in the following directions:
            4.1.1.1. Forward
            4.1.1.2. Backward
            4.1.1.3. Left
            4.1.1.4. Right
        4.1.2. A Character should be able to jump and land
        4.1.3. A Character should collide with walls and the floor as well as any collision-based objects.
        4.1.4. A Characters position should be tracked and when asked for, record the distance an object has moved away from a fear stimulus in order to provide new fitness data to the Fear Potency Detector.
        4.1.5. A Character should also be tracked to see if they encounter areas where a specific effect may happen. (Upon Entry Triggers for Sound)
    4.2. Interaction
        4.2.1. When the interaction button is pressed, whatever the crosshair at the centre of the screen is over will be interacted with
        4.2.2. This only works for objects at a 2-meter distance (In game meters).
        4.2.3. Interaction is required so that doors or objects may be used / opened as in the real world in order to as closely replicate this, this will be specified by a script tied to each interactable.
        4.2.4. If the object is not interactable, nothing will happen.
        4.2.5. This only works for the first object touched by the crosshair and not objects behind it.
    4.3. Survival Management
        4.3.1. UI Bars that display the amount of fuel and fatigue a character has.
        4.3.2. When the fuel hits 0, the lighter stops working.

4.3.3. When fatigue hits 0, the character falls asleep and reawakens at their bed. (Reloads the level)
4.3.4. Picking up specific items will refuel or remove fatigue

5. Menus
   5.1. Pre-game Menu
      5.1.1. Has an option to set screen resolution.
      5.1.2. Has an option to set graphical setting for lower end PCs.
      5.1.3. Has an option to rebind keys to allow a customised experience.
   5.2. In Game Main Menu
      5.2.1. Has an option to start the game.
      5.2.2. Has an option to quit the game.