SP4 Proposal

Roland Shum Weng Sang

Leow Wei Thou, Samuel

Chung Wei Qi

Chiu Ying Tzu

Amos Lee Jia Shen

# Game Idea 1

|  |  |  |
| --- | --- | --- |
| Name: | | Gas Mask Stall |
| Genre: | | 2.5D Casual Simulation |
| Platform: | | PC (C++) |
| Selling Point: | | Resource Allocation game (Mind game) |
| Game Description: | | It’s that time of the year again! The haze is back once more in Singapore, and you won’t want to let this chance go to waste. Set up a gas mask stall and start racking up profits! Live the dream and become the next billionaire tycoon in Singapore! |
| Game Play: | | The player starts with $1000 on the first day of the week. At the start of the day, an average forecast (PSI reading) which may not be accurate is shown to the player. The player will be in charge of three stores and has to buy however many gas masks he plans to sell on the day and set the selling price. After that, the player will go to the game screen where the simulation starts playing. Buyers affected by the haze will come forward to your stall to buy your masks. Whenever the game is paused, the player is allowed to alter prices and shop location based on the changing needs and haze locations. When the day ends, the total profit is shown to the player and all gas masks that aren’t sold are thrown. The next day starts and the process repeats. The game ends on the last day of the week where the PSI exceeds 400. |
| Features: (Work Load) |  | Game States🡪 Main Menu, settings  Textures  Win Lose condition  UI  Sound  Change to Orthogonal view  Money System (Buy/Sell/No. of mask)  Buyer Class  Haze Class  Forecast Class  Load/Save Class |
| Inspiration: | | https://www.msu.edu/~bistakwe/Lemonade%20Stand%20Game%20Pic2.jpg  Reference Image ^ |

Gameplay:

**Core:**

One player simulation.

Money system.

Haze and Weather Algorithms.

Forecast System

Time System.

**Good to Have:**

Bubble Feedback System.

Algorithms

Stock collection.

Buy during simulation

Walk to people convince them to buy.

Lua init and variables.

Encryption

Feature Breakdown:

|  |  |  |  |
| --- | --- | --- | --- |
| PLAYERS | FEATURE | DESCRIPTION | WORKLOAD |
| Roland | Highscore Class | Load and Save a list of highscores from text file |  |
| Roland | Buyer Class (Rendering algo, personality algo, waypoints) | Creating algorithm that dictates buyers attitude. | 9 |
| Roland | Weather (haze) Class (single weather) | Creating algorithm that randomizes weather density. | 8 |
| Ying Tzu | Textures | Creating and implementing textures | 3 |
| Ying Tzu | UI (Buttons, etc) | Creating game UI, button class and on click checking | 7 |
| Ying Tzu | Bar Class | Displays and update a timer bar for in game visual |  |
| Ying Tzu | Money Animation Class | Animation of money flying up when a buyer buys a mask |  |
| Ying Tzu | Money System  (Buy/Sell/No. of mask) | Implementing algorithm for money. | 4 |
| Ying Tzu | Stall Class | An entity child class that stores values for gas mask and price |  |
| Wei Qi | Forecast Class | Class that has random accuracy as to forecasting | 7 |
| Wei Qi | Load/Save Class | Load and Save from text file. | 5 |
| Wei Qi | Models | Models made in maya and UV mapping for Stalls |  |
| Wei Qi | Buying mask during gameplay | Able to select stalls to buy mask |  |
| Samuel | Game State Manager  (Framework) | Creating the base of game states and its manager class | 5 |
| Samuel | 3D Camera | Coding out camera and its functions | 5 |
| Samuel | Timer Class | Class with delta time and timers. | 1 |
| Samuel | Clicking and Stall Selection | Clicking on stalls to move them | 6 |
| Amos | Win Lose condition | Creating win/ lose conditions | 1 |
| Amos | Sound | Creating sound class to use. | 2 |
| Amos | Grid Map | A grid map for placement of shops. | 5 |
| Amos | Entity Class | Creates easy to manage entities | 3 |
|  | MenuState | The main menu |  |
|  | TutorialState | State that shows the tutorial images |  |
|  | PlayState | The *game state* with the *in-game state manager* and the list of *in-game states* |  |
|  | BuyMaskState | Player buys mask at the start of the day here |  |
|  | StartOfDayState | Player assign a number of gas mask to a stall and the price to sell |  |
|  | GamePlayState | The actual gameplay/simulation |  |
|  | EndOfDayState | Shows the profit earned and the forecast for the next day |  |
|  | SaveState | Saves the game progress |  |
|  | LoadState | Loads the game progress from a text file |  |
|  | EndGameState | When the game ends (reaches end of the week), checks WinLose condition and displays highscore |  |
|  |  |  |  |

**Good to Have:**

|  |  |  |  |
| --- | --- | --- | --- |
| PLAYERS | FEATURE | DESCRIPTION | WORKLOAD |
| Roland | Lua | Reading from Lua scripts for convenience | 6 |
| Samuel | Bubble generation algo. | Algorithm that randomly generates different speech text with different responses. | 6 |
|  | Competitors AI – Collaborative – Message board |  | 5 |
| Roland | Competitors AI – Genetic AI | AI algorithm that competes against the player | 10 |
|  | Losing customers rate algorithm. | Algorithm that changes the rate of losing customers. | 5 |
| Ying Tzu | Multiple Weathers. | Various weathers that affect gameplay | 5 |
|  | A star Path Finding. |  | 10 |
| Wei Qi | Negotiate Class | Code that simulates negotiating prices. | 7 |
|  | Stock Collection | Collecting various items that increase stock of masks | 7 |
| Roland | Encryption | Encrypting saved data to make editing of files harder. | 5 |

**Things Added in:**

* **Sound**
* **Finite state machine**
* **Save and Load**

**List of Sounds:**

* Background Music
* Clicking sfx
* Purchase sfx
* Money gain sfx
* Stock placement sfx
* Game Over sfx

**Sketches**





