## Characteristics

|  |  |
| --- | --- |
| **Description:** | A game where a player dodges simple enemy objects and raises a score. |
| **Actors:** | User |
| **Uses:** | 1. Entertainment and enjoyment |
| **Pre-conditions:** | 1. Game has been installed and/or java project downloaded |
| **Post-conditions:** | 1. Game successfully closes |
| **References:** | //git repo when committed |

**Process Flows**

|  |  |
| --- | --- |
| **Normal Flow:** | 1. The user opens the program. 2. The user is presented with a Menu with the following options:    1. **Play**       1. When selected, player is brought to beginning game screen,       2. Using W, A, S, D the user moves the white square up and down.          1. Cannot move character outside the window       3. Enemy objects spawn at given levels and bounce around the screen          1. Enemies are different based on level             1. Basic Enemy – Moves around the game at an average speed.             2. Fast Enemy – Velocity is higher, moves faster             3. Smart Enemy – Moves considerably slower but follows the player object.          2. Enemies cannot move outside of the window.       4. Enemies and Player have collision aspect          1. When Player object collides with Enemy object, HEALTH is decreased          2. When an Enemy object collides with a window boundary, the correlating velocities are reversed (i.e. if it hits a ceiling, the y velocity is reversed).       5. Score and Level are kept in a thread/ticker method that keeps track          1. Level starts at 1             1. Level cannot be less than 1;             2. Level rises by one for each 250 score count.          2. Enemies spawn at certain levels             1. Fast Enemies – spawn on levels divisible by 3.             2. Smart Enemies – spawn on levels divisible by 5.             3. Basic Enemies – spawn on all other levels   Cannot spawn on levels divisible by 3 or 5.   * 1. **Help**      1. The user is brought to a screen where instructions are given on how to play the game.      2. The user is given the option to return back to the Menu.   2. **Quit**      1. The player clicks this to end terminate the program (can also hit Escape)  1. The Player’s HEALTH eventually falls to 0    1. HEALTH cannot fall below zero or be greater than 100.    2. HEALTH changes the green RGB level as health decreased, changing to a red as the number gets lower. 2. **END GAME SCREEN**    1. All objects are removed from the window, including Player object and Enemy Objects.       1. No other objects can exist.    2. User is presented with “Game Over”       1. User is presented with their Final Score       2. An option button “Try Again” is present. 3. Try Again    1. When Selected, the user is brought back to 2.1       1. Level and Score counters are set back to default values (1, 0 respectively)       2. Program checks to make sure there are no objects, removing any if presence. |