Space Battle

I. How to Play:

In this game, the player controls a small, blue spaceship, located on the left side of the screen, and the computer controls a red spaceship, located on the right side of the screen.

The player may move the blue ship up and down with the mouse, avoiding the red missiles the enemy fires. The player may fire missiles of its own by clicking the left mouse button on the screen.

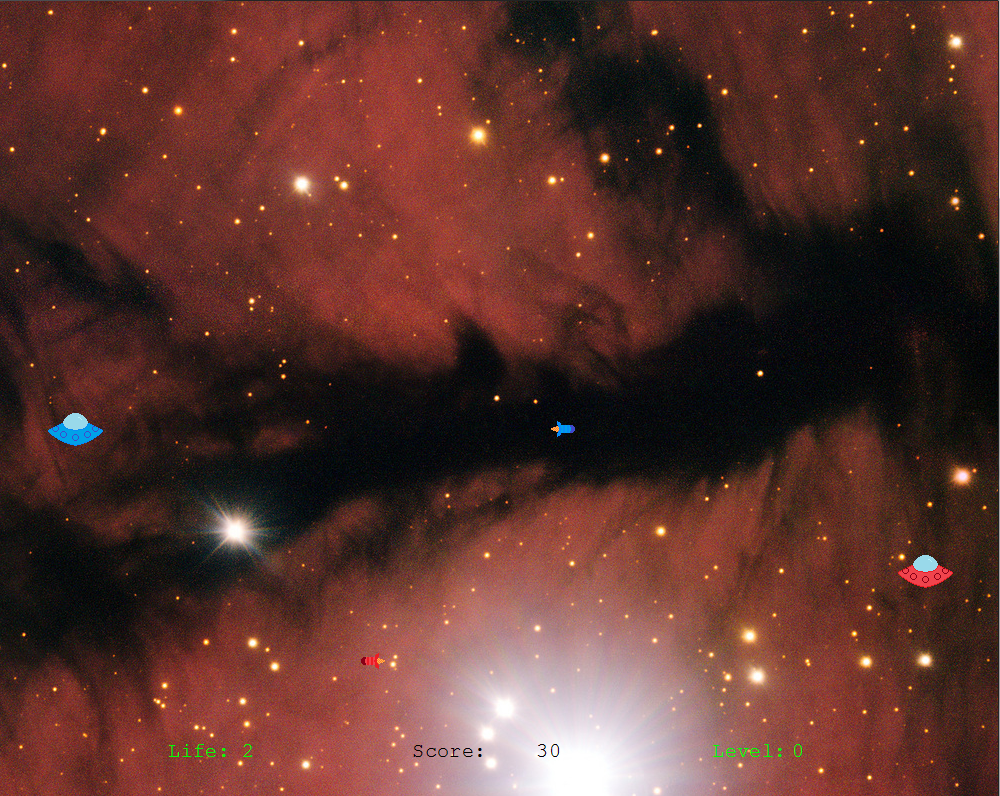


Fig 1: Space Battle

The more the player's missiles hit the enemy, the faster the enemy becomes. The object of the game is for the player to hit the enemy as many times as possible before the enemy lands five hits.

After the game ends, clicking on the restart button will restart the game.



Fig 2: The game in a "Game Over" state.

II. Installation:

Place spaceBattle.sl, enemy.png, hero.png, eShot.png, pShot.png, and background.jpg in the same folder as the easel game files. (Note: Background courtesy of space.com's free wallpaper section.)

III. Technical Specification:

III.i. Data Model:

A ***direction*** is 1 or -1, denoting up and down, respectively.

A ***shot*** is a point, representing the center of a shot on the screen.

The "move zone" is defined as the area of the screen between and including the horizontal line through (0,770) and the horizontal line through (0,30).

A ***pixel distance*** is a natural number, indicating the number of pixels the enemy may travel before turning around. This number may only be exceeded if it is not a multiple of the enemy's current speed.

An ***animation*** is a list of sprites, which, when displayed in succession, depict some animation.

A ***frame counter*** is a natural number.

A ***playback*** is a pair (p, L), where L is an animation and p is a frame counter in the range [0, size(L)). Playback (p, L) denotes an animation that is currently on its (p+1)'th frame, called its "current frame."

III.ii The State:

A ***State*** is an 8-tuple (gOver, pStat, eStat, shotTimer, pShots, eShots, score, playList), where

the following are true:

1. gOver is a natural number in the range [0,20]. Its initial value is 0.
   1. A gOver of 0 means that the game is not over. is in its gOver'th frame.
   2. A gOver of 20 means the game is over.
   3. A gOver in the range [1,19] means that the game over animation is in its gOver'th frame.
2. pStat is a triple (pVert, pLife, cooldown) where the following are true:
   1. pVert is the hero's y-coordinate. Its initial value is 500.
   2. pLife is the remaining number of hits the hero may take before the game ends. Its inital value is 5.
   3. cooldown is the remaining number of frames before the hero may next fire a shot. Its initial value is 0.
3. eStat is a 4-tuple (eVert, eDirec, eDist) where the following are true:
   1. eVert is the enemy's y-coordinate. Its initial value is 500.
   2. eDirec is the direction in which the enemy will next move. Its initial value is 1.
   3. If eSpeed is the enemy's current pixel speed, and traveling eDist pixels in eDirec would not take the enemy more than eSpeed pixels out of the move zone, then eDist is the minimum number of pixels the enemy may travel before changing direction. Its initial value is 120. Otherwise, eDist is arbitrary.
4. shotTimer is the number of frames before the enemy will fire another shot. Its initial value is 160.
5. pShots is the current list of player shots. Its initial value is [].
6. eShots is the current list of enemy shots. Its initial value is [].
7. score is the number of points the player has accumulated. Its initial value is 0.
8. playList is the list of playbacks to be animated on the screen. Its initial value is [].

III.iii Rules:

Movement:

The player may move the blue ship up and down so long as the ship's center remains in the move zone. The move zone, as mentioned above, is the area of the screen starting thirty pixels above the bottom of the screen and ending thirty pixels from the top of the screen. In other words, the ship will always be entirely on the screen.

The blue ship is limited to a maximum speed of ten pixels per frame (200 pixels per second,) but it may be moved more slowly.

The enemy ship will start by moving upward. Whenever the enemy ship reaches (or passes) the upper edge of the move zone, or when its pixel distance is less than or equal to zero, it will reverse direction. When it reverses direction, its pixel distance is set to a pseudo-random, natural number in the range [200,1000].

The enemy starts out moving at a rate of ten pixels per frame. For every level the enemy gains, its speed will increase by two pixels per frame.

Combat:

The player may fire missiles at the enemy. When a missile is fired, it is created with its center 40 pixels to the right of the center of the player's ship, then travels to the right at a rate of 32 pixels per frame. Every time the player fires a missile, he/she must wait one second before firing another missile.

The enemy fires missiles at regular intervals, firing more rapidly as its level increases. When an enemy missile is fired, it appears with its center 40 pixels to the left of the enemy's center.

The enemy fires its first missile 160 frames (eight seconds) after the game begins. Each time the enemy fires a missile, the shot timer is set to a pseudo-random, natural number in the range [30-3\*L, 60-7\*L], where L is the enemy's current level.

Once the timer goes off, the next shot is fired, and the process repeats.

A missile "hits" its shooter's opponent if the missile's center comes within 37 pixels of the opponent's center (when it is touching/nearly touching the ship.) Whenever a missile hits its shooter's opponent, it is removed from the game, and an explosion animation is played where it hit.

The player loses one life every time the player's ship is hit by an enemy missile. The player starts with five life.

The player scores 10 points every time the enemy is hit by a player missile.

Scoring and Levels:

The player's score starts at 0. As described above, the player scores ten points every time he/she hits the enemy with a missile.

The enemy's "level" starts at 0. For every 50 points the player gains, the enemy's level will increase by 1, to a maximum of 6. The level determines the enemy's movement speed and rate of fire, as described above.

Ending the Game:

For 20 frames after the player's life hits zero (in other words, after it has been hit five times,) the player will no longer be able to control anything on the screen. During this period, the enemy and any shots on the screen will continue to act as normal, but no more points may be scored.

After the 20 frames, the game ends. The player may then choose to restart the game.

Controls:

The player moves the ship by moving the cursor up and down. The ship will move vertically until it is level with the cursor, or until the edge of the "move zone" blocks its path. So long as the cursor is vertically at least 10 pixels away from the ship, the ship will move at its maximum rate of 10 pixels per frame.

If the ship and the cursor are vertically less than ten pixels from each other, the ship will instantly move to be level with the cursor.

The player may fire missiles by clicking the left mouse button. As mentioned above, a missile will only be fired if a missile was not fired in the past 20 frames (1 second.)

When the game ends, a 20-frame "game over" animation will be displayed, followed by a "game over" message and a restart box. Clicking anywhere within the restart box will restart the game, returning it to its initial state.