README

Dino Run (Chrome Dinosaur Game)
Samantha Kolasa, Chloe Lok Yee Yeo
Project TA: Ze

Intro

When the game begins, the Dino starts running and three types of obstacles randomly appear moving from right to left: a flying bat, a shrub, or a rock. The Dino must duck to evade the bat and jump to evade the shrub or rock. If the Dino does not evade an obstacle, the game is over.

Project Approval

Our project was approved by Ze.

Instructions

To start the game, the user presses the [s] key. When the game starts, the [down arrow] key is used to duck by holding it down and the [spacebar] is used to jump by pressing it once. Once the user hits an object and loses the round, they are prompted to put in their name which will pull up a leaderboard of top scores. To play again and try to get a better score, the user can press [s] to restart the game. If the user wishes to clear the leaderboard and play again, they can push the [flag] button.

Features

1. DINOSAUR MOVEMENT - this feature causes the dinosaur to appear “running” and move appropriately when the duck or jump keys are pressed.
2. OBSTACLE MOVEMENT - this feature generates random obstacles to be routinely sent towards the Dino’s position. this feature is also responsible for checking if the Dino correctly evades an obstacle and stops the game if the required evasion circumstances are not met.
3. LEADERBOARD - this feature keeps a record of a user’s local score during each attempt at the game. If the score is greater than or equal to the top 3 highest scores (on the leaderboard presented at the end game screen), their name and score will be added to the leaderboard appropriately.

Justification for Complexity

- The Chrome Dinosaur Game was given as an example of an appropriately complex project on the Final Project Proposal Spec. The game relies heavily on if two objects are touching, but we were able to create a “sensing” system without using is-touching blocks. We utilized features of Snap! not taught in class like “broadcast”, “send”, and “receive” in order to communicate between sprites and have gameplay actions occur concurrently, implementing concurrency and Object Oriented Programming [OOP].
- Our leaderboard combines local and global variables to keep track of individual game scores and compare them to past saved scores. A global variable containing a list was necessary to retain previous scores and display only the top three sorted by score value.

We implemented a sorting leaderboard using a table recording the name of the player as well.

<u>Lists & Script Variables</u>

- Script Variable: score
    - The value of this variable increases based on the amount of time the user stays alive after [s] is pushed to start the game. We use this variable to track the individual gameplay score and the score resets if the user chooses to replay the game (by pushing [s] on the game over screen). The score is then used in the leaderboard list.
- List: top 3 (leaderboard)
    - This is a list of the highest three scores of the game and the player's name. It is sorted in descending order with the highest scores displayed in the top positions. After each game concludes, the local score is compared with the existing scores on the leaderboard. If the local score is greater than or equal to any preexisting scores on the leaderboard, the score is inserted at the appropriate position in the list. If the local score is added to any position, the last (fourth) score is removed from the leaderboard, keeping only the three top positions in the table.
    - If the player's local score is less than the top three scores on the leaderboard, the score is not saved and the leaderboard is shown unchanged.
    - Pushing the green flag to restart the game instead of pushing [s] as directed will clear the leaderboard and all saved scores.

Function Table

Block / Function Name	**Domain (inputs)**	**Range (outputs)**	**Behavior (role in the context of the project)**
check high score + name + top 3	tempScore (number), name (any type), tempTop3 (list)	tempTop3 (list)	compares existing top3 score list with local score value. if the local score is higher than a value on the existing top3, the leaderboard list is changed and the new version is reported.
test score card with score + name + top 3 expecting output	testscore (number), testname (any type), testTop3 (list), testLeaderboard (list)	boolean	tests leaderboard cases to make sure the correct scores are shown based on input test values and expected outcomes. boolean reportsTrue if tests were passed.
count score	N/A	N/A	abstracted block for process of local score value increasing and checking/reporting leaderboard values
check touching bat	N/A	N/A	checks proximity of obstacles and dinosaur and dinosaur behavior. broadcasts end game if conditions are not successful.
check touching shrub/rock	N/A	N/A	checks proximity of obstacles and dinosaur and dinosaur behavior. broadcasts end game if conditions are not successful.
obstacles appear	N/A	N/A	on a repeating interval, broadcasts a message to send a random obstacle across the screen

obstacle move across screen	N/A	N/A	causes obstacles to move from right to left across screen with constant y value.
dino jump	N/A	N/A	causes dino to “jump”
dino duck	N/A	N/A	causes dino to change to “duck” costume when ↓ key pressed
dino run	N/A	N/A	causes costume changes to appear like dino is “running”
bat wing movement	N/A	N/A	causes costime changes to appear like bat is “flying”

Extra Credit: Additional features (3+ for groups of 1-2, 4+ for groups of 3) aka "Really involved project"

In addition to our two basic functions of movement/ obstacles and ending the game, we implemented an additional third scorekeeping and leaderboard feature. Our non-trivial list could have been only for scorekeeping history by just adding the local score to the end of a basic list. Instead, we created a top-3 leaderboard table that sorts scores in descending order and displays only the top three scores, automatically adjusting to include the local variable after each round if the value is higher than or equal to any preexisting scores in the top 3.