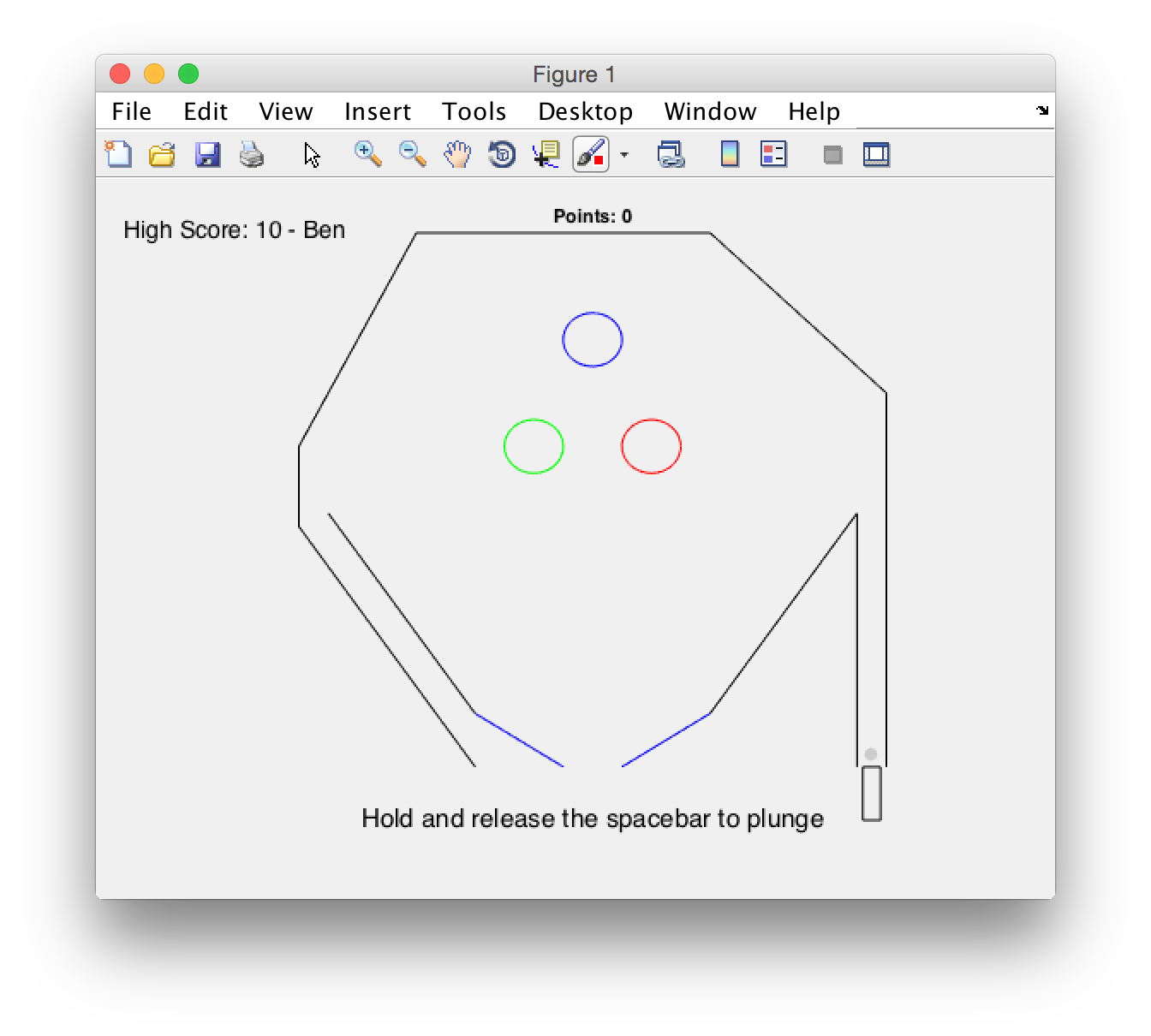
Pinball’s Return

Ben Bernstein



The goal of this project is to make an entertaining game of pinball using MATLAB. This could ideally replace the Pinball game that was removed from Windows Vista and later. The game should be fun to play as well as challenging.

At its most basic level, this game involves tracking ball as it moves around a predefined board. There is some downward acceleration pulling the ball towards the bottom of the field, so the kinematics of the ball are calculated repeatedly. During each repetition, the system checks for collisions between the ball and game pieces and factors them into the motion of the ball. Modeling the velocity of the ball and slopes of game pieces as linear equations, then solving the system, determines collision points. The angle of the resulting velocity is determined using the incoming velocity and the angle of the wall. Any coefficients of restitution or additional velocity imparted by moving flippers are considered as well.

In order to start the game, a plunger is activated by holding down the space bar. This activates a counter that determines the “depth” of the plunger and imparts a certain initial upward velocity. At the end of the game, the score is compared with the high score. If it is greater, the user is prompted to enter their name. To keep track of high scores, a .mat file stores the highest result and name. This is updated after the game ends if a new high score is reached.

Along with determining the correct angle and magnitude of velocities after collisions, I think the most challenging aspect of this game was modeling collisions with moving flippers. Because both the ball and the flipper move incrementally, not continuously, opportunities for errors became apparent. The ball would occasionally appear to move through the flippers because the “time to collision” calculated did not account for the flipper moving into the ball’s path. I have mostly negated this problem by running through two collision checks each repetition. The first occurs before the flippers are moved and the second occurs after. If the velocity changes after the first check, the first returned ballState is used. Otherwise, the second is used.

The major piece from the project proposal that is not included is an options panel. While this could be added, I think it would make the program too more complicated. Additionally, the user may decide to set conditions that are unrealistic, making the game less fun. With predefined initial conditions, the speed and behavior of the game is optimal.