# The story of Boop the Snoot

The Boop the Snoot game is a two-person game where you have to boop (tap) the nose of animals. Each one only shows up for a short while and then moves to a new spot.

# The game code

The game code has three basic parts: Initialization, the **Frame** per-frame function which changes the opacity of each **face** on the screen and possibly moves them, and the **OnTap** function which is called each time the screen is tapped.

The **faces** array is the current set of tappable faces on the screen. It’s set up once per level in the *SetupFaces* function which takes the overall graphics (g), the animals and faces array, and the current level (1, 2, 3, etc.). A single “face” is just a **Text** object (make with g.Text() given an area on the screen to be displayed) where the **Data** value is the number of points for the face.

The **animal** array is a pretty trivial array of strings; each string is just a Unicode animal-face character. There are copied from the Unicode tables in Best Calculator.

The **OnTap** function takes in a tap position and the “g” graphics area for the tap. It sets the opacity of the blue **Tap** object to 1 (it will be decremented in the **Frame** function so that it fades out neatly) and checks to see if the tap overlaps any visible faces. If a face is tapped, the corresponding player score is incremented by the amount in the Data field (that way, we can make for more interesting scores if we want)

# Timeline for making the game

Following is the simple system for tracking the coding for the game. The table started with just the first entry which got the game to a point where there was “something” on the screen. After that, new lines were added to the table as needed. Each item in the table was done in order (but that’s not a hard-and-fast rule).

|  |  |
| --- | --- |
| Not tracked | Started game code; created animal and |
| 8:40-9:33  53m | Create two different scores and score texts. |
| 9:34—9:39  5m | Taps will set the correct score |
| 10:01-10:31 30m | When one player get all their animals, do a fanfare and give them more points. Freeze the game so that the other player can’t tap. Do this with a state value  State r=running g=goal animation w=win animation |
| 8:28-9:58  90m | Have one player actually win (how many rounds do we play to? Just 5? Or more?). Store this in WINLEVEL. Make DIM ScoreLevel(2) to count the number of levels each has won. Update code to use more arrays! And make a little trophy case (ScoreTrophy) so everyone knows how many levels each side has won. Do the win animation when one side wins ADDED: can get the .Text value of a g.Text () object – that lets me set up the trophy.  Was heavily distracted b the dog 😊 |
| 10:33-11:00  27m | Let the players start a new tap – check for a Tap while state is “w” and do a WinAnimationEnd() and reset the game. (ResetScores() + ResetTrophies())  Bug: after finishing a level, reset the scores back to zero. |
|  |  |