Bloons Tower Defense Lite

Coded by: Jocelyn Chin, Kofi Nightingale, Razaq Aribidesi

Instructions

Bloons Tower Defense (Lite) is a simple game where the user is tasked with placing different towers down along a path strategically to pop incoming waves of balloons that would, in theory, become progressively more difficult. To score points, the user pops balloons with towers. Keep the balloons from hitting the end of the path or the health bar will decrease. When the health bar fully decreases, the game is over.

Upon running the code, there are a few instructions and/or guidelines that the user must strictly follow to have the most seamless experience.

1. There are **three towers** available to be placed. All have an ammunition capacity of **20** bullets and all shoot at the same speed with the same bullet types. The only difference is their direction of attack
   Tower #1 (pink missile) → Shoots directly upwards
   Tower #2 (superhero monkey) → Shoots directly downwards
   Tower #3 (blue contraption) → Also shoots directly upwards
2. **Only one tower can be placed at a time**. You are not allowed to place more than one tower down at the same time. Additionally, the user can only place another tower once one tower is completely finished shooting. **Knowing this, it is important to think, time and place strategically to maximise your chances of clearing a wave.**
3. Finally, watch the magic happen as you pop balloons! Have fun with it!

The score will go up the more balloons you pop. Try each time to get a high score by clearing out the wave!

There is also a banana counter present in the top right corner of the game. This is intended to be a form of currency that one could use to purchase more towers of different strengths. Placing a tower will decrease the banana amount by 2000 and the bananas linearly increase the longer the player stays in the game.

Summary

The intent of this project is to recreate a lite version of the established game franchise Bloons Tower Defense (specifically attempting to mimic later versions of the game Bloons TD 4 and onwards). The program utilizes numerous aspects of coding in MATLAB learnt this semester and seeks to build upon some of them.

The high-level explanation of this algorithm is that the basic user interface of the map, towers, health bar, point counter, and banana counter are displayed using axes and image based code on a singular figure. Using timers, the balloons are incrementally moved upon a predetermined path with given timesteps and towers are dragged and placed using callback functions. Bullets are shot from the towers using timers to iterate a predetermined amount of ammunition across a screen for a predetermined range. Finally, using a series of handle data and if then statements, the score is increased when a balloon is “popped’ by a bullet (i.e. if it is in the same vicinity) and if a balloon reaches the end, the health bar decreases (damage increases). If the health bar runs out, then the game stops and the user ends with a certain amount of points.

Firstly, the map is loaded using a combination of user interface concepts and also image processing. Taking an established map from the Bloons Tower Defense game, it is loaded onto the map with the Image Processing Toolbox. Then, using a series of commands to alter the figure and corresponding axes, it is formatted and stacked so multiple axes sit on one another.

Secondly, GUI data was utilized throughout this whole game to store and pass a variety of objects such as the score(points), bananas, balloons, and bullets. Balloons and towers were created using more AlphaData from image processing and image X and Y data was utilized to move the images.

To move the balloons, this is encoded in a moveballoon function. Following a series of pre-determined timesteps, each individual balloon image changes its path based on where it is along the path. The timestep is encoded with a handles variable and increments only after the balloon has moved past X == 0 or the beginning of the map.

Finally, towers are placed via basic drag and drop mouse callback functions. The place function is more fleshed out which contains the necessary code that determines whether or not the towers can be placed (based on the number of bananas the user has in possession) and the code that starts the timer objects responsible for the shooting of the bullets. These are organized and contained in different nested functions within the callback function for the release of the mouse button.

To execute both the movement of the balloons and the shooting of the towers simultaneously, timers are utilized to run both functions in conjunction with one another. It was found that creating multiple timers for multiple balloons resulted in intensive memory and CPU usage. Thus, although other balloons were encoded and loaded into the script, this game only currently has one balloon type moving. Due to the limitations of matlab, if there is a slight lag on the computer, this will result in the slower movement of balloons or bullets not being deleted.[1]

Other features throughout the code include a home screen with callback functions for buttons to start the game and to access a help menu. Counters iterate along with a timer and use logical statements encoded with the position of the balloon.

Note: no outside example code was used for the development of this project.

Utilized ToolBoxes

- Image Processing Toolbox

[1] In some instances, the bullets are not deleted and continuously delete the balloons. However, after many tests of the individual code, we have attributed this error to MATLAB timer malfunctioning.