Term Project Proposal**:**

Name: Catherine Bernaciak

Andrew ID: cbernaci

August 13, 2021

# *TP3 Update*

Updates were made to the Gameplay section below where specific difficulty levels were added for rounds 1-5 of gameplay. Also, an auto-play option was added. Levels 2-3 were not implemented so were removed from Gamplay section. UI section reduced to eliminate features not implemented.

# Project Description

This term project is an attempt to recreate the popular Atari XE game, Q\*bert1. Q\*bert is a creature that lives in a pyramid constructed of 28 cubes. Starting at the top cube with each game play, Q\*bert must hop on the top of each cube in the pyramid to change its color and receive points all while avoiding a point-deducting collision with a several different enemy creatures hopping along the board. To complete a round, Q\*bert must change all rectangles a required number of times, which depends on level. The game progresses to a new level after Q\*bert has completed a certain number of rounds. Movement for both Q\*bert and enemy creatures is constrained to nearest neighbor cubes above or below the current cube. Movement strictly left-to-right is not part of gameplay.

|  |
| --- |
|  |
| Figure 1. Example of Q\*bert gameplay |

# Structural Plan

The game will require the following elements:

**Board** – The board is a pyramid of stacked cubes. The only relevant part of the cube is the top surface. A dict will be used to map the (row, col) location of each cube top (the active part of the game board) to the (x, y) coordinate of the center of the cube top.

**Gameplay** – There are 5 rounds per level, with difficulty increasing with each round. There is always only one snake that follows Qbert, but the number of red and green enemies, how fast they move, and how the green enemy undoes Q\*berts squares changes as follows:

***Level 1:*** one red enemy, one green enemy, slowest speed,

***Round 1 difficulty:*** green enemy undoes only the corner squares

***Round 2 difficulty:*** green enemy undoes only the bottom row

***Round 3 difficulty:*** green enemy undoes left edge and right edge

***Round 4 difficulty:*** green enemy undoes left edge, right edge, and bottom row

***Round 5 difficulty:*** green enemy undoes all but the top square

**Game pieces**: Q\*bert and enemy creatures will be part of a class given that their movements are all constrained to movement between adjacent cubes top-to-bottom.

* class Player(object)
  + Q\*bert and enemy will inherit from player
  + class qbert(player)
  + class enemyCreature(player)
  + there includes a method for movement that provides curvilinear motion (not teleportation) for all game pieces
  + there will be attributes for cube tops that have been touched
  + there are several functions related to choosing the next move, whether it be randomly, to follow Q\*bert, or for autoplay. All are contained within the Player class object.

**Movement -** Movements of all game participants (Q\*bert and enemies) will only be between adjacent cubes from top-to-bottom or bottom-to-top. In other words, no lateral movement directly left-to-right or right-to-left is allowed.

**Physics of movement** – The motion of jumping between cubes, for all game pieces, is smooth, and not simply just teleportation from one cube top to another. The movement has to happen over several clock ticks and follow a curvelinear trajectory.

**Moving Q\*bert** – Q\*bert’s movement is controlled by the 4 keys ‘u’, ‘i’, ‘h’, ‘j’ and will be driven by the def keyPressed(app) function.

* ‘u’ 🡪 up and to the left
* ‘i’ 🡪 up and to the right
* ‘h’ 🡪 down and to the left
* ‘j’ 🡪 down and to the right

**Moving of enemy creatures –** there are 3 types of enemy creatures in the original game, snakes, red, and green blobs. They move in different patterns as follows:

* Snake (named ‘Coily’) – there is only one on the board at a time. In the original game it’s represented as a coil that springs when landing on the top of a cube. Here, it’s a purple oval. It follows Q\*bert along the board, so it’s movement will mirror Q\*berts. It moves at a constant rate always trying to get closer and closer to Q\*bert. Q\*bert can get rid of Coily by luring him off the board to jump onto a hoverboard that sits on either side of pyramid (more on that below). If Coily is within one cube of Q\*bert he will jump off the board after him into the void.
* Green Blob (names ‘Slick’ and ‘Sam’) – there is just one of Slick or Sam on the board in level 1. There would be more in higher levels but those are not implemented. These move randomly but when they encounter a cube Q\*bert has changed the color of, they change it back to the original color. Q\*bert can get rid of them by jumping onto a cube they are currently on. Q\*bert loses 10 points if it jumps on him. Otherwise, they are harmless to Q\*bert.
* Red Blobs – (names ‘Ugg’ and ‘Wrongway’) – there is just one of these on the board in level 1 (would be more in higher levels). They also move randomly over the board. If Q\*bert touches one, he gains +10 points, it disappears and a new one appears at random. If one touches him, he loses 10 points. Their speed would increase with each level.

**UI/Screens** – There are 2 different modes/screens:

* Instruction screen – The instructions for how to play are posted here
* Game play screen – the screen where the game is played

# Algorithmic Plan

Q\*berts autoplay proceeds as follows: at each location, the path with the highest number of untouched cubes is determined, Q\*bert takes a step in that direction. If there is a tie, the move is at random.

# Version Control Plan:

All work is backed up to my github account:

