Term Project 1 Deliverable**:**

Name: Catherine Bernaciak

Andrew ID: cbernaci

August 8, 2021

# 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 2d list will be used to keep track of the top surface of each cube using a (row, col) tuple. In general, rows run top-to-bottom 0 to 6 and cols run right-to-left 0 to 6. The top cube is (row 0, col 0), the cube down and to the left is (row 1, col 1), the cube down and to the right is (row 1, col 0), etc. Player and creature movements will only be between adjacent cubes, and only top-to-bottom or vice versa movement is allowed.

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

* class player(object)
  + Q\*bert and enemy will inherit from player
  + class qbert(player)
  + class enemyCreature(player)

**Moving Q\*bert** – Q\*bert’s movement is controlled by the arrow keys and will be driven therefore by the def keyPressed(app) function.

* ‘left’ 🡪 (row-1, col+1) moves to the cube up one row and left one col
* ‘right’ 🡪 (row+1, col-1) moves to the cube down one row and right one col
* ‘up’ 🡪 (row-1, col-1) moves to the cube up one row and right one col
* ‘down’ 🡪 (row+1, col+1) moves to the cube down one row and left one col

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

* Snake (named ‘Coily’) – there is only one on the board at a time. It’s represented as a coil that springs when landing on the top of a cube. 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 (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 Blobs (names ‘Slick’ and ‘Sam’) – there is at least one of Slick or Sam on the board. These guys 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. Otherwise, they are harmless
* Red Blobs – (names ‘Ugg’ and ‘Wrongway’) – there are at least one of these on the board. They run up and down the left/right sides of the board. If Q\*bert touches one, he loses points. Once at the bottom they fall off the screen and reappear at the top after some time delay. Their speed increases with each round of a level.

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

**Screens** – There are 3 different screens:

* Instruction screen – can be accessed at any time by pressing ‘h’. The instructions for how to play are posted here
* Game play screen – the screen where the game is played
  + Game winning screen – when a round is one, the creatures disappear, and the top surface of each block flashes between original and finishing color quickly. Victory music should play too.
* New level screen - At the beginning of each level (level 1, 2, 3, etc) a screen appears displaying the level number and a short animation of Q\*bert hopping on cubes to show how to complete each round in the level. As levels progress, Q\*bert will have to jump on the cubes more than once to change their colors. This is part of the increase in game difficulty.

**UI –** There are several UI elements I’d like to incorporate, to make the experience more like playing the original game:

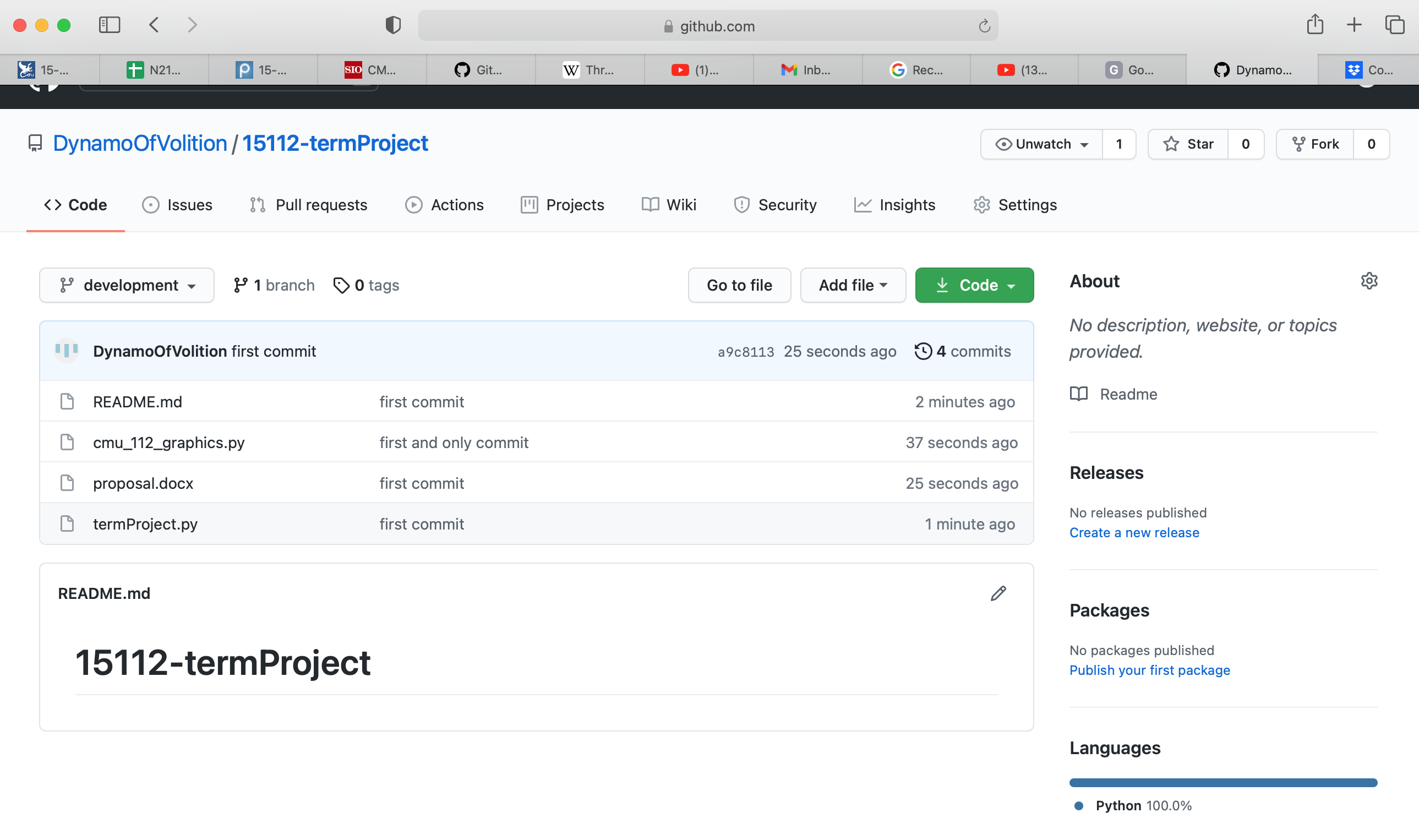
* Pixelated characters – just use cached images??
* Swirly hoverboards – cached images?
* Music and sounds from the original game: cube landing sound, round and level completion music

# Algorithmic Plan

I plan to implement AI to have an auto-play option of the game. The user will have the option at any point during the game to switch to an auto-play option, where Q\*bert will move according to some AI algorithm.

# Version Control Plan:

I will be backing up all work to my github account:



**EXTRA DETAILS**

enemyCreatures:

The creatures hindering Q\*berts progress in different ways:

* Coily – does not wander the board randomly, but chases after Q\*bert, always a hop or two away.
  + Should Coily arrive on same square as Q\*bert, some number of points are deducted (TBD).
* Slick and Sam – green blob creatures. If Q\*bert lands on a square they are already on, all creatures except Q\*bert freeze and Q\*bert has (TBD) seconds to traverse the board unhindered.

Hoverboard:

Q\*bert has some assistance in the form of hoverboards that sit to the left and right side of the board. Should Q\*bert feel cornered, if near a hoverboard, he can jump off the board onto it and be transported to the top square. After using a hoverboard, it disappears. Hoverboards vary in location and number based on level. The hoverboard is also used to lure Coily off the board. If Q\*bert jumps on a hoverboard and Coily is within one adjacent cube, Coily’s next hop will be off the board!.

There are 3 different pages/views that the player can see:

**Opening Instructions:**

|  |
| --- |
|  |
| Fig 1. Opening Instructions |

**In-between Level Board**

In between levels, a screen comes up stating the level number, and a short animation of Q\*bert traversing the cubes to demonstrate how to play this level. For example, Q\*bert animation on level 1 shows him hopping on each of the 4 demo squares once, changing their color.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Levels 1-3 Intermission Boards** | | |

**Game Board**

**During Play:**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Level 1: Round 1, 2, 3 Game Boards** | | |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

To simplify the display, only the score and the level and round number will be displayed on the background.

**Round Conclusion:**

At the end of a round, all creatures will disappear from board, a little music will play, and the top face of each cube will quickly flash between original and changed color. The next round will immediately begin, with a new, blank pyramid being drawn with no creatures. Q\*bert will fall from the top of the screen onto the top face of the top-most cube. The new score will be the old score with 1000 points added.