BurGerCing
Grace Cuenca, Brian Kwong, Connie Lei

APCS1 pd1
01/06/17

# Final Project Proposal

## “The point is to play a beautiful game.”

Tak is a board game taken out of the KingKiller Chronicles and made into reality. Our project will be a terminal implementation of Tak. It is a two player game that can be played on various different board sizes ranging from 3x3 to 8x8, although the most common is 5x5. Using the same rules, we will implement our project using player entered commands. The board will continuously be reprinted after each update. Similar to a chess or checkers game implementation, players will enter in different commands and use a grid to identify spaces on a board.

The first person to create a “road” with their “stones” between two opposite sides of the board wins. Roads do not have to be in a straight line. Players will attack, capture and defend their own pieces.

Official rules can be found at: http://cheapass.com/free-games/tak/

Possible Implementation:

The board will be represented by a 2D array, the columns and rows in the array corresponding to the tiles. The 2D array will each hold an ArrayList. The ArrayList will keep track of the pieces on each tile, called a stack. At the beginning of the game, the two players will enter in a single integer between 3 to 8 inclusive. A board with the integer inputted will be generated and the corresponding number of pieces each user will be allowed to have.

Players will enter in the tile using a column-row system similar to chess or checkers. They will be allowed 4 different moves depending on the tile in question. Each of these will be a method that checks certain requirements.

1. Placement of standing or flat stones or capstones on empty tiles.
2. Movement of standing or flat stones one tile onto a stack without a standing stone on it OR onto an empty tile
3. Movement of capstone one tile onto a stack OR a standing stone that falls OR onto an empty tile
4. Movement of a stack, tile distance will be determined by the height of the stack(max movement) and the user’s choices

We will have a parent class called Piece and two subclasses called Capstone and Stone (possibly a third called StandingStone and the Stone class would become FlatStone).

We will also need to have a check for a winning road which could be used as the boolean in a while loop to continuously run the game until someone wins. Players are limited by the number of stones they have and the rules.