Sungka

Programmer's Guide

Machine Problem 1

CS 11 AC2-M2-HRU

2018-02742	Figueroa, Petronilo III D.
2018-00479	Manansala, Paul Daniel E.
2018-09010	Pua, Aimee Julia U.
2018-21319	Taylo, Basil Benard A.

I. Game Description

The game is called “Sungka.” It is a classic game which has been played in the Philippines for a long time. This game is a virtual recreation of the classic game. It is a one (1) player game, wherein the player plays against the computer. The flow of the game will be discussed further in the Program Flow.

II. Modules

The program is divided into three (3) components:

The first module is the main.py. This module is the file to be executed for the game. It includes the syntax for the event handling (mouse hovering) in the Main Window. It has one class called class MainWindow(pyglet.window.Window).

The second module is the engine.py. It includes the syntax for file comprehension and the syntax for the game of Sungka as well. The scoring system depends on how many “shells” you collected in your “house.” Part of this module is the list of scores, wherein the the game numbers and the corresponding scores are listed.

The third module is the interface.py. This module includes the syntax for the graphic user interface (GUI). The reference used for the GUI is *Pyglet.* The code is divided into two (2) classes for the Sungka buttons, and the other for the main menu buttons. There are also two (2) labels: one for the base, and one for the house.

III. Resources

The resource used to learn *Pyglet* for the game's graphical user interface was the *Pyglet* Documentation website by Alex Holkner (https://pyglet.readthedocs.io/en/pyglet-1.3-maintenance/).

The program used the following images:

1. *Homepage*

   The leaves on the homepage background were taken from a free PNG resource website, Plus PNG (http://pluspng.com/png-48845.html), and then edited via Photoshop CC 2017.

2. *Game page*

   The banana leaf in the background and the Sungka board were made from scratch by one of the group members via Photoshop CC 2017.

The program also used the following fonts:

1. *Bellamy Script* (main game header) - https://creativemarket.com/Calamar_Art/1890451-Bellamy-Script

2. *CF Jack Story* (other text) - https://www.dafont.com/cf-jack-story.font

IV. Game Flow

The first thing that the player will see when he/she launches the game is the main menu, wherein the name of the game, "Sungka," is seen, with three buttons: Start, Scores, and Help. If the player selects start, it launches the game. If the player clocks the Scores option, he/she will see the list of scores of the previous games with their corresponding game numbers. When Help is selected, the instructions for the game is presented.

A player's goal in "Sungka" is to collect as many shells as you can and to place them into your base, and to have more shells than his/her opponent by the end of the game. In this modified, computerized version of the classic Filipino game, numerical digits will be used instead of shells, to make it easier for the player to know where they stand.

Here are the game mechanics:

1. The game is first prepped with each "house" (the smaller circles) having 7 "shells" (although in this version, instead of seeing 7 shells, each house will display the numerical digit 7).

2. Originally, the game starts off with both players taking their first turns at the same time, but in this version, only one player goes first. The player will choose which house he/she wants to start with, from his/her side of the board.

3. The player should pick up all the shells of the chosen house, and will go around the board, dropping a single shell into every house he/she passes (regardless whether or not they are his/hers, or the opponent's) by in a clockwise direction, including his/her own base.

4. If the last shell falls into a home that still contains any number of shells, he/she must pick up all those shells and go again.

5. If the last shell falls into one of the player's empty homes, he/she may collect the shells in the opposite home and add them to his/her base. His/her turn ends, and it's the opponent's (the computer's) turn to play.

6. If the last shell falls into one of the opponent's empty homes, the player will collect that last shell and add it to his/her base. His/her turn ends.

7. If the last shell falls into the player's base, he/she gets another turn, and can once again select any of his/her houses to begin another round.

8. The game ends when all houses are empty.

The player is scored based on how many shells are in his/her base when the game ends. He/she is considered the winner if his/her base has more shells than the opponent's.

The program uses the random function, which acts as the opponent of the player. This makes the game a one (1) player game.