DBZ: 112 Battle!

By: Surafel Tsadik

Description

DBZ: 112 Battle is a puzzle/strategy game where you and an opponent battle it out with teams of 7 characters each. When in battle, match up orbs of the same color to collect ki, the more you collect the stronger the attack! Gather 12 ki to do a Super Attack, which does the most damage! A player wins when all 7 characters on the opposing side are defeated!

Competitive Analysis

This project is inspired by the mobile game DBZ: Dokkan Battle, a game that I have played for many years now. There are many features in my game that are taken from this game to match the core gameplay and aesthetic. For example, the ki collection, 7-character teams, and the turn-based gameplay are very similar to that of Dokkan. I have chosen to make this game because the one thing that Dokkan is missing is an actual PvP game mode. Because of this, I wanted to put my own spin on Dokkan to make it more PvP-friendly (maybe at some point it will actually be played by two players). For example, in Dokkan you have one big health bar containing each character's HP added up. In my game I have decided to keep the HP for each character separate (to be able to defeat each character in a team separately). Also in Dokkan,

Structural Plan

In my one game file, there will be three possible screens, with a possibility of a few more later on. There is the start screen, the team select screen, and the game screen. The start screen is a simple screen that allows you to go to the team select screen when the app is opened. In the team select screen, you can pick between three set teams to go into battle with (the computer player will choose a random team between the two that are left). Finally, the game screen will show the actual game in progress.

Algorithmic Plan

The two trickiest parts of my game involve the ki collection and the attack phase of a turn. For the ki collection, the game has to be able to find the path (based on the starting orb) that will collect the most ki. A path is determined by orbs of the same color that appear consecutively. I plan to achieve this by using a backtracking function that with check each possible move from some orb on the board (possible moves involve orbs that are up one row and either a +1, 0, or -1 change in the column), and if it is a legal move, it will recall the function as if it was the next row. For each of the possible next moves, they will return a score of how many orbs will be collected if that orb is selected as the next move. Then using the best template, it will find the best possible next move and make that its next move.

For the attack phase, I plan on using a version of the minimax algorithm to create an AI that will either find a way to deal the most damage in aa single attack phase, or go for a kill if it presents itself to the AI (if more than one kill presents itself, the AI will go for as many kills as possible). I also plan on using recursion for this feature. To determine a specific node's score, I will define it as how many kills can be achieved with a single attack. If no kills can be achieved, it will be differentiated by how much total damage is done with an attack. The move tree will go down to a depth of two (where both the AI and player attack someone with a single character) and a score will be defined for each node, and I will make the AI as the maximizing player and the human player the minimizing player.

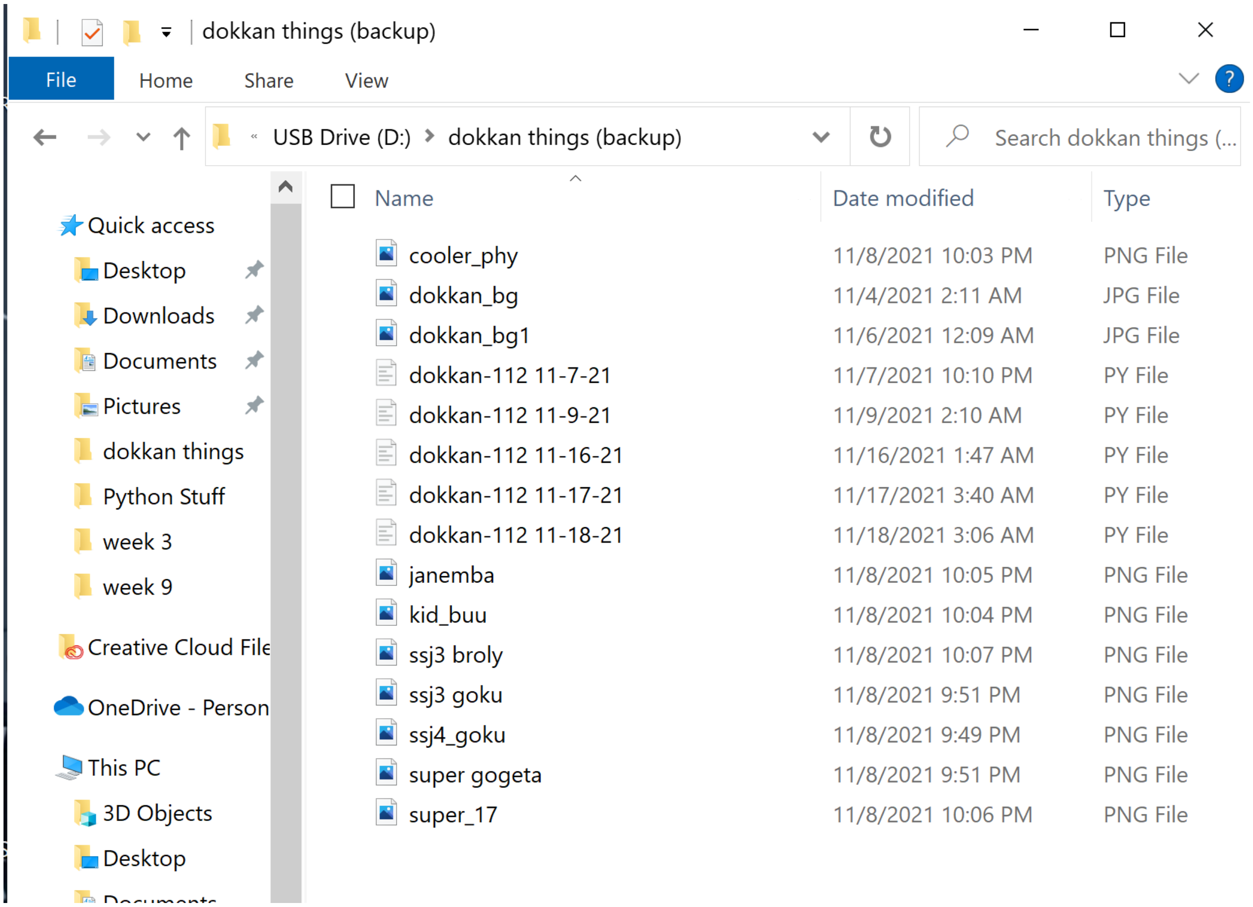
Timing Plan

I plan on having the ki collection and most of the game functionality by November 20th (if not before then as I have a lot of the game's functionality mostly done), and I will spend the weekend and the Monday afterward max to get the minimax AI done (that should be more than enough time to get this done). With how much I have done already, I should be able to make MVP by the 23rd (hopefully).

Version Control Plan

To back up my code, I have a separate flash drive that not only has all of the images that I will potentially need, it also has all of the past versions of my project. Basically, at the end of every day that I work on the project, I make a whole new save file of the project and save it into the flash drive. This means the flash drive will have multiple saves of the project that includes the work done on the day listed.

This is what it looks like now:



Name	Date modified	Type
cooler_phy	11/8/2021 10:03 PM	PNG File
dokkan_bg	11/4/2021 2:11 AM	JPG File
dokkan_bg1	11/6/2021 12:09 AM	JPG File
dokkan-112 11-7-21	11/7/2021 10:10 PM	PY File
dokkan-112 11-9-21	11/9/2021 2:10 AM	PY File
dokkan-112 11-16-21	11/16/2021 1:47 AM	PY File
dokkan-112 11-17-21	11/17/2021 3:40 AM	PY File
dokkan-112 11-18-21	11/18/2021 3:06 AM	PY File
janemba	11/8/2021 10:05 PM	PNG File
kid_buu	11/8/2021 10:04 PM	PNG File
ssj3 broly	11/8/2021 10:07 PM	PNG File
ssj3 goku	11/8/2021 9:51 PM	PNG File
ssj4_goku	11/8/2021 9:49 PM	PNG File
super gogeta	11/8/2021 9:51 PM	PNG File
super_17	11/8/2021 10:06 PM	PNG File

Module List

No need! (at least before MVP)

TP2 Update

In between the tp1 and tp2 checkpoints, many things have been added. The biggest thing that has been added is the Attack Phase. This phase is a very important part of the project as it is where the game can progress to some end (w/o this there wasn't an end to a match). Each character for each player will have a chance to attack a character from the other team. The damage dealt depends on the ki gained in the previous phase (if 12 ki is obtained, the attack stat for that character is doubled). After every character gets the chance to attack, the current rotation goes to the "end of the line", the next three characters for each team become the current rotation, and we go back to the ki collection phase. With this, the game now has an end (where a player wins if he defeats all of the opponents characters in their team. Since this game has an AI, the AI for the Attack Phase is made using an adapted minimax algorithm to make decisions on which character to attack. Other smaller changes include some more images to the various screens in the game, and the ability to restart the game from the beginning when it finishes.

TP3 Update

In between the tp2 and tp3 checkpoints, I was able to add to features that are not as important, but were still needed. I was able to allow for the 3 characters on rotation to be switched into a different order by the choice of the player. This is done by just clicking on the character that you want to switch and then clicking on the character that you want to switch with. The second feature allows the player to see the order of their team and figure out which character is coming up next. This is done by simply clicking on the “Your Team” button. To back out of this just click the “Back” button located in the same place as the previous “Your Team” button.