John Burke and Daniel Li

Comp 401 A

Final Project

**Background:**

The purpose of our app, Benkyo Blob, is to gamify studying and to help build better study habits. Users can use the app to track their study time, take quick notes, study with quizzes and flashcards, and save the dates of their future tests. The audience of our app is students that are looking to build better study habits and to gain motivation to study.

**Significance:**

Our app uses gamification elements to encourage students to develop their studying. The impact of our app is to improve student’s performance in school and university with better studying.

**Innovation:**

Although there are other studying apps that exist, our app has the exclusive feature of incorporating digital pets. Similar to Digimon or Tamagotchi, our users take care of a digital pet. The user’s studying determines the health of the pet, which is a feature unique to our app.

**Approach:**

We use computed properties when determining the stage, mood, and evolution of the pet. It makes sense to use computed properties here so that we wouldn’t have to store every possibility and iterate over them. The code is also more readable with the computed properties.

We used NavigationViews to navigate between each screen. We used custom buttons to travel backward in the NavigationView. It makes sense to use NavigationView since we have many views that we are often switching back and forth between each other. We used a List in the EditQuizView to list out each of the questions that could be edited. We also used a List when listing the saved dates for future exams in the CalendarView. It makes sense to use Lists here since we are displaying an array of items all in the same way. We used a ScrollView when displaying the quizzes and the flashcard decks in the QuizesView and FlashcardsView. It makes sense to use a ScrollView here because a user could have a long list of quizzes or flashcard decks, and in order to view them all it would be easy and familiar to scroll through the list. We use a DatePicker in the CalendarView to choose the date when adding a future exam. It’s self-explanatory why we would use a DatePicker to pick a date.

We used a WebView in the UnlockablesView to view more information about the collectible you unlocked. You can view a video about the collectible, such as a video on how to cook it if the collectible is a food. This provides interesting information to the user about the collectible that they unlocked.

We parsed local JSON data to get the default quiz and flashcard sets. It made sense to do this so that the defaults didn’t have to be hard-coded into the ViewModel.

We used custom art and fonts in our app to give it a pixel art theme. This was to reference back to the original digital pets and to give our app a unique look.

**Team Contributions:**

John:

* All quiz views and models
* Started the CalendarView using cocoapods (we later switched to DatePicker, which Daniel implemented)
* Parsing local json data for quizzes and flashcards
* Quiz and calendar navigation from MainBlobView

Daniel (Sora):

* Custom art and font
* Study timer and notes
* All flashcard views and models
* MainBlobView
* CalendarView
* UnlockablesView

**Process Appraisal:**

We think that the project went well and that we each had tasks separate enough from each other that we could work without being dependent on the other’s work. An unexpected challenge was changing from using cocoapods for the calendar to using the DatePicker, as the DatePicker offered a better way of doing what we wanted. Testing that the collectibles worked properly was also a challenge, as we had to continually use the app to find them based on random chance. The application does well in having an appealing art style and in being simple enough that it is useful but not distracting from the user’s studying. Future versions would include saving data, the ability to move flashcards between decks, building quizzes off of flashcard decks, more collectibles, and other quiz modes (such as hangman).