TitlePage
- #onCreate()
- +goToPreGameSelection(view: View)
- +goToLeaderboard(view: View)
- +goToOptions(view: View)

GameTimer
- +GameTimer(startTime: long, interval: long)
- +onFinish()
- +onTick(millisUntilFinished: long)

SinglePlayer
- -model: SinglePlayerModel [1]
- -gameTimer: GameTimer [1]
- -numWordsDisplayed: int [1]
- -bg: int [1]
- #onCreate(savedInstState: Bundle)
- +onKeyDown(key: int, event: KeyEvent)
- +initialDisplay(animalID: Drawable, backgroundID: Drawable)
- +displayWord(wordIndex: int, word: String)
- +displayTime(secondsLeft: long)
- +displayScore(score: int)
- +highlightWord(wordIndex: int, word: String, letterIndex: int)
- +update(Observable: Object)
- +goToPostGame()

PostGameScreen
- #onCreate()
- +goToTitlePage(view: View)
- +goToPreGameSelection(view: View)

PreGameSelection
- -diff: View [1]
- -animal: View [1]
- -background: View [1]
- #onCreate()
- +setDiff(view: View)
- +setAnimal(view: View)
- +setBackground(view: View)
- +goToSinglePlayer(view: View)
- +goToTitlePage(view: View)

SinglePlayModel
- -currWordIndex: int [1] //index of current word in words displayed
- -currLetterIndex: int [1] //index of current letter in current word
- -wordsDisplayed: int [1] //array of 5 represent indices in words list
- -wordsList: String [1] //array = [apple, bear, cat, dog, elephant, ...]
- -nextWordIndex: int [1] //index of next word in words list
- -score: int [1] //current user score, # of letters typed
- -numWordDisplayed: int [1] //final value
- -am: AssetManager [1]
- -currKeyLetters: Set<Character> [1]
- +SinglePlayModel (diff: States.difficulty, am: AssetManager, wordsDis: int)
- +getWordsList (diff: States.difficulty)
- +populateDisplayedList()
- +typedLetter (letter: Char) //takes in the letter typed
- +updateWordsDisplayed() //notify view to put words on screen

Handwritten annotations:

- cause 2
- observer
- same class
- 1 vertex.
- creates / cause 1
- creates / cause 2
- multiplicity [MA]
- single page model
- observable
- view: construct model (1)
- model updates view (1)
- cause Y / creates

MA: { SP → GT → same cause.
     { SP → SPn

Circled numbers: 1, 2, 3, 4, 5, 8, 9, 9, 12, 17