

BOGGLE BY TEAM 12

REQUIREMENTS ANALYSIS DOCUMENT

Team 12
Kolby Johnson
Quinn Lennemann
Nischal Neupane

INTRODUCTION

PURPOSE

Our purpose is to create a digital single player Boggle game that is readily available to users to experience fun by themselves if they are lonely, such as during Valentine's Day.

SCOPE

The scope of this project is relatively small due to the purpose of creating a simple and easy to use game.

OBJECTIVES AND SUCCESS CRITERIA

Our objective is to recreate Boggle in a digital environment. The application should be a fully functional and implement all the original Boggle rules.

DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

1. RAD : Requirements Analysis Document
2. TBD : To be determined
3. Timer : an object enforcing the time limit of the game

REFERENCES

1. Increment 1.pdf
2. [Boggle Game Rules](#)

OVERVIEW

Recreate an easy to use single player boggle game digitally with proper implementation of the rules of a boggle game.

CURRENT SYSTEM

OVERVIEW

The application will be a GUI application consisting of a 4x4 matrix and viable dice letters. The players can type out all the words, and the correctness of the words will be determined with the use of a local dictionary file. The UI will also display a 3 minute timer and live score updates with every word that user gets correct.

PREVIOUS FUNCTIONAL REQUIREMENTS

- F1. Start a new game
- F2. Recognize player inputs
- F3. Check for valid words
- F4. Enforce time limit
- F5. Calculate the player's score
- F6. Track submitted words
- F7. Display animation at start of game

NEW FUNCTIONAL REQUIREMENTS

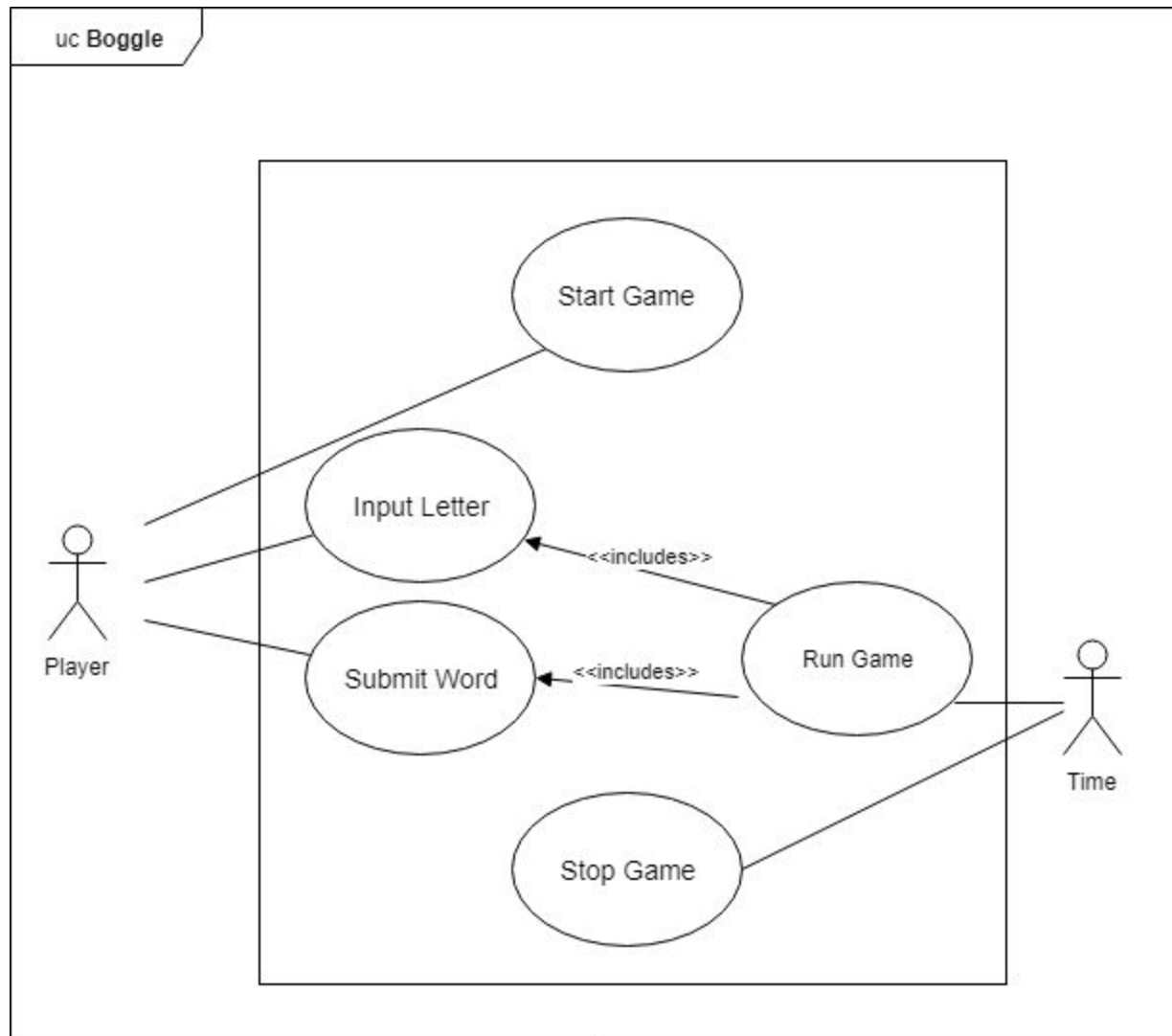
- F8. Supports German
- F9. Player is able to choose which language they want to play the game in

NONFUNCTIONAL REQUIREMENTS

- N1. User should be able to understand and play the game within a minute
- N2. The UI will accurately display the state of the game
- N3. There should be visual feedback for user input
- N4. Program should be fast enough to keep up with user input
- N5. Application should be able to run on multiple platforms

SYSTEM MODELS

USE CASE MODEL



Start Game:

- 1) Player starts the game
- 2) Board displays animation

- 3) Board generates dictionary
- 4) Board randomizes
- 5) Board starts timer

Post-condition: game started

Input Letter:

Pre-condition: game started

- 1) Player inputs letters
- 2) Board checks if word is valid
 - a) If it is invalid, end use case
- 3) Board highlights the letter entered

Submit Word:

Pre-condition: game started

- 1) Player submits a word by pressing enter
- 2) Dictionary checks if the word is valid
- 3) Game calculates the score of the word
- 4) Board clears the highlighted letters

Run Game:

Pre-condition: game started

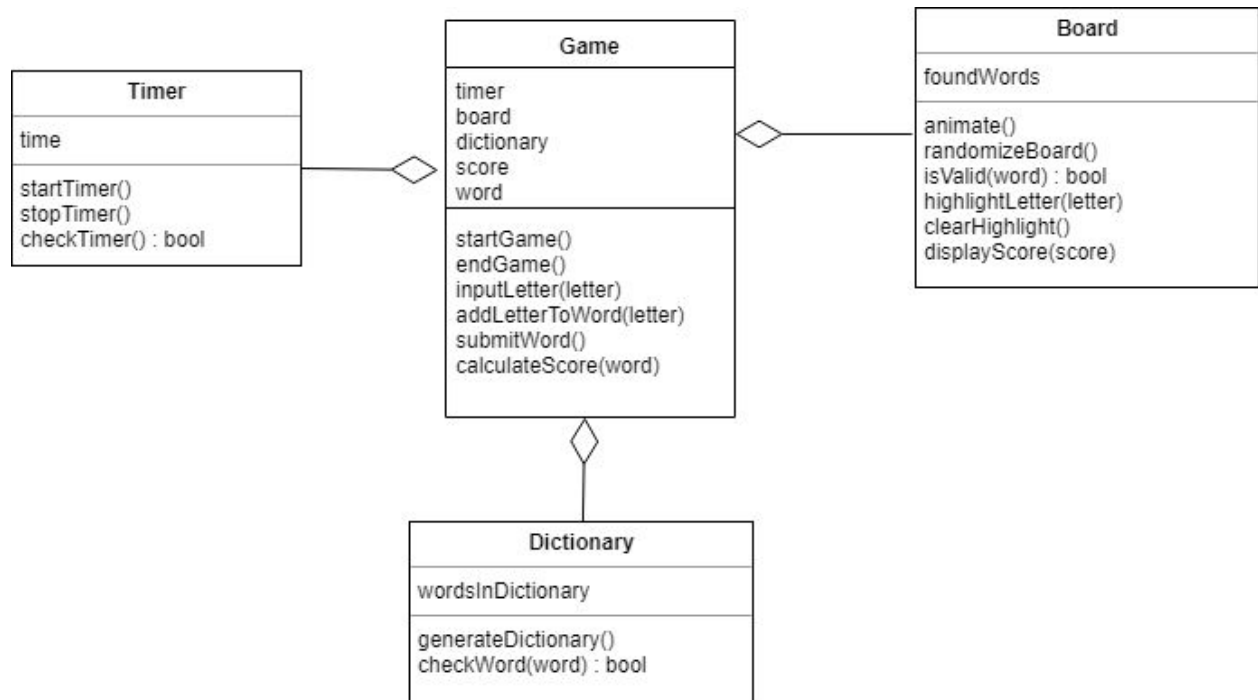
- 1) Timer does initial time check
- 2) Input Letter
- 3) Submit Word
- 4) If timer has expired, move to post-condition
 - a) If timer has not expired, go to Step 2

Post-condition: game ended

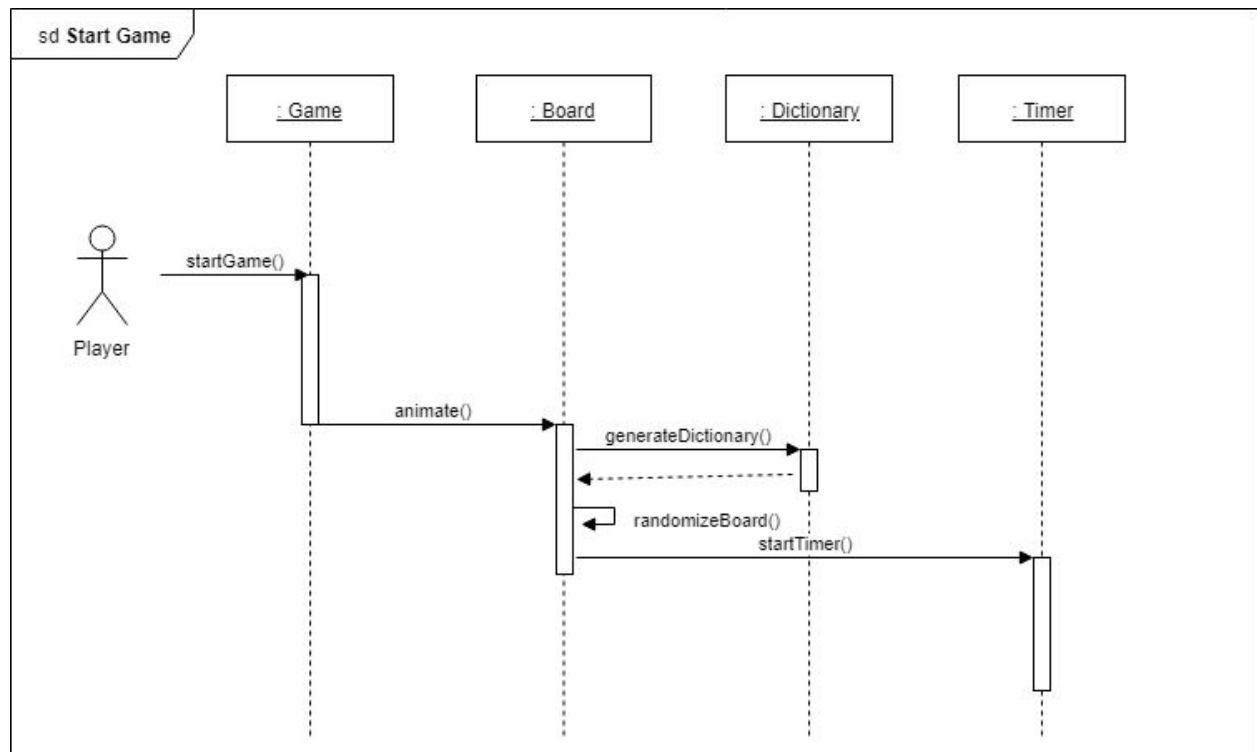
Stop Game:

Pre-condition: game ended

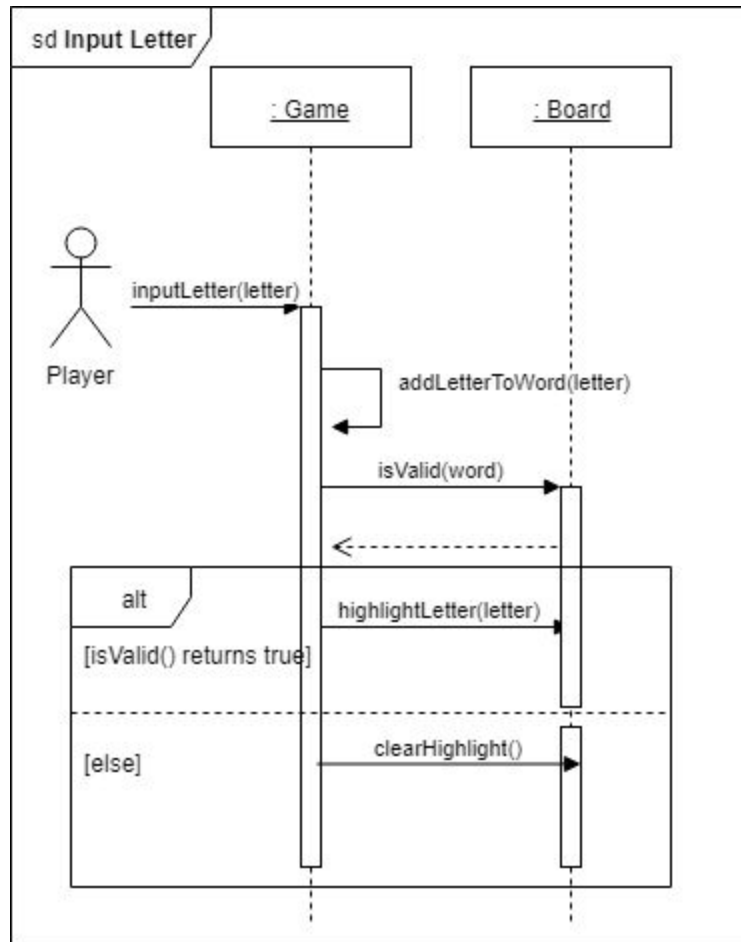
- 1) Game ends game when time runs out
- 2) Timer stops
- 3) Board clears the highlighted letters and displays score



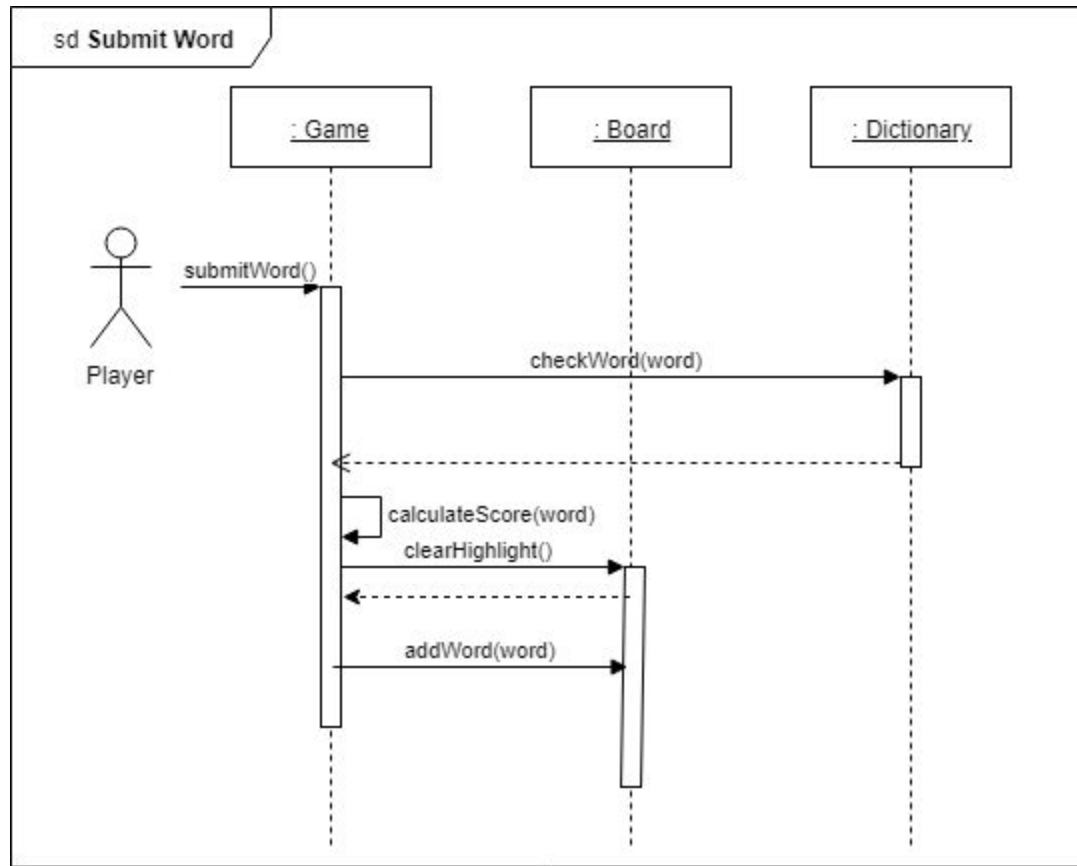
Start Game:



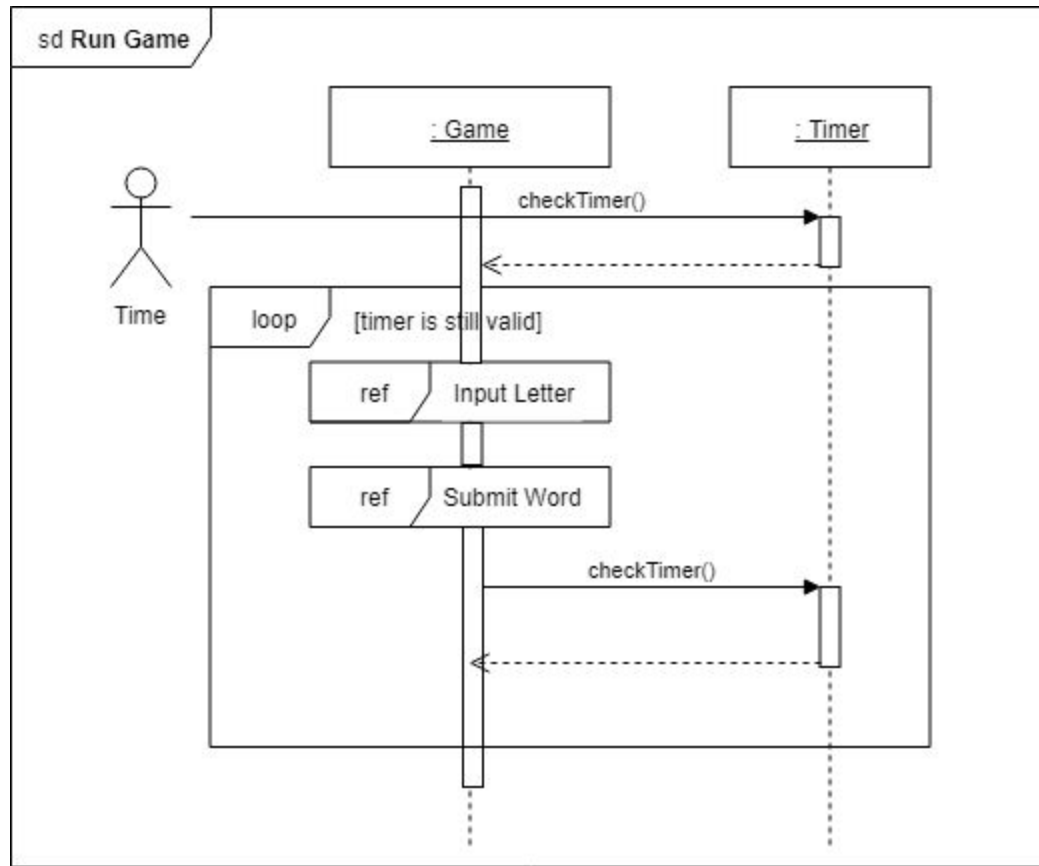
Input Letter:



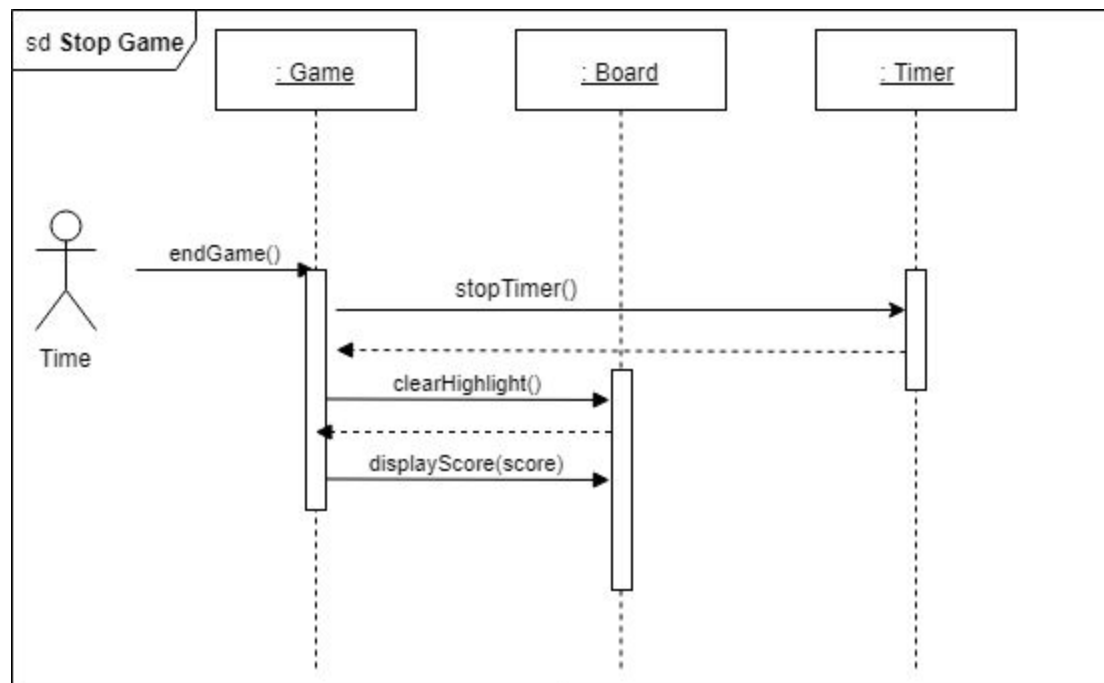
Submit Word:



Run Game:



Stop Game:





GLOSSARY

1. Boggle
 - Simple word game created by Hasbro
2. Word
 - As defined by the Scrabble dictionary
3. Requirements
 - Tasks to be completed in order to have a successful program