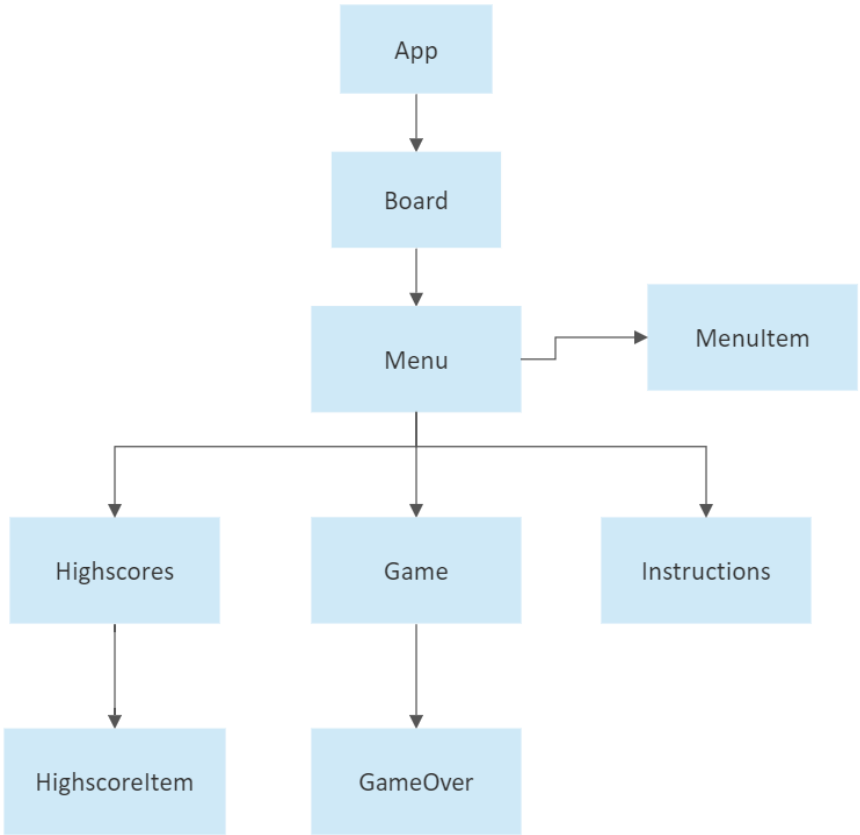
Bug Squasher

Demo:
https://64b82c8babfec9117f71bd8b--preeminent-liger-c59629.netlify.app/

Features:

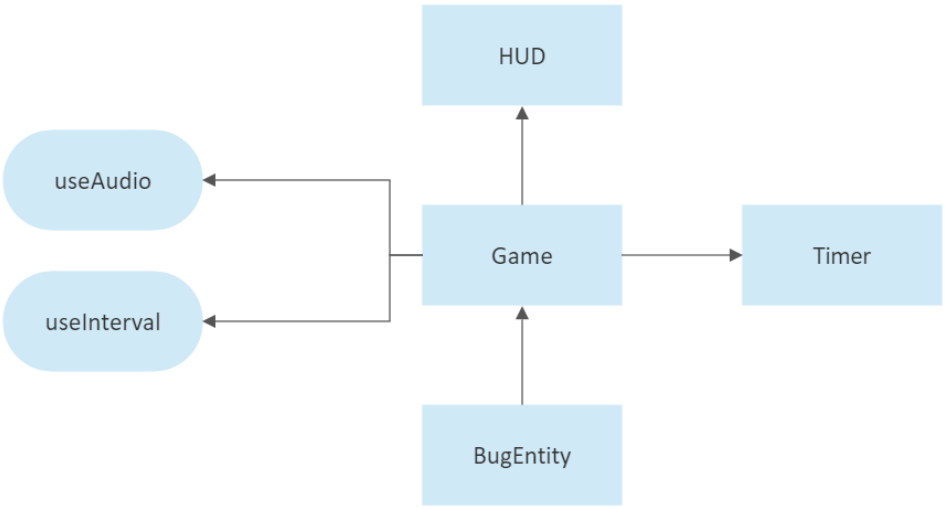
- Smashing bugs never felt this good.
- Hit sounds.
- Highscores table using local storage.
- Different scene views.
- Responsive design.
- Three different types of bugs to smash. Each bug type has its own speed/direction threshold.

Architecture:



Components hierarchy:

- **App (and main.tsx)**: Main entry point of the application.
- **Board**: Game board which encapsulates the persistent states of the application which will be registered to the context. Also handles the different scenes transitions.
- **Menu**: Main menu which is hosting the different navigation options of the application (play, instructions, highscores)
- **MenuItem**: Represents a single navigation option.
- **Highscores**: Displays a list of the highest scoring players (which are saved in local storage), the data is persistent.
- **HighscoreItem**: Represents a single highscore entry with the id, name and score value of the player.
- **Instructions**: A simple page which displays useful information for the player and gives a brief background for the game.
- **Game**: The main game components which stores all the logic and the game progression. Bugs are represented as entities which are derived from a GameObject type.
- **GameOver**: Displays the score for the player when the time's up. Also allows the player to save his progress persistently by registering to local storage.

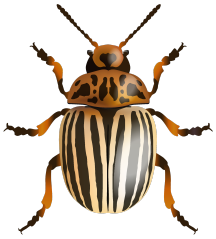


Game component hierarchy:

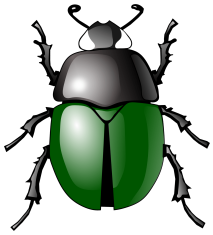
- **HUD**: Heads On Display - as the name states, displays the current scoring state and timer for the player.
- **Timer**: Handles the game ticks and is responsible for the representation of the in-game timer.
- **BugEntity**: The main entity base component which represents the 3 different bugs entities. Handles the logic for positioning and moving the bug, changing the bug's direction over time and squashing the bug.
- **useAudio**: Custom hook that handles all the sound and audio related stuff. (with caching)
- **useInterval**: Custom hook that handles the progression and animation of every entity in the game. (Bugs, Timers, ..)

Bugs Types:

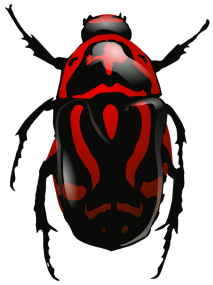
There are 3 different bug types:



This is the starter bug, his movement and direction changes are mostly slow. Squashing it scores the player with **10** points.



The green bug moves moderately faster than the starter bug, and changes directions quite often. Squashing it scores the player with **40** points.



The red bug is the fastest and slippiest of the three bugs, it switches directions really fast. Squashing it scores the player with **80** points.

Scenes

Navigation throughout the application is made possible and controlled by the Board component. These are the existing scenes that can be navigated to inside the application:

- Menu
- Highscores
- Instructions
- Game
- GameOver

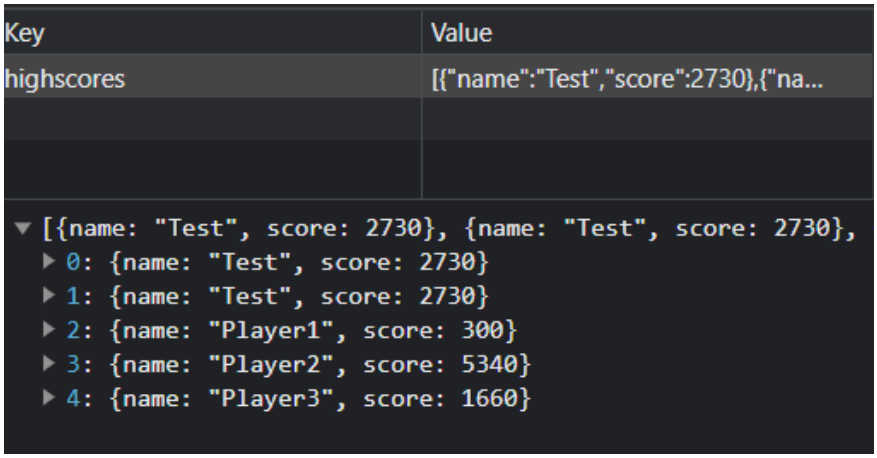
Techincal Stack:

The project was made with React running Vite as the development environment.
Functional components together with hooks and custom hooks to handle all the logic of the application.
useContext was used for the persistent state throughout the app.

Caching:

The persistent player's data for highscores is saved through local storage.
The name of the key is 'highscores'.

This is how the data is represented:



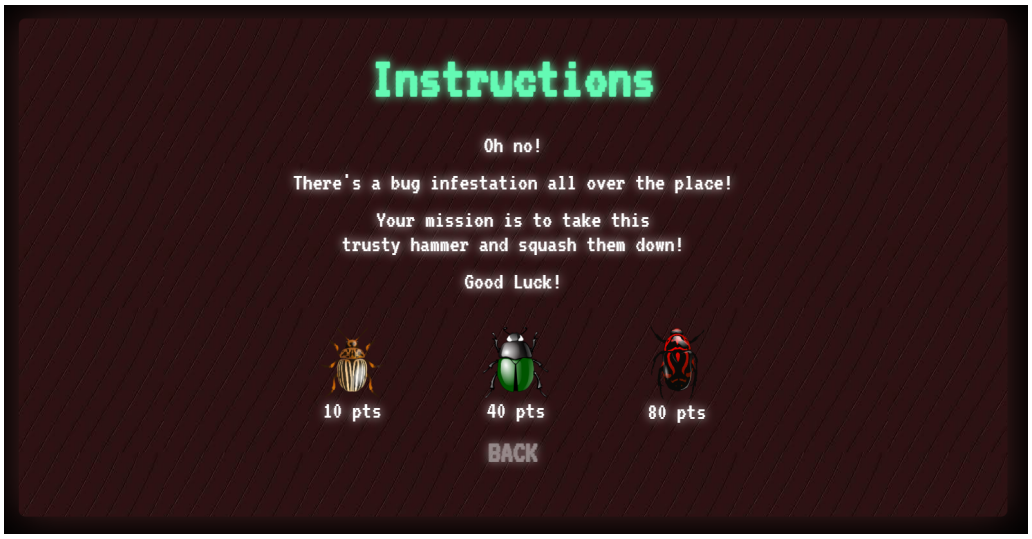
Responsiveness:

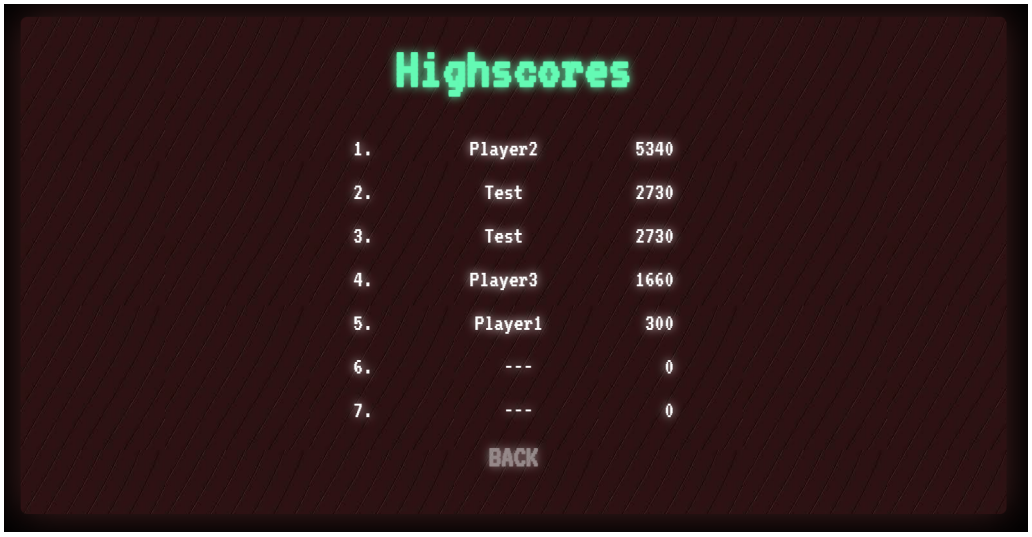
The responsiveness of the application was made possible by the use of media-queries.

Things I woud do differently / additionally:

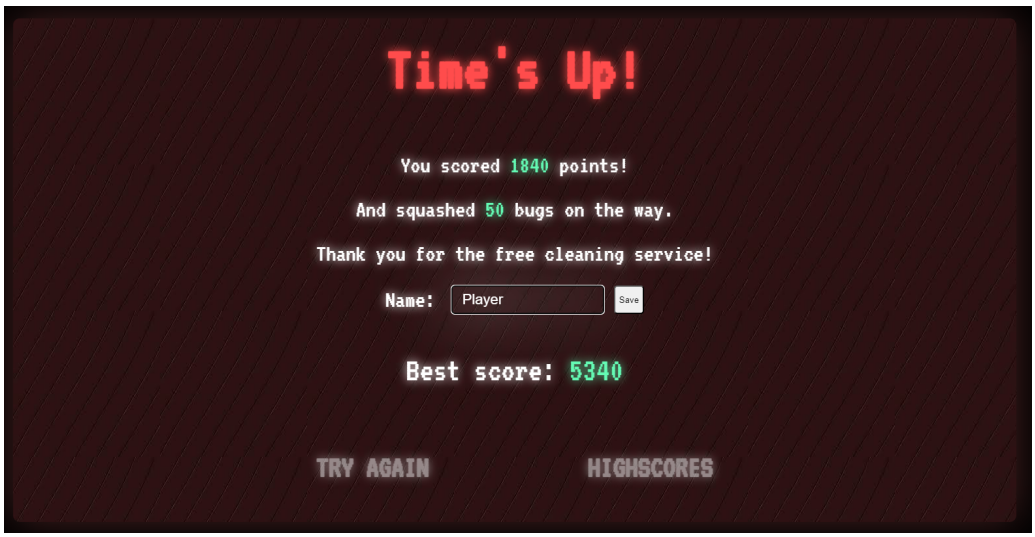
- The most important thing that I had no time finishing was the removal of dead bugs from the GameObject array structure.
- Add more sound effects.
- Increase performance by using an html canvas.
- Add more effects like the breaking of glass when not hitting a bug.
- Add more directional choices for the bugs and not just north/south/east/west.
- Add smooth animations for walking/turning.
- Add a scene transition mechanism for fading in/out of scenes when switching between them.
- Connect the highscores feature to a real database to engage players to play against one another.
- Add an option menu for sound volume / effects on or off.
- … This project can be taken to so many more heights

Applications Screenshots:




Highscores
1. Player2 5340
2. Test 2730
3. Test 2730
4. Player3 1660
5. Player1 300
6. --- 0
7. --- 0
BACK


SCORE: 480
00:14


Time's Up!
You scored 1840 points!
And squashed 50 bugs on the way.
Thank you for the free cleaning service!
Name: Player
Save
Best score: 5340
TRY AGAIN
HIGHSCORES

Mobile: