#2

PROJECT MANAGEMENT LOG	
First Name: Swoyam	**Surname:** Pokharel
Student Number: 2431342	**Supervisor:** Prakriti Regmi
Project Title: Oops!Mate.	**Month:** September

What have you done since the last meeting

- Implemented & Tested 3 different approaches to generating moves. (iterative, magic bitboards, PEXT boards)
- Slides for explaining how each of those implementations work
- Updated the previous `board` representation for a performance gain (~10-30ns to ~0-10ns)

What do you aim to complete before the next meeting

- Implement a unified `Move` type
  Create a compact `u32`-based representation that stores `from`, `to`, `capture`, `flags`, and other metadata.
- Attack -> `Move` type generator
  Use the PEXT-generated attack tables to efficiently produce moves, then wrap them into the `Move` type with proper flag handling (promotion, en passant, castling)
- Wrapper around the board to generate moves as `Move` type, so the `Moves` are stateful
- TL;DR: Generate Psuedo-Legal moves.

Supervisor comments

Appendix - Project Log Book Entry Template

#1

PROJECT MANAGEMENT LOG	
First Name: Swoyam	**Surname:** Pokharel
Student Number: 2431342	**Supervisor:** Prakriti Regmi
Project Title: Oops!Mate.	**Month:** September

What have you done since the last meeting

- Bitboard Based Board Representation
- U8 based piece representation
- Utils for moves, captures, removes / set bits
- Utf8 characters for piece representation
- Slides explaining how it all works

What do you aim to complete before the next meeting

- Implement Iterative Approach For Move Gen
- Implement Magic Bitboards For Move Gen
- Implement PEXT Boards For Move Gen
- Benchmark All 3.

Supervisor comments

Appendix - Project Log Book Entry Template