EECS 314 - Project Report

Haley Eisenshtadt, Jason Kuster, Ian Lavelle, Joseph Satterfield, James Wright

April 29, 2015

1   Problem Statement

While learning MIPS in EECS 314, our group noticed a lack of fun, interactive tools to make learning MIPS easier. There are helpful demonstrations in the textbook and you can always open up QtSpim and play around, but there wasn't anything particularly graphical and interactive to help people learn MIPS. Our group wanted to put together an Android app to make the process of learning MIPS easier. The tool we settled on to build this was Unity, a cross-platform game engine which several of us had used before which made game development easy.

2   Major Challenges

There were a couple of challenges we ran up against while developing this set of games.

1. Item 1

2.

3   Key Components

Our project has three key components which come together to make the MIPS learning process easier. We have the MIPS Simulator, which is a graphical MIPS simulator in the style of ALICE for Java, we have MIPS Bejeweled, a twist on the classic Bejeweled game which helps teach people how MIPS instructions work, and we have the Pipeline Bakery which presents the MIPS pipeline in a fun, graphical way.

3.1   MIPS Simulator

3.2   MIPS Bejeweled

MIPS Bejeweled takes the classic bejeweled game and adds the twist of only completing rows when those rows create a valid MIPS instruction. In order to make this work we needed to remove the requirement in Bejeweled of reversing a move if it doesn't make a
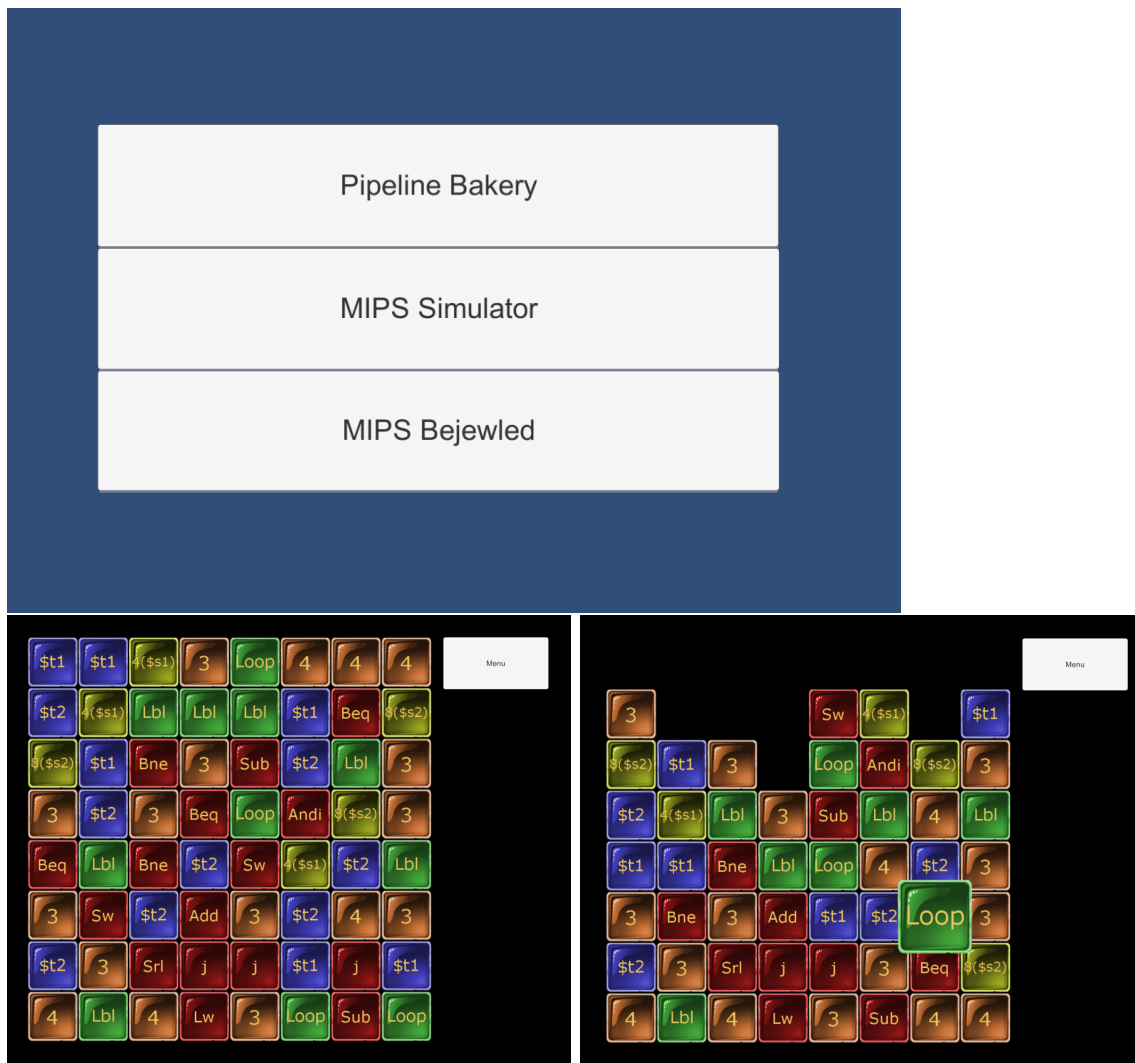
valid combination because with the added complexity of MIPS instructions the game was impossible to play. We feel that this is an acceptable compromise in the name of usability. We also changed out the typical falling physics of classic Bejeweled for a swapping mechanism that encourages the player to think ahead.
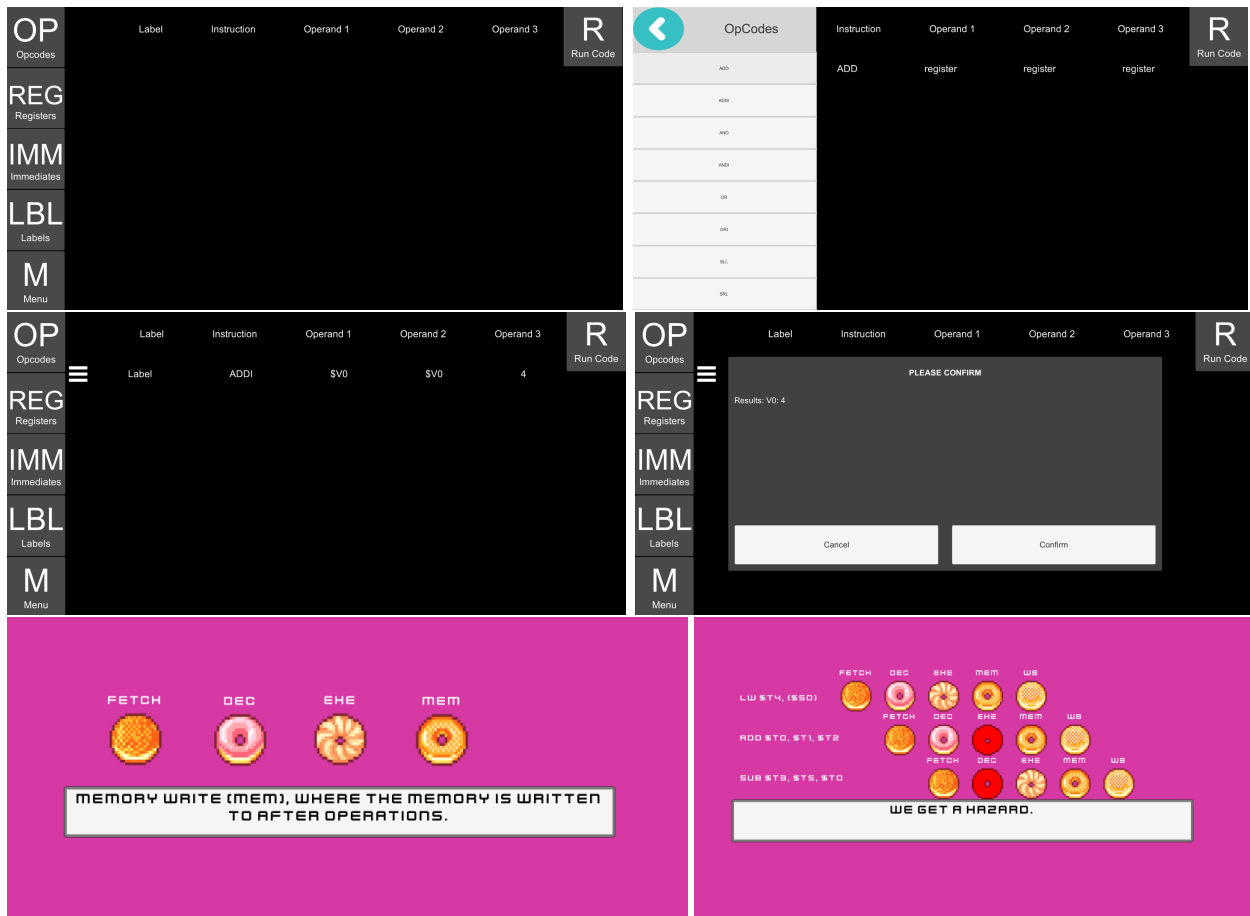
3.3 Pipeline Bakery

4 Integration

5 User Interface

6 Snapshots

OP
Opcodes

REG
Registers

IMM
Immediates

LBL
Labels

M
Menu

Label   Instruction   Operand 1   Operand 2   Operand 3   R
Run Code

OpCodes   Instruction   Operand 1   Operand 2   Operand 3   R
Run Code

ADD
ADDI
AND
ANDI
OR
ORI
SLL
SRL

ADD   register   register   register

OP
Opcodes

REG
Registers

IMM
Immediates

LBL
Labels

M
Menu

Label   Instruction   Operand 1   Operand 2   Operand 3   R
Run Code

Label   ADDI $V0 $V0   4

OP
Opcodes

REG
Registers

IMM
Immediates

LBL
Labels

M
Menu

Label   Instruction   Operand 1   Operand 2   Operand 3   R
Run Code

PLEASE CONFIRM

Results: V0: 4

Cancel   Confirm

FETCH   DEC   EHE   MEM

MEMORY WRITE (MEM), WHERE THE MEMORY IS WRITTEN
TO AFTER OPERATIONS.

LW $T4, ($S0)   FETCH   DEC   EHE   MEM   WB
ADD $T0, $T1, $T2   FETCH   DEC   EHE   MEM   WB
SUB $T3, $T5, $T0   FETCH   DEC   EHE   MEM   WB

WE GET A HAZARD.

7 Individual Contributions

- Haley Eisenshtadt

    Pipeline Bakery Design/Implementation

- Jason Kuster

    Bejeweled Block Physics

    Project Report Lead

    Project Report Compilation

- Ian Lavelle

    Bejeweled Base Implementation

    Bejeweled Instruction Logic

- Joe Satterfield

    MIPS Learner Co-Implementation

- James Wright

  MIPS Learner Co-Implementation