

ECE 350 Project Proposal
Matthew Barbano, Nathaniel Brooke, Jesse Yue

Overall Idea:



Plan for Inputs/Outputs:

Inputs:

- Game Controllers (2 to 4)

Outputs:

- VGA
- Audio
- RGB LEDs

Tasks:

Task 1: Physics Engine

- This is the core of character movement. Movement in the game involves gravity, inertia, and the ability for characters to double jump and control their acceleration while airborne. On a surface, control of movement is more direct, controlling character velocity directly.
- This is a difficult task. It will require a lot of programming, which in MIPS is no small feat.
- Point Value Proposal: 30 points
- Input: Game Controller (see other tasks)
- Output: VGA

- Processor Use: Extensive use of processor to perform necessary calculations for the physics, and to read inputs from the controller and determine character's movement.

Task 2: Hacked Controller

- Hack a single existing controller to make it interface with the game. One feasible possibility is to connect a Nunchuck to an Arduino, then interface via additional breadboard circuitry this with the DE2 board. Another possibility is a Game Cube controller (which we choose depends on accessibility of documentation of each controller and which controllers we have access to).
- This will be a difficult task because it requires learning how the Nunchuck or other controller works via documentation, and because it requires designing additional circuitry on a breadboard to interface it with the DE2.
- Point Value Proposal: 20 points
- Input: Nunchuck, Game Cube controller, or other controller we can access
- Output: None
- Processor use: A significant amount to convert input signal into changes on VGA screen (e.g. character movement)

Task 3: Controller Finishing

- This task will be to solder the breadboarded circuit interfacing the hacked controller with the DE2 board onto a perfboard, or to use Eagle CAD to print a PCB of the circuit (choose one).
- This will be an intermediate task because it requires learning about perfboards or Eagle CAD, which have not been taught in this class.
- Point Value Proposal: 20 points
- Input: This is the main game controller.
- Output: None
- Processor Use: Nothing Significant

Task 4: The Advanced Stage

- This task involves designing a stage that combines several components: The stage must have a graphics image, and several types of surfaces. There will be the vanilla surface that you can stand on and cannot travel through, but there will also be platforms that you can stand on, and can travel through. The edge of the stage will have locations onto which you can grab and hold on to if you hit the stage from the side.
- This task is of moderate difficulty.
- Point Value Proposal: 20 points
- Inputs: None
- Outputs: VGA

- Processor Use: Significant use to compute player response to different surfaces.

Task 5: Displaying Characters

- This is how characters will be visualized on a display. This will involve movement of characters and animation of attacks.
- Displaying and animating characters with mif files is a difficult task
- Point Value Proposal: 20 Points
- Input: none
- Output: VGA
- Processor Use: Consistent use to refresh location of characters on screen

Task 6: Scoring System

- This will store and display the amount of damage characters take and display how many lives a character has left
- This is a simple task to keep track of lives and damage
- Point Value Proposal: 10 Points
- Input: none
- Output: VGA
- Processor Use: Some, to update number values

Task 7: Character Selection

- This task involves adding a menu to the game, such that players can choose their character among a list of options.
- This is a difficult task, as it requires adjusting the game so that the menu is shown before the game actually runs, and allowing characters to be swapped out in gameplay.
- Point Value Proposal: 30 points
- Inputs: Game Controller
- Outputs: VGA
- Processor Use: Heavy processor use to control character selection and load a character into the game based on the user's choice.

Task 8: Stage Selection

- This task involves adding a second menu to the game, such that one player can choose the stage on which the bros smash each other.
- This is an intermediate difficulty task, as it requires adjusting the game so that multiple stages are supported and can be selected from.
- Point Value Proposal: 20 points
- Inputs: Game Controller
- Outputs: VGA

- Processor Use: Heavy processor use to control game selection and adjust game parameters based on the selected game.

Task 9: Sound Effects

- This task will involve adding sound effects when certain events occur in the game, such as a punch sound when a player attacks. Note these are not simple single-frequency sounds as produced in lab, they will often require experimentation with overlaying of multiple harmonics to produce the desired timbral effects.
- This is an intermediate level task, as significant experimentation will be required to achieve the desired timbral effects.
- Point Value Proposal: 20 points
- Input: none
- Output: speakers
- Processor Use: Significant, to detect when the triggering events occur (especially significant use of branching instructions for condition detection and response)

Task 10: Music

- This task involves composing or finding a musical piece to use as background music by generating tones. A Java (or other high-level language) script will need to be written to convert musical notes into bits for a mif file, and the processor used and Verilog code added to output this to the speakers.
- This is a medium difficulty task, generating and sequencing tones correctly takes significant time
- Point Value Proposal: 20 Points
- Inputs: none
- Outputs: Speakers
- Processor Use: Some, to loop the music and read the mif file

Task 11: Amplification System

- This task involves constructing the hardware for amplifying the sounds produced by the speakers of the system.
- This will be an intermediate level task, as it involves using knowledge from outside this class and interfacing it with the components from this class.
- Point Value Proposal: 20 points
- Inputs: none
- Outputs: speakers
- Processor Use: none

Task 12: NPC/AI

- This will allow a player to play by themselves against a computer controlled character
- This is a difficult task as we will need to create an autonomous character
- Point Value Proposal: 20 Points
- Inputs: none
- Outputs: VGA
- Processor Use: Significant processor use to generate its own inputs

Task 13: Knockback

- In the typical game of Smash Bros, damage affects your knockback, and you only die by flying too far away from the screen. All attacks accelerate you away from the attack, and this task involves modulating the amount you accelerate away by your current damage level to achieve one of the signature elements that distinguishes super smash bros from other fighting games.
- This task is moderately difficult as it requires modulating the parameters of the physics engine for certain specific moves.
- Point Value Proposal: 20 points
- Inputs: None
- Outputs: VGA
- Processor Use: Significant processor use for performing the physics-based calculations needed to achieve this effect.

Task 14: Items

- This task involves randomly generating items that the characters can use and interact with. It will require creating models for the items and defining their properties or how they affect the properties of characters
- This is a moderately difficult task since it requires potentially modifying many properties
- Point Value Proposal: 20 Points
- Inputs: none
- Outputs: VGA
- Processor Use: Minimal use

Task 15: 4 Player Smash

- This task involves expanding the original game to allow 4 players to play. It will require building 2 additional controller setups and expanding many previous tasks to account for the additional players.
- This task incorporates a large number of the previous tasks and is thus extremely difficult
- Point Value Proposal: 30 points
- Inputs: 2 Additional Controllers

- Outputs: VGA
- Processor Use: Moderate - Additional controls and movements must be processed

Task 16: Wireless Controller Communication

- This task will involve converting the controller system into a wireless system communicating with the DE2 via XBee or a similar technology.
- This task is extremely difficult since it requires re-learning XBee (or similar), integrating it with the existing controller, and debugging (very difficult based on ECE 110 experience).
- Point Value Proposal: 30 points
- Inputs: Controller
- Outputs: none
- Processor Use: none

Task 17: RGB LEDs

- Add RGB LEDs in various locations on the project for varied aesthetic purposes. Implement aesthetically pleasing patterns using the VGA. For example, we could place them near the speaker and light them with a specific color when the sound effect for a character gaining a powerup is played.
- This task is of intermediate difficulty.
- Point Value Proposal: 10 points
- Inputs: none
- Outputs: RGB LEDs
- Processor Use: Significant, for detection of triggering events in gameplay (use of branching instructions) and possibly for producing light patterns.

Task 18: Compiler

- We will write a compiler that takes very basic higher level code (using things like loops, if statements, etc.) and compiles it into 350 MIPS. The goal of this compiler is to greatly enhance the speed of programming our game, since the game is a fairly large programming task, especially in such a low level language.
- This task is extremely difficult.
- Point Value Proposal: 30 points
- Inputs: none
- Outputs: none
- Processor Use: none

Task 19: Final Smash

- This is the pinnacle of artistic creation, Michelangelo rolls in his grave thinking of such an amazing creation
- This task is beyond human comprehension. It is harder than the floor of the 350 lab and longer than the biggest python in the everglades.
- Point Value Proposal: 50 Points
- Inputs: Every brain cell in the universe
- Outputs: VGA and the entire set of multiverses in existence
- Processor Use: All of it, as well as IBM's largest supercomputer

Total Proposed Points: 400

Timeline:

Week 0

This Report

Week 1

Compiler (18), Hacked Controller (2), Display Characters (5), Music (10)

Week 2

Physics Engine (1), Advanced Stage (4), Scoring System (6), Sound Effects (9)

Week 3

Character Selection (7), Stage Selection (8), 4-player (15), NPC/AI (12)

Week 4

Knockback (13), Items (14), Final Smash (19), Controller Finishing (3), Amplification System (11), Wireless Controller (16), RGB LEDs (17)

NOTE: Since we proposed many more points than the required 120, we will not be implementing all tasks for each week on the timeline (which includes them all). We will choose a subset for each week.