Wetris: CPE 301 Final Project

Team Wetris (Team 42)

December 12, 2025

Contents

1 Project Description **3**

2 Cost Analysis for this First Time Build **3**

3 Component Details **4**
- 3.1 Arduino Mega 2560 . . . 4
- 3.2 LCD Display (ST7796S controller) . . . 5
- 3.3 Door Lock Actuator . . . 5
- 3.4 Fuse . . . 5
- 3.5 Potentiometer . . . 5
- 3.6 Buzzers, Resistors, and Capacitors . . . 5
- 3.7 Relay Module . . . 5
- 3.8 Optocouplers . . . 6

4 System Overview **6**
- 4.1 Hardware Architecture . . . 6
  - 4.1.1 Tetris Module . . . 7
  - 4.1.2 Music Module . . . 7
  - 4.1.3 Water Gun Module . . . 7
  - 4.1.4 Potentiometer Module . . . 8
- 4.2 Software Architecture . . . 8
- 4.3 Power . . . 8

5 Images **9**

6 Schematic Diagram **9**

7 Links **10**
- 7.1 Repository . . . 10
- 7.2 Documentation . . . 10
- 7.3 Demo Videos . . . 10

8 Our Team **11**
- 8.1 Gabriel Jordaan . . . 11
- 8.2 Team Member 2 . . . 11
- 8.3 Team Member 3 . . . 12

8.4 Team Member 4 12
8.5 Acknowledgments 13

1 Project Description

Wetris is a suspenseful and slightly nerve-racking spin-off of Tetris, the classic block-stacking game. The player races to clear enough lines to reach the target score before their stack reaches the top, and if they fail, they get splashed with water! To make it even better, the spring powered water gun is very loud and is a garenteed jump scare even if you know it's coming. It makes for a great party game, and would go well as an attraction in an arcade.

2 Cost Analysis for this First Time Build

This cost analysis includes tools that will not need to be repurchased for future builds, such as the multimeter and crimping tool. These tools were necessary for the successful completion of this first build, and their costs are included to provide a comprehensive overview of the initial investment required for the Wetris project. A production version of Wetris could be built for significantly less, since the tools are one time purchases, and many of the components were purchased in higher quantities than necessary and the cost for those components could be spread across multiple builds. The largest cost would still be the 3d printed casing, but even that could be reduced with bulk printing or alternative manufacturing methods such as injection molding. A cost analysis for a production version is not included here, but may later be developed if there is interest in pursuing that avenue.

Material	Total Price	Notes
3D printed casing	$263.52	UNR LIBRARIES TRANSACTION 201937 DLM
Arduino ATmega 2560	$54.02	Amazon
Multimeter	$34.63	Amazon
Crimping tool	$34.63	Amazon
LCD Display	$22.72	Amazon (Jorge also bought us a second one for testing)
Wire stripper	$20.56	ACE hardware
Door lock actuator	$17.05	Amazon
Relay Module	$14.42	Amazon
Syringes	$14.06	Amazon
12V power adapter	$14.06	Amazon
Extension cord	$12.98	Amazon
Caulk gun	$10.82	Newborn DC012 from ACE hardware
Pin connector for actuator	$10.27	Amazon
Electrical tape	$9.30	ACE hardware
Wire	$8.66	ACE hardware
Inline fuse holder	$7.13	ACE hardware
Buzzers	$6.49	Amazon
Sprinkler head	$5.40	ACE hardware
Fuses	$5.40	Amazon
Barrel jack splitter	$4.30	Amazon
Misc. we on hand	$0.00	Jumper wires, buttons, tools, resistors, breadboard, 9V battery, USB cable, optocouplers etc.
Epoxy	$0.00	Took a little from another project of Gabe's
Grand Total:	**$570.43**	

Table 1: Component cost breakdown for Wetris system

3 Component Details

This section provides detailed information about each major component used in the Wetris system, including their specifications and roles within the project.

3.1 Arduino Mega 2560

Central microcontroller unit (MCU) responsible for managing game logic, user inputs, and controlling peripherals. Full technical specifications can be found at:
`https://cdn.robotshop.com/media/a/ard/rb-ard-33/pdf/arduinomega2560datasheet.pdf` for the development board and
`https://www.alldatasheet.com/datasheet-pdf/download/107092/ATMEL/ATMEGA2560.html` for the microcontroller itself.

3.2 LCD Display (ST7796S controller)

Hosyond 4.0” TN Capacitive Touch Display
Display Controller: ST7796S driver IC
Touch Controller: FT6336U capacitive touch IC (not used in project)
Size: 4.0” diagonal, 320x480 resolution
Physical Dimensions: 60.88x108.0x14.80mm (WxHxD)
Interface: 4-wire SPI (display) + I2C (touch)
Power: 5V/3.3V compatible, 0.5W consumption (We used 5V)
Connectivity: 14P 2.54mm header + FPC connector
Additional: Micro SD slot, LED backlight (300 cd/m^2) (Both unused in project)

3.3 Door Lock Actuator

The door lock actuator used is intended for a Dodge Grand Caravan, and operates on 12V DC and draws 2A. In order to reverse the direction of the actuator, the polarity of the voltage applied to it must be reversed. It is energy efficient, only drawing current when moving, and has a built-in limit switch to stop current flow when fully extended or retracted. Full specifications could not be found past what the amazon page provided. Pins had to be experementally probed to determine their propper function. Two pins still have unknown function and are left unconnected. They are possibly for feedback. More information can be found at:
`https://www.amazon.com/dp/B0C43BNSQM?ref_=ppx_hzsearch_conn_dt_b_fed_asin_title_3&th=1`

3.4 Fuse

Standard 3A automotive blade fuse used to protect the water gun actuator circuit from overcurrent conditions.

3.5 Potentiometer

Standard 10k ohm rotary potentiometer used to adjust game speed and music tempo.

3.6 Buzzers, Resistors, and Capacitors

Standard piezoelectric buzzers used for audio output, along with 100 Ω resistors and 10 µF capacitors for signal conditioning and circuit protection.

3.7 Relay Module

Forward and Reverse Relay Module, 12V 10A Pre-Wired with LED Light, for Motor/Linear Actuator, Reversing Relay Module. More information can be found at:
`https://www.amazon.com/Weasch-Forward-Pre-Wired-Actuator-Reversing/dp/B0FJQV4316?pd_rd_w=4c3sO&content-id=amzn1.sym.97c08ff8-4f39-4f9e-b360-8dbdbec34ee6&pf_rd_p=97c08ff8-4f39-4f9e-b360-8dbdbec34ee6&pf_rd_r=YHHMAAYABW9QQTN9YD5D&pd_rd_wg=pirQI&pd_rd_r=114b95df-9ba3-4874-ba37-76236030403c&pd_rd_i=B0FJQV4316&psc=1&ref_=pd_basp_d_rpt_ba_s_1_t`

3.8 Optocouplers

4N35 optocouplers with a foreward voltage of 1.18 V and a maximum collector-emitter voltage of 30 V used to isolate the high current 12V actuator control circuit from the low voltage 5V logic of the Arduino. More information can be found at:
`https://www.alldatasheet.com/datasheet-pdf/view/98359/FAIRCHILD/4N35.html`

4 System Overview

For this project we designed a series of peripherals to interface with the microcontroller in the Arduino Mega 2560 development board, and connected them in a 3d printed casing. The design is highly modular, with each peripheral being controlled by a library, and being handled from a high level main loop. See code and schematic for most in depth explaination. The peripherals are:

- The Tetris module, consisting of just an LCD, three buttons, and the software to run tetris game logic
- The music module, with three buzzers playing the tetris theme song (Korobeiniki)
- The water gun module, which consists of a syringe, which is pushed by a spring powered plunger let back by a door lock actuator controlled by a relay module, which is in turn controlled by opto-couplers.
- The potentiometer module, which is used to adjust the speed of the music module and the Tetris module.

4.1 Hardware Architecture

The following table outlines the microcontroller resources allocated to each component of the Wetris system, and details the connections used:

Reserved Component	**Name of User**	**Purpose**
OC1A(pin11)+ Timer1	Gabe	PWM signal for music
OC3A(pin5) + Timer3	Gabe	PWM signal for music
OC4A(pin6) + Timer4	Gabe	PWM signal for music
Button 1(pin13)	Roman+Jorge+Gabe	Left Movement
Button 2(pin12)	Roman+Jorge+Gabe	Right Movement
Button 3(pin10)	Roman+Jorge+Gabe	Rotation Movement
LCD(pins 7-9)	Roman	LCD connections
LCD(Dig. Pins50-52)	Roman	LCD connections
LCD LED(pin2)	Roman	Backlight control
FWD Pin (22)	Son+Gabe	Drive actuator forward
REV Pin (23)	Son+Gabe	Drive actuator reverse

Table 2: Hardware resource allocation for Wetris system

4.1.1 Tetris Module

In order to reach the back of the casing where the MCU is mounted from the LCD and buttons in the front of the casing, we used 16 AWG wire to connect the buttons and dupont wires linked end to end to connect the LCD. The LCD pin connections in detail are as follows:

- Pin 2 → LED pin on the LCD
- Pin 4 → Rotate Piece Button
- Pin 8 → RESET pin on LCD
- Pin 9 → CD pin on LCD
- Pin 10 → CS pin on LCD
- Pin 12 → Right Button
- Pin 13 → Left Button
- Pin 50 → SDO
- Pin 51 → SDI
- Pin 52 → SCK

4.1.2 Music Module

The music module uses output compare pins OC1A (pin 11), OC3A (pin 5), and OC4A (pin 6) to output PWM signals to three separate piezo buzzers. Each buzzer is connected in series with a 100 ohm resistor to limit current and a 10 µF capacitor to smooth the signal. The code directly loads frequency and duration values into memory and iterates through them to play the song. This method is highly memory intensive, but allows for accurate timing and frequency control. The music was converted from midi format to c style arrays using a custom python script found in the repository.

4.1.3 Water Gun Module

In order to isolate the high current 12V power supply used to drive the water gun actuator from the 5V logic of the Arduino, we used a relay module controlled by opto-couplers. The relay module is powered by the 12V supply, and the opto-couplers are powered by the 5V supply from the Arduino. The opto-couplers are connected to digital pins 22 and 23 on the Arduino, which control the forward and reverse movement of the actuator respectively. When the game is over and the safety threshold has not been reached, the Arduino activates the relay in reverse to trigger the water gun mechanism. The foreward function is not used in normal operation, but is included for completeness. Very importantly, an inline fuse is included in the 12V power line to protect against short circuits.

4.1.4 Potentiometer Module

The potentiometer is our most simple peripheral, being connected to the 5V and GND pins on the Arduino, with the wiper connected to analog pin A0. The potentiometer is used to adjust the speed of the game and music modules, allowing the player to customize their experience. Also allowing their friends to set it to an impossible speed in the middle of a game for extra fun.

4.2 Software Architecture

The software is structured in a modular fashion, with each peripheral being handled by its own library. The main loop coordinates the interactions between the different modules, ensuring smooth gameplay and timely responses to user inputs. All modules had to be engineered to be non-blocking and work together. This presented a difficult engineering challenge that had to be overcome with many optimisations and plan-B methodologies. For example, the LCD tetris module had to render only when the game state changed, otherwise the music module would stutter as the updates took too much time. Clearing the screen with the LCD library took too long, so we had to implement our own function to only update the parts of the screen that changed, and now we only use the clearing function when absolutely necessary, such as at the start of the game or when a row is cleared.

4.3 Power

The Wetris system needs two power sources to operate correctly. The first is power to the Arduino, which is supplied through the barrel jack port via a 9V battery (the 9V is internally regulated to 5V by the Arduino). The second power source is a 12V power supply which powers the water gun actuator via a relay module. The relay module is controlled by the Arduino through opto-couplers to isolate the two power sources.

5 Images

These images showcase the physical implementation of the Wetris system, including the circuit and various stages of gameplay.

Circuit Photo Placeholder

Figure 1: Physical circuit implementation of Wetris system

6 Schematic Diagram

Below is a schematic diagram illustrating the electrical connections and components used in the Wetris system. There is a high level diagram that shows how the peripherals are connected, and a more detailed schematic for each peripheral.

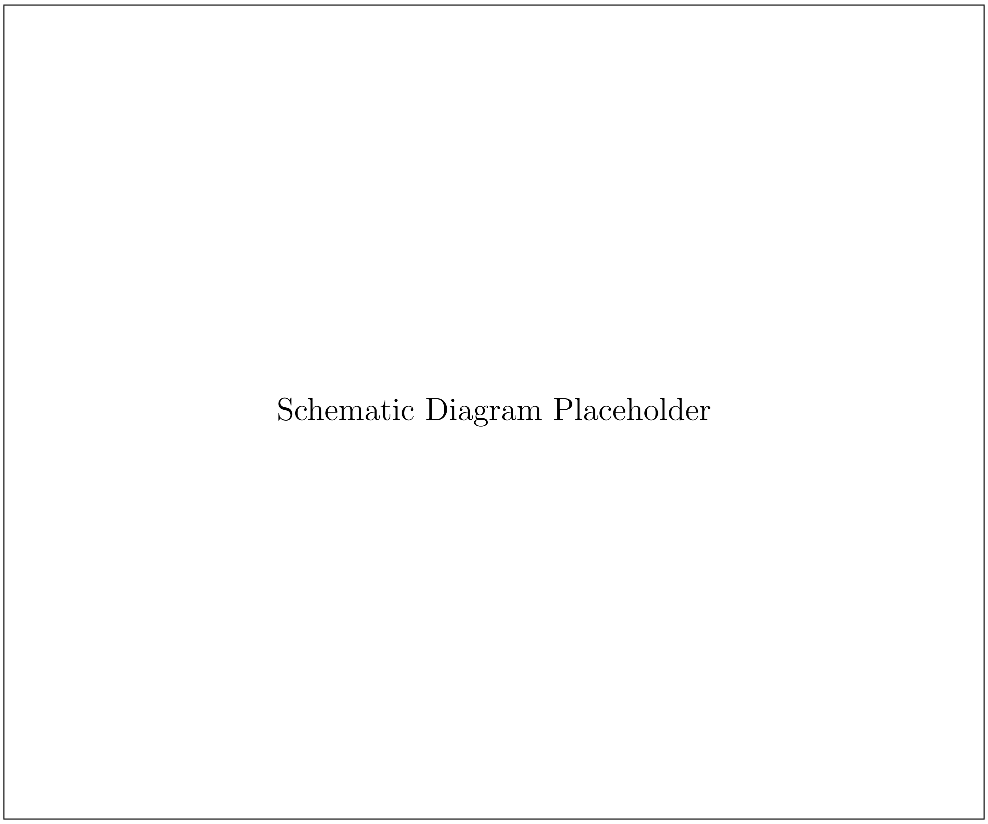

Figure 2: Electrical schematic of Wetris system

7 Links

7.1 Repository

GitHub Repository:
`https://github.com/ACertainArchangel/Wetris-CPE-301-Final-Project`

7.2 Documentation

Additional documentation and resources can be found in the project repository.

7.3 Demo Videos

Link to demo videos:
`https://drive.google.com/drive/folders/1Q4Bg4BSVfFWwpvJh6P31OyMYkDlfBsM4?usp=sharing`

8 Our Team

8.1 Gabriel Jordaan

Gabriel Jordaan

Major: Electrical Engineering
Email: gvanrijnjordaan@unr.edu

Role & Contributions:
Project manager, hardware developer, enclosure design and 3d printing, music module, main.cpp, python utilities, documentation, and final report.

About Me:
I am an electrical engineering student in the dual enrollment program at UNR, and as of fall 2025 I am 17 years old. I plan to graduate with my bachelors degree in 2027 and move on to a masters and PhD in robotics. Embedded systems and machine learning are my main interests in terms of engineering.

8.2 Team Member 2

Photo Here

[Member 2 Name]

Major: [Your Major]
Email: [email@unr.edu]

Role & Contributions:
[Describe your specific contributions to the project - hardware design, software modules, system integration, testing, etc.]

About Me:
[Brief professional bio highlighting your skills, interests, career goals, relevant coursework, internships, projects, and what makes you stand out to potential employers.]

8.3 Team Member 3

Photo
Here

[Member 3 Name]

Major: [Your Major]
Email: [email@unr.edu]

Role & Contributions:
[Describe your specific contributions to the project - hardware design, software modules, system integration, testing, etc.]

About Me:
[Brief professional bio highlighting your skills, interests, career goals, relevant coursework, internships, projects, and what makes you stand out to potential employers.]

8.4 Team Member 4

Photo
Here

[Member 4 Name]

Major: [Your Major]
Email: [email@unr.edu]

Role & Contributions:
[Describe your specific contributions to the project - hardware design, software modules, system integration, testing, etc.]

About Me: [Brief professional bio highlighting your skills, interests, career goals, relevant coursework, internships, projects, and what makes you stand out to potential employers.]

8.5 Acknowledgments

Thanks to our CPE 301 professor, Dr. Bashira Anima, for a great series of classes in fall 2025 and for being a good sport and letting us soak her with water!

We would also like to acknowledge the open-source libraries that made this project possible:

- LCDWIKI_SPI and LCDWIKI_GUI libraries - Created by the lcdwiki team (`https://github.com/lcdwiki/LCDWIKI_gui`), based on Adafruit GFX lib and Adafruit SPITFT lib, with init code from Rossum. Released under MIT license.

- Adafruit GFX and SPITFT libraries - For providing the foundational graphics and display functionality.

Thanks to Pastor David Minott for the idea of using a door lock actuator instead of a solenoid.

Thanks to Nikolai Nekrasov for writing the original Korobeiniki song in 1861. (He's dead and can't sue us for copyright infringement right?)