

# CarbonCraft: Visualizing Hydrocarbon Derivatives

Mohamed Eltahawy<sup>\*a</sup>, Yousef Fayad<sup>†b</sup> and Khaled Walied<sup>‡c</sup>

<sup>a,b,c</sup>El-Sadat STEM School

This manuscript was compiled on April 15, 2026, research supervised by Rabab Elshafeiy

**Abstract**—CarbonCraft is an open-source, 3D molecular modeling and visualization platform developed as a core component of the graduation chemistry project at El-Sadat STEM School. The software provides an interactive environment for the real-time assembly and analysis of hydrocarbons and their derivatives. Developed entirely in Python, CarbonCraft utilizes the Ursina engine for high-performance 3D rendering, while integrating the RDKit and PubChemPy libraries to handle cheminformatics tasks. The application features an intuitive “drag-and-drop” interface that allows users to manipulate atoms and functional groups, providing immediate visual feedback on the compound’s 3D geometry. The software autonomously generates SMILES formulas, identifies associated functional groups, and retrieves IUPAC nomenclature and chemical data via an API. By providing an open-source framework, CarbonCraft offers an easy solution for chemical education, helping students visualize hydrocarbon compounds while learning about them to enhance their learning experience.

**Keywords**—Hydrocarbon derivatives, CarbonCraft, Molecular visualization, Organic chemistry, Computational chemistry, Chemoinformatics

## 1. INTRODUCTION

CarbonCraft is a 3D molecule builder and visualizer designed specifically for organic hydrocarbon derivatives.

### 1.1. Project Overview

The system enables students to construct organic molecules interactively by adding atoms (e.g. Carbon, Halogens, Oxygen), modifying bonds (e.g. Single, double, triple) and forming cyclic structures, while simultaneously visualizing the resulting molecule in a dynamic 3D environment. In addition to structural visualization, CarbonCraft integrates cheminformatics tools to generate formal representations such as SMILES strings, functional groups, and IUPAC names, bridging the gap between intuitive molecular construction and formal chemical notation.

### 1.2. Objectives

The primary objective of this project was to develop an interactive and friendly application that allows students to both build and understand organic molecules in 3D space. Specifically, the project aimed to:

1. Provide a real-time 3D visualization of the organic compounds.
2. Interactive editing via a drag-and-drop dispenser to add atoms and bonds.
3. Automatically generate IUPAC Names and SMILES representation.
4. Serve as an educational tool for concepts such as isomerism and bond geometry.

### 1.3. Motivation

Understanding organic molecules is often challenging for students, incredibly when dealing with concepts such as isomerism, bond geometry and functional derivatives. The static diagrams or 2D sketches fail to fully capture and cover the spatial nature of molecular structure. CarbonCraft was motivated by the need of students for a more

interactive educational tool that allows users to directly frame and manipulate molecules and observe how structural changes affect geometry and IUPAC naming. By combining 3D visualization with automated chemical naming, the project helps students develop a deeper understanding of molecular behavior.

## 2. BACKGROUND RESEARCH

### 2.1. Hydrocarbon

A **hydrocarbon** is an organic chemical compound that is made up **exclusively** of only two types of atoms: **Carbon (C)** and **Hydrogen (H)**. They are the “parent” molecules of organic chemistry. All derivatives (like Alcohols, Esters, and Amines) are created by taking a basic hydrocarbon and replacing one or more hydrogen atoms with a different group (like  $-\text{OH}$  or  $-\text{COOH}$ ).

#### Key Characteristics of Hydrocarbons:

1. **Composition:** Only Carbon and Hydrogen. No Oxygen, Nitrogen, or Halogens.
2. **Source:** Primary components of fossil fuels, including petroleum (crude oil), natural gas, and coal.
3. **Polarity:** They are **non-polar**, meaning they do not dissolve in water (hydrophobic).
4. **Combustibility:** They are highly flammable and are most commonly used as fuels (like methane, propane, and gasoline).

### 2.2. Main Types of Hydrocarbons

Hydrocarbons are generally divided into two main categories:

#### 1. Aliphatic Hydrocarbons (Chains)

- **Alkanes:** Contain only single bonds ( $\text{C}-\text{C}$ ). They are “saturated” because they hold the maximum number of hydrogens (e.g., Methane, Ethane).
- **Alkenes:** Contain at least one double bond ( $\text{C}=\text{C}$ ). They are “unsaturated” (e.g., Ethene).
- **Alkynes:** Contain at least one triple bond ( $\text{C}\equiv\text{C}$ ) (e.g., Ethyne/Acetylene).

#### 2. Aromatic Hydrocarbons (Rings)

- These contain special stable ring structures, the most common being the **Benzene ring** ( $\text{C}_6\text{H}_6$ ). They often have distinct smells, hence their name, “aromatic”.

### 2.3. Laboratory Preparation Methods

In the laboratory, hydrocarbons are prepared using specific chemical reactions that either break down larger molecules or build them from smaller parts. The following are the most common laboratory methods for preparing each type:

#### 2.3.1. Aliphatic Hydrocarbons (Chains)

##### A. Alkanes (Saturated: Single Bonds)

The most common lab method is the **Decarboxylation of Carboxylic Acid Salts** (Soda-lime method).

- **Reaction:** Heating a sodium salt of a carboxylic acid with soda-lime (a mixture of  $\text{NaOH}$  and  $\text{CaO}$ ).
- **Chemical Equation:**  $\text{CH}_3\text{COONa}$  (Sodium Acetate) +  $\text{NaOH} \xrightarrow{\text{CaO}, \Delta} \text{CH}_4$  (Methane) +  $\text{Na}_2\text{CO}_3$

<sup>\*</sup>Mohamed.2523061@stemelsadat.moe.edu.eg

<sup>†</sup>yousef.2523066@stemelsadat.moe.edu.eg

<sup>‡</sup>Khaled.2523016@stemelsadat.moe.edu.eg

**i. Wurtz Reaction (Coupling Method)**

Used to prepare **higher alkanes** by joining two smaller alkyl groups.

- **Reaction:** Reacting an alkyl halide with metallic sodium in dry ether.
- **Equation:** 
$$2 \text{CH}_3\text{CH}_2\text{Br} + 2 \text{Na} \xrightarrow{\text{Dry Ether}} \text{CH}_3\text{CH}_2\text{CH}_2\text{CH}_3 \text{ (Butane)} + 2 \text{NaBr}$$

**ii. Kolbe's Electrolytic Method**

- **Reaction:** Electrolysis of a concentrated aqueous solution of a sodium or potassium salt of a carboxylic acid.
- **Equation at Anode:** 
$$2 \text{CH}_3\text{COONa} + 2 \text{H}_2\text{O} \xrightarrow{\text{Electrolysis}} \text{CH}_3-\text{CH}_3 \text{ (Ethane)} + 2 \text{CO}_2 + \text{H}_2 + 2 \text{NaOH}$$

**iii. From Grignard Reagent (High Purity)**

- **Reaction:** Adding water or alcohol to a Grignard Reagent (R-Mg-X).
- **Equation:** 
$$\text{CH}_3\text{CH}_2\text{MgBr} \text{ (Ethyl magnesium bromide)} + \text{H}_2\text{O} \longrightarrow \text{CH}_3\text{CH}_3 \text{ (Ethane)} + \text{Mg(OH)Br}$$

**B. Alkenes (Unsaturated: Double Bonds)**

The standard method is the **Dehydration of Alcohols**.

- **Reaction:** Heating an alcohol with concentrated acid (like  $\text{H}_2\text{SO}_4$ ).
- **Chemical Equation:** 
$$\text{CH}_3\text{CH}_2\text{OH} \text{ (Ethanol)} \xrightarrow{\text{conc. H}_2\text{SO}_4, 170^\circ\text{C}} \text{CH}_2=\text{CH}_2 \text{ (Ethene)} + \text{H}_2\text{O}$$

**i. Dehydrohalogenation of Alkyl Halides**

- **Reaction:** Heating an alkyl halide with **alcoholic** Potassium Hydroxide (KOH).
- **Equation:** 
$$\text{CH}_3\text{CH}_2\text{Cl} + \text{KOH}(\text{alc}) \xrightarrow{\Delta} \text{CH}_2=\text{CH}_2 \text{ (Ethene)} + \text{KCl} + \text{H}_2\text{O}$$

**ii. Dehalogenation of Vicinal Dihalides**

- **Reaction:** Reacting a dihalide with Zinc dust.
- **Equation:** 
$$\text{CH}_2\text{Br}-\text{CH}_2\text{Br} + \text{Zn} \longrightarrow \text{CH}_2=\text{CH}_2 + \text{ZnBr}_2$$

**C. Alkynes (Unsaturated: Triple Bonds)**

Preparation of **Ethyne (Acetylene)** from **Calcium Carbide**.

- **Reaction:** Adding water to calcium carbide.
- **Equation:** 
$$\text{CaC}_2 + 2 \text{H}_2\text{O} \longrightarrow \text{C}_2\text{H}_2 \text{ (Ethyne)} + \text{Ca(OH)}_2$$

**i. Dehydrohalogenation of Vicinal Dihalides**

- **Reaction:** Treating a vicinal dihalide with a very strong base like Sodamide ( $\text{NaNH}_2$ ).
- **Equation:** 
$$\text{CH}_2\text{Br}-\text{CH}_2\text{Br} + 2 \text{NaNH}_2 \longrightarrow \text{CH}\equiv\text{CH} \text{ (Ethyne)} + 2 \text{NaBr} + 2 \text{NH}_3$$

**2.3.2. Aromatic Hydrocarbons (Rings)****Method A: Decarboxylation of Sodium Benzoate**

- **Equation:** 
$$\text{C}_6\text{H}_5\text{COONa} \text{ (Sodium Benzoate)} + \text{NaOH} \xrightarrow{\Delta} \text{C}_6\text{H}_6 \text{ (Benzene)} + \text{Na}_2\text{CO}_3$$

**Method B: Reduction of Phenol**

- **Equation:** 
$$\text{C}_6\text{H}_5\text{OH} \text{ (Phenol)} + \text{Zn} \xrightarrow{\Delta} \text{C}_6\text{H}_6 \text{ (Benzene)} + \text{ZnO}$$

**2.4. Hydrocarbon Derivative**

A derivative is derived from a hydrocarbon by **replacing** one or more hydrogen atoms with an atom or group of atoms of **other elements** (Oxygen, Nitrogen, Chlorine, Sulfur, etc.).

**The Basic Concept: Substitution**

To create a derivative, one performs a chemical "swap":

- **Hydrocarbon:** Contains only C and H. Formula: R-H.
- **Derivative:** Contains CH and at least one **Functional Group**. Formula: R-Z.

**Table 1.** Comparison: Ethane vs. Derivatives

Molecule	Type	Formula	Change	Behavior
Ethane	Hydrocarbon	$\text{C}_2\text{H}_6$	Pure C & H	Flammable gas
Ethanol	Derivative	$\text{C}_2\text{H}_5\text{OH}$	Replaced H w/ -OH	Liquid, antiseptic
Chloroethane	Derivative	$\text{C}_2\text{H}_5\text{Cl}$	Replaced H w/ -Cl	Industrial refrigerant
Ethanoic Acid	Derivative	$\text{C}_2\text{H}_3\text{COOH}$	Replaced H w/ -COOH	Sour taste, acidic

**Why do we care about them?**

Hydrocarbon derivatives make up medicines (Aspirin), food (Sugars, vinegar), plastics (PVC, Nylon), and smells (Esters).

**2.5. Function Groups**

A Functional Group is a specific atom or group responsible for the characteristic chemical reactions. It is the "tool" or "engine" attached to the carbon "skeleton."

**Table 2.** Table 1: Alcohols (R-OH)

C	IUPAC Name	Condensed Formula	Reaction
C2	Ethanol	$\text{CH}_3\text{CH}_2\text{OH}$	$\longrightarrow \longrightarrow$ Ethanal (Oxid.)
C3	2-Propanol	$\text{CH}_3\text{CH}(\text{OH})\text{CH}_3$	$\longrightarrow \longrightarrow$ Propanone (Oxid.)
C4	2-Methyl-2-propanol	$(\text{CH}_3)_3\text{COH}$	Resistant to Oxid.
C5	1-Pentanol	$\text{CH}_3(\text{CH}_2)_4\text{OH}$	$\longrightarrow \longrightarrow$ Pentanoic Acid
C6	3-Hexanol	$\text{CH}_3\text{CH}_2\text{CH}(\text{OH})(\text{CH}_2)_2\text{CH}_3$	$\longrightarrow \longrightarrow$ 3-Hexanone
C7	1-Heptanol	$\text{CH}_3(\text{CH}_2)_6\text{OH}$	$\longrightarrow \longrightarrow$ Heptyl Ethanoate
C8	2-Octanol	$\text{CH}_3\text{CH}(\text{OH})(\text{CH}_2)_5\text{CH}_3$	$\longrightarrow \longrightarrow$ 2-Octanone
C9	1-Nonanol	$\text{CH}_3(\text{CH}_2)_8\text{OH}$	$\longrightarrow \longrightarrow$ Nonanal
C10	2-Methyl-2-nonanol	$\text{CH}_3(\text{CH}_2)_6\text{C}(\text{OH})(\text{CH}_3)_2$	No reaction

**Table 3.** Table 2: Halocarbons (R-X)

C	IUPAC Name	Halogen	Reaction
C2	Chloroethane	Chlorine	$\longrightarrow \longrightarrow$ Ethanol
C3	2-Bromopropane	Bromine	$\longrightarrow \longrightarrow$ Propene
C4	1-Iodobutane	Iodine	$\longrightarrow \longrightarrow$ 1-Butanol
C5	1-Chloropentane	Chlorine	Substitution
C6	2-Bromohexane	Bromine	Elimination
C7	1-Fluoroheptane	Fluorine	Rare/Stable
C8	1-Chlorooctane	Chlorine	Substitution
C9	2-Iodononane	Iodine	Substitution
C10	1-Bromodecane	Bromine	Grignard Formation

**Table 4.** Table 3: Aldehydes (R-CHO)

C	IUPAC Name	Property	Reaction
C2	Ethanal	Pungent	Oxidation to Acid
C3	Propanal	Fruity	$\longrightarrow \longrightarrow$ 1-Propanol (Red.)
C4	Butanal	Strong smell	Oxidation
C5	Pentanal	Flavoring	Oxidation
C6	Hexanal	Grass-like	Reduction
C7	Heptanal	Oily scent	Reduction
C8	Octanal	Citrus scent	Oxidation
C9	Nonanal	Rose scent	Reduction
C10	Decanal	Orange peel	Oxidation

**Table 5.** Table 4: Ketones (R-CO-R)

C	IUPAC Name	Position	Reaction
C3	Propanone	2-position	→ → 2-Propanol
C4	Butanone	2-position	→ → 2-Butanol
C5	3-Pentanone	3-position	→ → 3-Pentanol
C6	2-Hexanone	2-position	Reduction
C7	4-Heptanone	4-position	Reduction
C8	2-Octanone	2-position	Reduction
C9	3-Nonanone	3-position	Reduction
C10	2-Decanone	2-position	Reduction

**Table 6.** Table 5: Carboxylic Acids (R-COOH)

C	IUPAC Name	Common Name	Reaction
C2	Ethanoic Acid	Acetic Acid	Ethanol → Ester
C3	Propanoic Acid	Propionic Acid	Salt formation
C4	Butanoic Acid	Butyric Acid	Amine → Amide
C5	Pentanoic Acid	Valeric Acid	Esterification
C6	Hexanoic Acid	Caproic Acid	Esterification
C7	Heptanoic Acid	Enanthic Acid	→ → 1-Heptanol
C8	Octanoic Acid	Caprylic Acid	Neutralization
C9	Nonanoic Acid	Pelargonic Acid	Herbicide use
C10	Decanoic Acid	Capric Acid	Soap production

**Table 7.** Table 6: Esters (R-COOR)

C	IUPAC Name	Parts	Reaction/Scent
C2	Methyl Methanoate	Acid:1, Alc:1	Acid + Alc
C3	Methyl Ethanoate	Acid:2, Alc:1	Hydrolysis
C4	Ethyl Ethanoate	Acid:2, Alc:2	Solvent use
C5	Propyl Ethanoate	Acid:2, Alc:3	Pear scent
C6	Ethyl Butanoate	Acid:4, Alc:2	Pineapple scent
C7	Pentyl Ethanoate	Acid:2, Alc:5	Banana scent
C8	Octyl Ethanoate	Acid:2, Alc:8	Orange scent
C9	Pentyl Butanoate	Acid:4, Alc:5	Apricot scent
C10	Pentyl Pentanoate	Acid:5, Alc:5	Apple scent

**Table 8.** Table 7: Ethers (R-O-R)

C	IUPAC Name	Type	Reaction/Use
C2	Dimethyl ether	Symmetrical	Gas at Room Temp
C3	Methoxyethane	Asymmetrical	Solvent
C4	Diethyl ether	Symmetrical	Anesthetic (old)
C5	Methyl t-butyl ether	Branched	Fuel additive
C6	Dipropyl ether	Symmetrical	Solvent
C7	Methoxyhexane	Asymmetrical	Industrial solvent
C8	Dibutyl ether	Symmetrical	High boiling point
C9	Ethoxyheptane	Asymmetrical	Chemical intermediate
C10	Dipentyl ether	Symmetrical	Stable/Non-reactive

**Table 9.** Table 8: Amines (R-NH<sub>2</sub>, R<sub>2</sub>NH, R<sub>3</sub>N)

C	IUPAC Name	Class	Reaction
C2	Ethanamine	1°	Acid → Salt
C3	1-Propanamine	1°	Basic nature
C4	Diethylamine	2°	N-Substitution
C5	N-Methylbutanamine	2°	Reaction with HCl
C6	Triethylamine	3°	Catalyst in Lab
C7	1-Heptanamine	1°	Industrial Base
C8	Dibutylamine	2°	Corrosion Inhibitor
C9	1-Nonanamine	1°	Surfactant
C10	1-Decanamine	1°	Solid at RT

**Table 10.** Table 9: Amides (R-CONH<sub>2</sub>)

C	IUPAC Name	Property	Reaction
C2	Ethanamide	Solid	+ Ammonia
C3	Propanamide	Polar	Hydrolysis
C4	Butanamide	Stable	Hydrolysis
C5	Pentanamide	Intermediate	Hydrolysis
C6	Hexanamide	Repellent	Red. → → Amine
C7	Heptanamide	Waxy solid	Hydrolysis
C8	Octanamide	Detergent part	Hydrolysis
C9	Nonanamide	Specialty chem	Hydrolysis
C10	Decanamide	Lipid-like	Red. → → Amine

## 2.6. Molecular Representations and Theoretical Constraints

To accurately translate chemical entities into a digital environment, CarbonCraft utilizes several layers of representation, ranging from alphanumeric strings to spatial embeddings based on geometric theory.

### SMILES (Simplified Molecular Input Line Entry System)

The SMILES notation is a chemical “language” that compresses a 3D molecular graph into a single line of text. In this system, atoms are represented by their chemical symbols, and bonds are represented by specific characters (e.g., = for double bonds, # for triple bonds). This representation is vital for the software’s backend, as it allows for the rapid computation of molecular weight and the searching of chemical databases. For example, Ethanol is represented as CCO, while Benzene is represented as c1ccccc1.

### IUPAC Nomenclature

The International Union of Pure and Applied Chemistry (IUPAC) naming system provides a systematic method for naming organic compounds. The name is derived from the longest continuous carbon chain (the “parent”) and modified by prefixes and suffixes that denote the presence of specific functional groups (e.g., *-one* for ketones, *-al* for aldehydes). CarbonCraft retrieves this data to ensure that users can connect the visual 3D model with the formal nomenclature required in scientific communication.

### 3D Geometry and Computational Embedding

The spatial arrangement of atoms in CarbonCraft is governed by **VSEPR (Valence Shell Electron Pair Repulsion) Theory**, which dictates that electron pairs around a central atom will position themselves as far apart as possible to minimize repulsion. This results in specific bond angles (e.g., 109.5° for tetrahedral Carbon).

In the software, these angles are achieved through 3D Embedding. Using the **RDKit** library, the software calculates a “force-field” that simulates these physical repulsions, placing atoms in their most energetically stable 3D conformation. This allows students to visualize the difference between planar molecules (like Ethene) and non-planar molecules (like Cyclohexane).

### Fundamental Valency and Bonding Rules

The “grammar” of organic chemistry is defined by **Valency**; the measure of an atom’s combining power with other atoms. To prevent the creation of chemically impossible structures, the software enforces strict valence limits. These limits ensure that every “drag-and-drop” action adheres to the octet rule and the specific bonding patterns of the elements provided in the software.

The following table outlines the valency constraints implemented within the CarbonCraft logic engine:

**Table 11.** Logic Engine Valency Constraints

Element	Symbol	At. No.	Valency	Common Bonding Pattern
Carbon	C	6	4	Tetravalent (up to 4 bonds)
Nitrogen	N	7	3	Trivalent (typically 3 bonds, 1 lone pair)
Phosphorus	P	15	3 or 5	Trivalent or Pentavalent
Oxygen	O	8	2	Divalent (typically 2 bonds)
Sulfur	S	16	2, 4, or 6	Commonly Divalent in organic chains
Hydrogen	H	1	1	Monovalent (1 sigma bond)
Fluorine	F	9	1	Monovalent (Halogen)
Chlorine	Cl	17	1	Monovalent (Halogen)
Bromine	Br	35	1	Monovalent (Halogen)
Iodine	I	53	1	Monovalent (Halogen)

### 3. METHODOLOGY

CarbonCraft is a hybrid computational program that combines 3D visualization with cheminformatics-based validation to ensure that constructed molecular structures adhere to chemical principles. The visualization layer is made via Ursina Engine, enabling interactive rendering and plotting of molecular geometries.

Chemical structure representation and validation are handled using RDKit, which provides robust tools for atom and bond management, valency and bond angles. Besides, PubChemPy is used to get back standardized chemical information, IUPAC nomenclature, based on generated SMILES strings. This integration validates the generated structures.

#### 3.1. Atom Management

The `add_group_to_atom` function is responsible for synchronization of the 3D representation and the chemical model of an atom. When a user adds an atom, the system creates a 3D sphere while inserting a corresponding atom into the RDKit molecular structure. Each atom is assigned an identifier that directly corresponds to its index within the RDKit molecule. This mapping ensures consistency between the rendered model and the chemical data. Code 1 shows a Python example.

```
def add_group_to_atom(target_idx, group):
    global current_rwmol
    backup = Chem.RWMol(current_rwmol)
    try:
        if group in ['C', 'O', 'N', 'S', 'P', 'F', 'Cl', 'Br',
                    ↪ 'I']:
            atomic_num = Chem.GetPeriodicTable().
            ↪ GetAtomicNumber(group)
            new_idx = current_rwmol.AddAtom(Chem.Atom(
            ↪ atomic_num))
            current_rwmol.AddBond(target_idx, new_idx, Chem.
            ↪ BondType.SINGLE)

        if not try_update(f"Added {group}!"):
            current_rwmol = backup
    except:
        current_rwmol = backup
```

Code 1. Python code example.

#### 3.2. Bond creation and Cycling bond

Chemical bonds in organic molecules can exist as single, double, or triple bonds. To support this, a bond cycling mechanism was implemented. When a student clicks on a bond, the system recaptures its current type from the RDKit molecule and updates it.

The function ensures that bond transitions follow the order:

**Single** → **Double** → **Triple**, while also preventing invalid valency that would violate chemical principles. Code 2 shows a Python example.

```
class InteractiveBond(Entity):
    def __init__(self, idx1, idx2, current_order, **kwargs):
        super().__init__(**kwargs)
        self.idx1 = idx1
        self.idx2 = idx2
        self.order = current_order
        next_order = "Double" if current_order == 1 else "
        ↪ Triple" if current_order == 2 else "Single"
        self.tooltip = Tooltip(f"Click → Make {next_order}
        ↪ Bond")

    def on_click(self):
        global current_rwmol
        backup = Chem.RWMol(current_rwmol)
```

```
bond = current_rwmol.GetBondBetweenAtoms(self.idx1,
↪ self.idx2)

new_t = Chem.BondType.DOUBLE if self.order == 1 else
↪ Chem.BondType.TRIPLE if self.order == 2 else Chem.
↪ BondType.SINGLE
bond.SetBondType(new_t)

name = "Double" if self.order == 1 else "Triple" if
↪ self.order == 2 else "Single"
if not try_update(f"Bond changed to {name}!"):
    current_rwmol = backup
```

Code 2. Python code example.

#### 3.3. Valency Protection

To maintain chemical correctness, the system enforces valency rules, such as carbon atoms, which cannot exceed four bonds. Before visualizing any structural modification, the molecule is validated via RDKit's built-in sanitization functions. Code 3 shows a Python example.

```
iupac_text.text = " IUPAC: Offline/Unknown"

def try_update(success_msg=""):
    global current_rwmol
    try:
        Chem.SanitizeMol(current_rwmol)
        render_molecule()

        smiles = Chem.MolToSmiles(current_rwmol)
        smiles_text.text = f"SMILES: {smiles}"
        functional_groups_text.text =
        ↪ format_functional_groups(detect_functional_groups(
        ↪ current_rwmol.GetMol()))
        iupac_text.text = " IUPAC: Computing..."
        threading.Thread(target=fetch_iupac, args=(smiles,),
        ↪ daemon=True).start()

        if success_msg: show_msg(success_msg)
        return True
    except Exception as e:
        show_msg("INVALID VALENCY! (Blocked)", is_error=True)
```

Code 3. Python code example.

#### 3.4. Data Retrieval

Once a molecule is constructed, the system converts it into a text representation. First, the RDKit molecule is transformed into a SMILES string. This string is then used as a query to get back chemical information (IUPAC names) from the PubChem database via PubChemPy.

```
def show_msg(msg, is_error=False):
    target = error_msg if is_error else action_msg
    target.text = msg
    target.color = color.red if is_error else color.cyan
    target.animate_color(color.clear, duration=2.5, delay
    ↪ =0.5)

def fetch_iupac(smiles):
    try:
        compounds = pcp.get_compounds(smiles, 'smiles')
        name = compounds[0].iupac_name if compounds else "
        ↪ Unknown Compound"
        iupac_text.text = f" IUPAC: {name}"
```

Code 4. Python code example.

```
def fetch_iupac(smiles):
    try:
        compounds = pcp.get_compounds(smiles, 'smiles')
```

```

name = compounds[0].iupac_name if compounds else "
↳ Unknown Compound"
iupac_text.text = f" IUPAC: {name}"
except:
iupac_text.text = " IUPAC: Offline/Unknown"

```

Code 5. Python code example.

## 4. RESULTS

CarbonCraft was able to visualize clean 3D molecules, while correctly fetching their IUPAC nomenclature and identifying associated functional groups in real-time. As demonstrated in the software interface captures, the system successfully handles simple structures like ethers (Fig. 1) and halides (Fig. 3), as well as complex multi-functional compounds containing amines, alkenes, and carboxylic acids (Fig. 2 and Fig. 4).

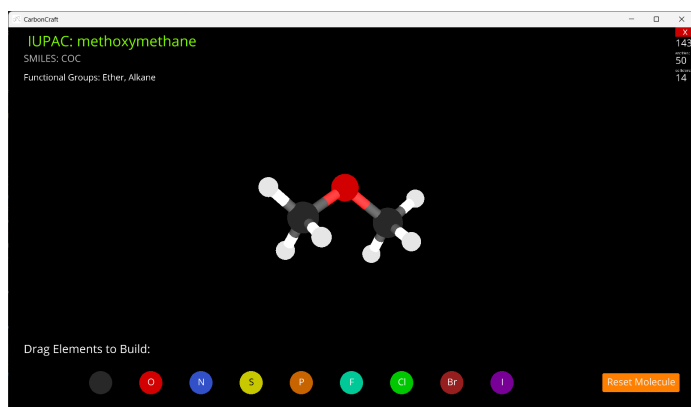


Figure 1. Visualization of methoxymethane (ether) showing SMILES and group detection.

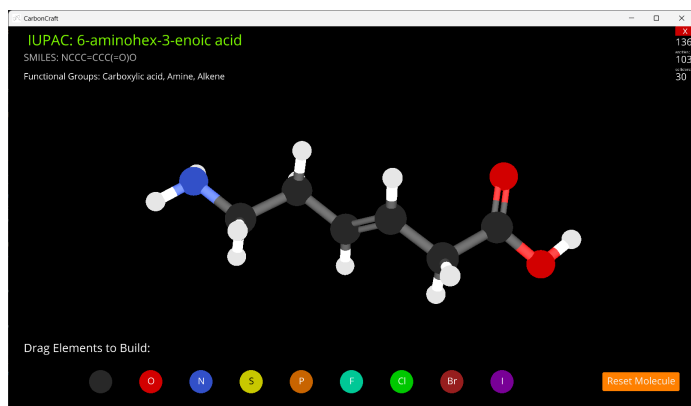


Figure 2. Complex structure: 6-aminohex-3-enoic acid featuring a double bond and multiple functional groups.

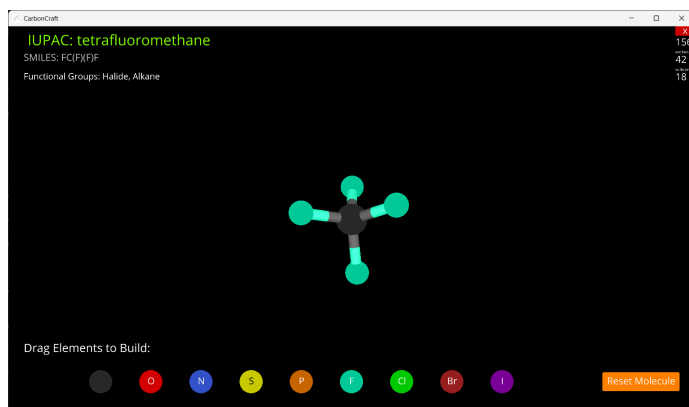


Figure 3. Tetrafluoromethane demonstrating accurate tetrahedral geometry based on VSEPR theory.

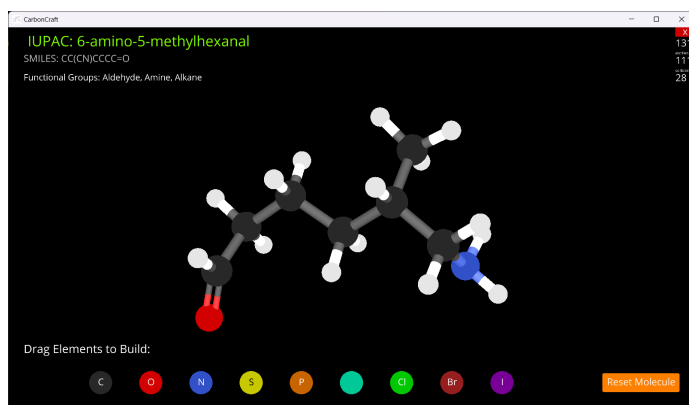


Figure 4. Visualization of 6-amino-5-methylhexanal showing branched chain logic.

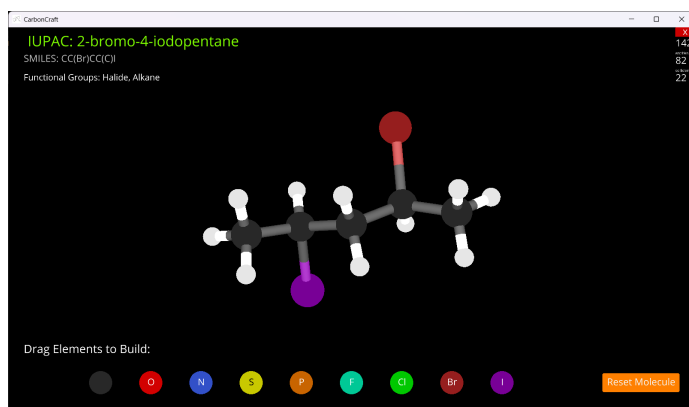


Figure 5. 2-bromo-4-iodopentane demonstrating the handling of multiple different halogens.

## 5. ANALYSIS AND DISCUSSION

### 5.1. Technical Performance and Real-Time Interaction

As highlighted in the project objectives, providing a “real-time” experience was a primary technical goal. During testing, **CarbonCraft** maintained a consistent **140 FPS** (Frames Per Second) on standard laptop hardware. This high performance is attributed to the use of procedural mesh generation within the Ursina engine, which avoids the overhead of loading complex external 3D files. The camera’s orbit and zoom functionality allow for seamless spatial navigation, solving the “motivation” problem described in the introduction—specifically, the failure of 2D sketches to capture the spatial nature of molecules.

## 5.2. Validation of Chemical Logic and Valency

The most significant achievement of the software is the enforcement of the Valency Constraints outlined in Table ???. By integrating RDKit's sanitization algorithms, the software acts as a "corrective tutor." For example:

- If a user attempts to add a fifth bond to a Carbon atom, the software prevents the update.
- The transition from a pure hydrocarbon to a derivative (as discussed in Section ??) is handled dynamically; when a halogen or oxygen is "swapped" in, the 3D embedding recalculates bond angles based on **VSEPR theory** to ensure geometric accuracy.

## 5.3. Analysis of Functional Group Recognition

The results shown in the screenshots (Fig. 1 through Fig. 5) demonstrate the software's ability to correctly identify complex structures like *6-aminohex-3-enoic acid*. By comparing the real-time output to the Functional Group Tables (Table ?? through Table ?? in the Background Research), we verified that the software accurately detects:

1. **Terminal Groups:** Such as the Aldehydes and Carboxylic acids.
2. **Internal Groups:** Such as the Ketones and Ethers.

The use of **asynchronous threading** ensures that while the IUPAC name is being retrieved from the PubChem API, the user interface remains responsive, allowing for continuous rotation and inspection of the 3D model.

## 5.4. Limitations and STEM Context

While the software is highly effective for educational purposes at **El-Sadat STEM School**, certain limitations exist. Currently, the IUPAC retrieval requires an active internet connection. Furthermore, while the software identifies functional groups, it does not yet visualize lone pairs of electrons, which are central to VSEPR theory. However, being **open-source**, these features represent opportunities for future student cohorts to expand the codebase.

## 6. CONCLUSION

CarbonCraft successfully achieves the objectives set forth for this graduation project at **El-Sadat STEM School**. By bridging the gap between theoretical organic chemistry and computational visualization, the platform provides students with a powerful tool to explore the world of hydrocarbons and their derivatives.

The project demonstrates three key successes:

1. **Integration:** Successfully combining high-level chemistry libraries (RDKit) with a modern 3D game engine (Ursina) to create a "digital laboratory."
2. **Accuracy:** Enforcing strict chemical valency and 3D geometric rules that traditional 2D diagrams cannot replicate.
3. **Educational Value:** Automating the generation of SMILES strings and IUPAC nomenclature, which helps students verify their knowledge and understand the mathematical-to-structural relationship in chemistry.

In conclusion, **CarbonCraft** is not just a visualizer, but an interactive educational framework. Its open-source nature ensures that it will remain a lasting resource for the STEM community, encouraging future students to continue developing tools that make complex scientific concepts accessible through technology.

## REFERENCES

- Aires-de-Sousa, J. (2023). *GUIDEMOL: A Python graphical user interface for molecular descriptors based on RDKit*. *Molecular Informatics*, 43(1). <https://doi.org/10.1002/minf.202300190>
- Ashenhurst, J. (n.d.). *Master Organic Chemistry*. Retrieved from <https://www.masterorganicchemistry.com/>
- Galati, S., Di Stefano, M., Macchia, M., Poli, G., & Tuccinardi, T. (2023). *MolBook UNIFI—Create, manage, analyze, and share your chemical data for free*. *Journal of Chemical Information and Modeling*, 63(13), 3977–3982. <https://doi.org/10.1021/acs.jcim.3c00278>
- International Union of Pure and Applied Chemistry (IUPAC). (2013). *Nomenclature of organic chemistry: IUPAC recommendations and preferred names (Blue Book)*. <https://iupac.qmul.ac.uk/BlueBook/>
- Landrum, G., Tosco, P., Kelly, B., et al. (2015). *RDKit: Open-source cheminformatics software* [Software]. Zenodo. <https://doi.org/10.5281/zenodo.591637>
- National Center for Biotechnology Information (NCBI). (n.d.). *PubChem database*. <https://pubchem.ncbi.nlm.nih.gov/>
- Open Source Initiative. (n.d.). *Open source definition*. <https://opensource.org/definition>
- Pocock, P. (n.d.). *Ursina Engine* [Software]. Retrieved from <https://www.ursinaengine.org/>
- PubChem. (n.d.). *Hydrocarbons*. Retrieved from <https://pubchem.ncbi.nlm.nih.gov/#query=hydrocarbons&tab=compound>
- Python Software Foundation. (2025). *Python (Version 3.12)* [Software]. <https://www.python.org/>
- Roucairol, M., Georgiou, A., Cazenave, T., Prischi, F., & Pardo, O. (2024). *DrugSynthMC: An atom-based generation of drug-like molecules with Monte Carlo search*. *ChemRxiv*. <https://doi.org/10.26434/chemrxiv.15001545.v1>
- Swain, M. (2025). *PubChemPy: Python wrapper for the PubChem PUG REST API* [Software]. <https://pypi.org/project/pubchempy/>
- Weininger, D. (1988). SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28(1), 31–36. <https://doi.org/10.1021/ci00057a005>