VGMech User Manual

Table of Contents

Introduction

Welcome to the user manual for VGMech! This document is intended to provide you with all the information you need to get started with and use the website effectively.

What is VGMech?

The Visual Game Mechanics website, abbreviated as VGMech, offers a visually-driven approach to learning game development, focusing particularly on game mechanics. Through interactive elements such as minigames that demonstrate common game mechanics alongside their code implementations, VGMech serves as a beginner-friendly platform for aspiring game developers to enter the field.

This user manual provides an in-depth overview of all the features available on VGMech. We are confident that you'll find this Visual Game Mechanics website invaluable for your upcoming journey into game development. Let's begin!

User System Requirements

Device - VGMech website utilizes embedded Unity WebGL interfaces to demonstrate each game mechanic alongside the minigames. These interfaces are developed to accept key inputs through which are reflected with the respective game logic inside the Unity interface. As such, the website's interactive features are only limited to be used in devices having keyboard functionalities.

Web Browser - As per mentioned, the system interactive functionalities relies on the integration of Unity WebGL. In relation, the system requirements related to Unity WebGl users must be met upon the use of VGMech's Learn system.

Desktop Browser	Desktop Platforms
Google Chrome	Windows, macOS, Linux
Mozilla Firefox	Windows, macOS, Linux
Apple Safari	macOS
Microsoft Edge	Windows, macOS, Linux

Notes (Referenced towards Unity WebGL Manual):

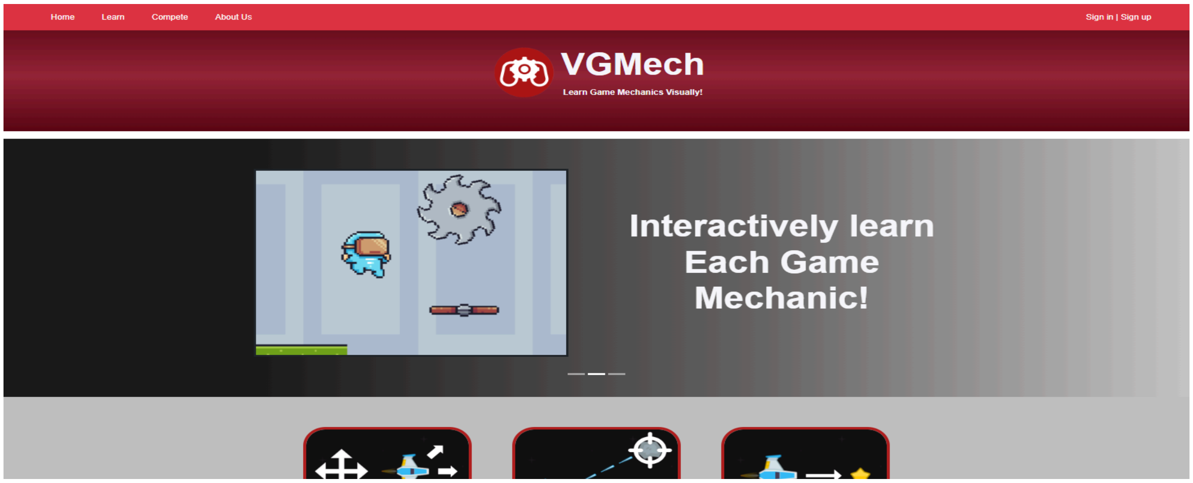
- Unity WebGL also supports the latest version of the Chromium-based Edge browser.
- Apple Safari doesn't support WebGL 2 in versions before Safari 15.
- Apple Safari doesn't support IndexedDB for content running in an iFrame.
- On Linux, you might have to install Advanced Audio Coding (AAC) codec support via a package manager (for example, the GStreamer package).

Operating System: VGMech’s content can generally run on any operating system that supports compatible web browsers for Unity WebGL.

Internet Connection: An active internet connection is required to load and run Unity WebGL content for the Learn Page Module since the content is hosted online and streamed to the user's web browser.

Using VGMech

The main areas of the VGMech are labeled below and are described in detail on the following pages.


Home
Learn
Compete
About Us
Sign in | Sign up
VGMech
Learn Game Mechanics Visually!
Interactively learn Each Game Mechanic!

- Home
- Learn
- Compete
- About Us
- Sign in | Sign up

Home

The Home section of VGMech typically serves as the main landing page or starting point for visitors providing overview contents of the website, purpose, and navigation options.

VGMech
Learn Game Mechanics Visually!
Collaboratively learn with other learners!

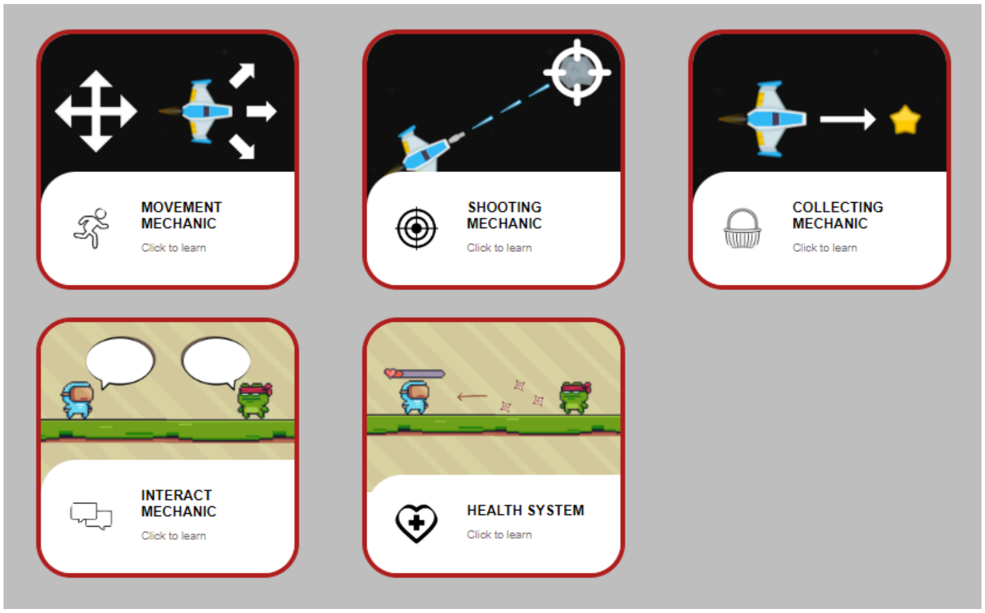
Learn

The Learn section of VGMech comprises of multiple game mechanics that includes:

- Movement Mechanic
- Shooting Mechanic
- Collecting Mechanic
- Interact Mechanic
- Health System
- More to be added soon

To use the Learn module:
Step 1: Go to the learn section of the homepage.
Step 2: Click the desired game mechanic to learn.
Step 3: To make use of the interactive demonstration, wait for Unity to load and click on the interactive window.
Step 4: Using the mentioned interactive controls, interact with the display and witness the mechanic demonstration.
Step 5: Learn how to implement the mechanic programmatically with the sample coding implementation.
Step 6: You may collaborate and converse with other learners through the comment section NOTE: Users must sign in/ sign up first to enable commenting.

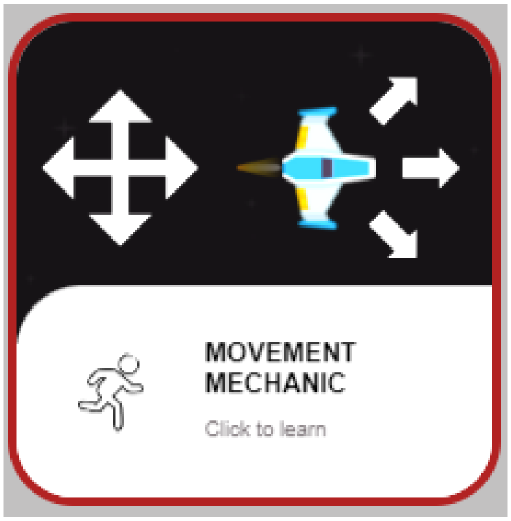
Below are detailed demonstrations

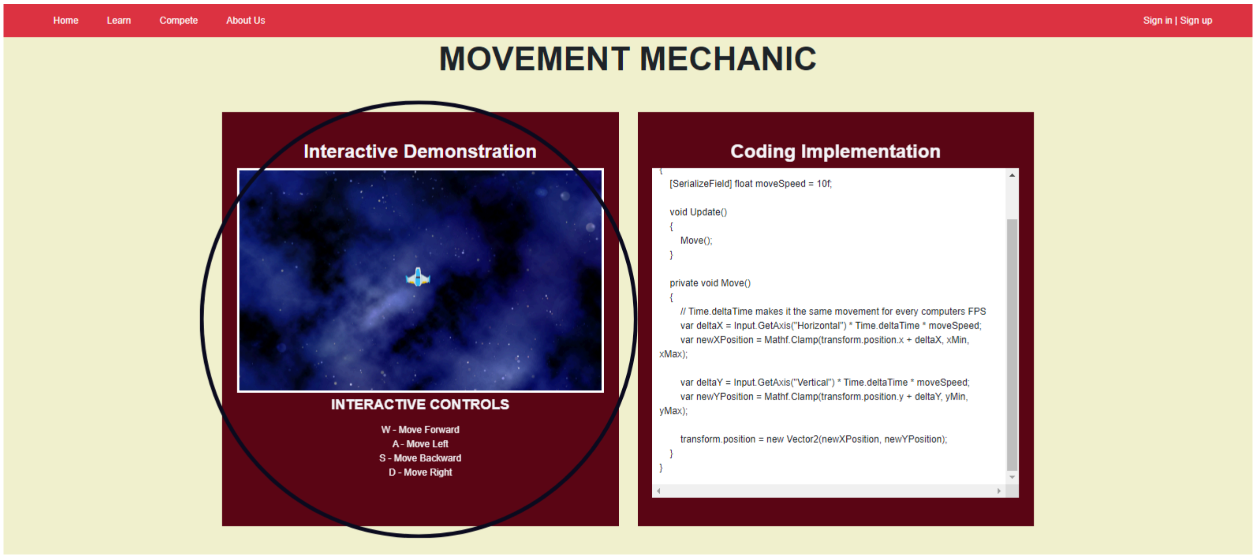


Movement Mechanic

The movement mechanic aspect constitutes a gaming element centered on the manipulation and control of the player's motion within the game's framework.

When the user clicks the movement mechanic card, the user will be transported to a webpage where the game mechanic will present its interactive demonstration, coding implementation, and its description.





MOVEMENT MECHANIC

Interactive Demonstration

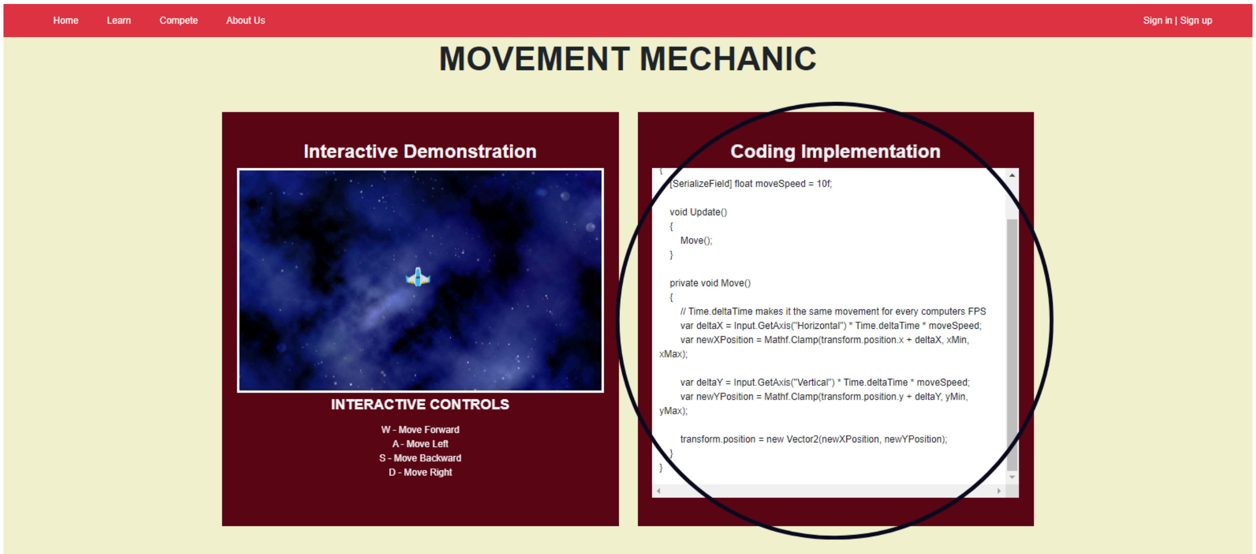
INTERACTIVE CONTROLS

W - Move Forward
A - Move Left
S - Move Backward
D - Move Right

Coding Implementation

```
{
    [SerializeField] float moveSpeed = 10f;

    void Update()
    {
        Move();
    }

    private void Move()
    {
        // Time.deltaTime makes it the same movement for every computers FPS
        var deltaX = Input.GetAxis("Horizontal") * Time.deltaTime * moveSpeed;
        var newXPosition = Mathf.Clamp(transform.position.x + deltaX, xMin,
xMax);

        var deltaY = Input.GetAxis("Vertical") * Time.deltaTime * moveSpeed;
        var newYPosition = Mathf.Clamp(transform.position.y + deltaY, yMin,
yMax);

        transform.position = new Vector2(newXPosition, newYPosition);
    }
}
```

The interactive demonstration of the movement mechanic is exhibited through Unity WebGL, allowing users to engage directly with the game and experience firsthand the movement mechanics of the player character.



MOVEMENT MECHANIC

Interactive Demonstration

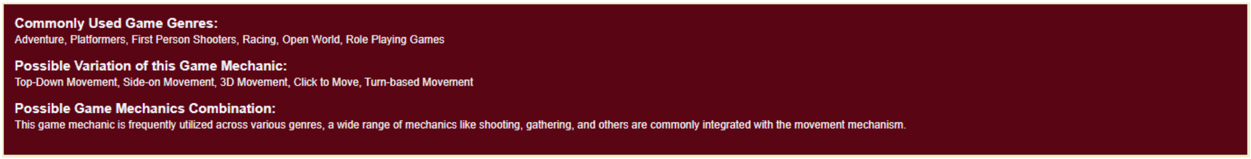
INTERACTIVE CONTROLS

W - Move Forward
A - Move Left
S - Move Backward
D - Move Right

Coding Implementation

```
{
    [SerializeField] float moveSpeed = 10f;

    void Update()
    {
        Move();
    }

    private void Move()
    {
        // Time.deltaTime makes it the same movement for every computers FPS
        var deltaX = Input.GetAxis("Horizontal") * Time.deltaTime * moveSpeed;
        var newXPosition = Mathf.Clamp(transform.position.x + deltaX, xMin,
xMax);

        var deltaY = Input.GetAxis("Vertical") * Time.deltaTime * moveSpeed;
        var newYPosition = Mathf.Clamp(transform.position.y + deltaY, yMin,
yMax);

        transform.position = new Vector2(newXPosition, newYPosition);
    }
}
```

The coding implementation of the movement mechanic is displayed on the right-hand side, featuring the syntax derived from Unity's C#, it presents the logic and functionality behind this game mechanic.

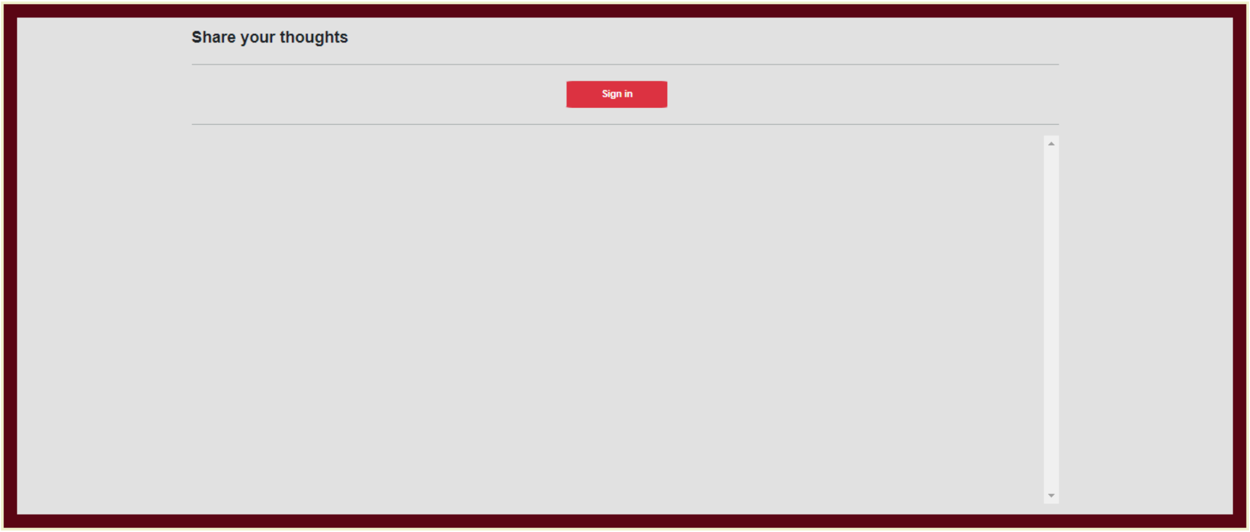


Commonly Used Game Genres:
Adventure, Platformers, First Person Shooters, Racing, Open World, Role Playing Games

Possible Variation of this Game Mechanic:
Top-Down Movement, Side-on Movement, 3D Movement, Click to Move, Turn-based Movement

Possible Game Mechanics Combination:
This game mechanic is frequently utilized across various genres, a wide range of mechanics like shooting, gathering, and others are commonly integrated with the movement mechanism.

Underneath the interactive demonstration and coding implementation is the description of the game mechanic. It describes the movement mechanic's commonly used game genres, possible variation of its game mechanic, and possible combinations of other game mechanics to this feature.



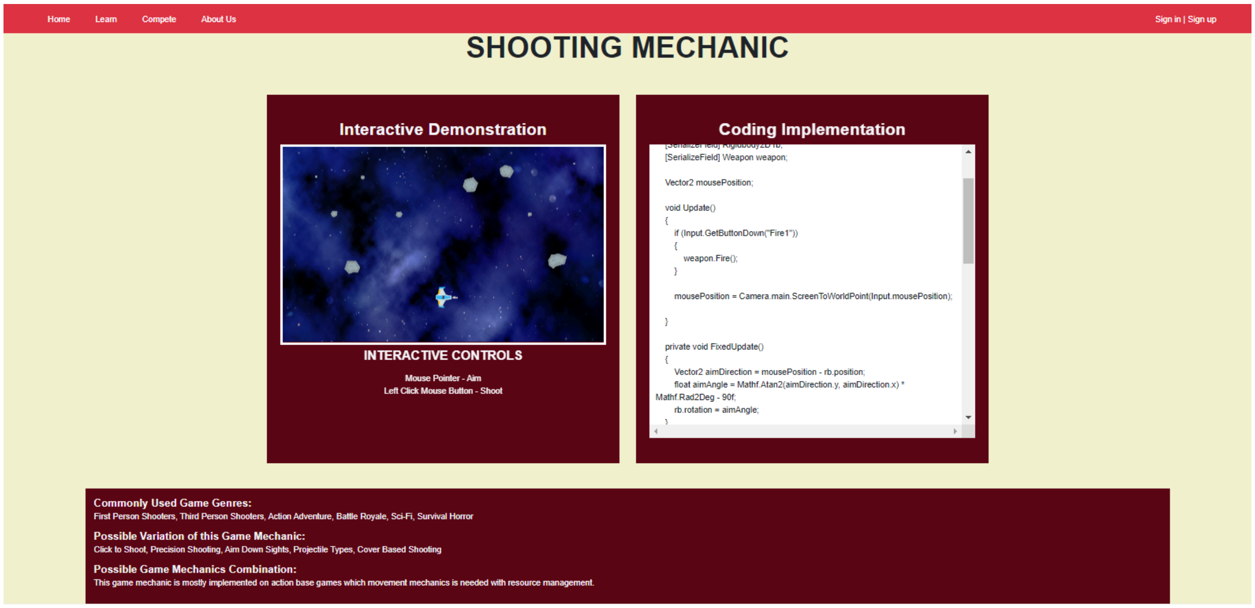
Share your thoughts

Beneath each game mechanic description, there is a comments section which is mandatory for users to have an account in order to contribute their thoughts or feedback about its game mechanic.

Shooting Mechanic

The shooting mechanic stands as a widely favored feature frequently incorporated across numerous gaming titles, serving as a fundamental aspect of gameplay. This particular game mechanic emphasizes the precision aiming and firing elements within the game's dynamics, focusing solely on these essential aspects of combat engagement.

Upon clicking on the shooting mechanic card, users will be redirected to a separate webpage where they can access the interactive demonstration, coding implementation, and description of the game mechanic.



SHOOTING MECHANIC

Interactive Demonstration

INTERACTIVE CONTROLS

Mouse Pointer - Aim
Left Click Mouse Button - Shoot

Coding Implementation

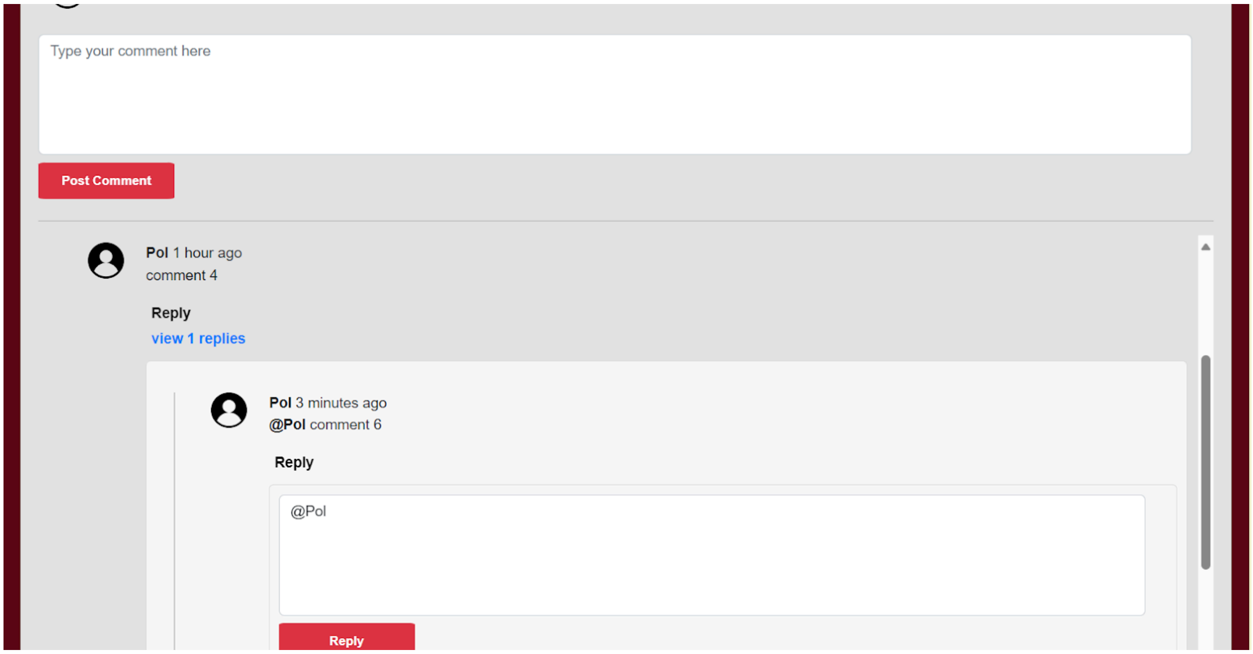
```
    [SerializeField] Rigidbody2D rb;
    [SerializeField] Weapon weapon;

    Vector2 mousePosition;

    void Update()
    {
        if (Input.GetButtonDown("Fire1"))
        {
            weapon.Fire();
        }

        mousePosition = Camera.main.ScreenToWorldPoint(Input.mousePosition);

    }

    private void FixedUpdate()
    {
        Vector2 aimDirection = mousePosition - rb.position;
        float aimAngle = Mathf.Atan2(aimDirection.y, aimDirection.x) *
Mathf.Rad2Deg - 90f;
        rb.rotation = aimAngle;
    }
```

Commonly Used Game Genres:
First Person Shooters, Third Person Shooters, Action Adventure, Battle Royale, Sci-Fi, Survival Horror

Possible Variation of this Game Mechanic:
Click to Shoot, Precision Shooting, Aim Down Sights, Projectile Types, Cover Based Shooting

Possible Game Mechanics Combination:
This game mechanic is mostly implemented on action base games which movement mechanics is needed with resource management.

Similar to the movement mechanic feature, the shooting mechanic also had an interactive demonstration featuring Unity WebGL that demonstrated the aiming and shooting elements. The coding implementation displays the logic of the game mechanic with the use of Unity C#. Underneath is the shooting mechanic's description which mentions the possible variations and combinations of the shooting mechanic and also states the genres that this game mechanic is commonly used.



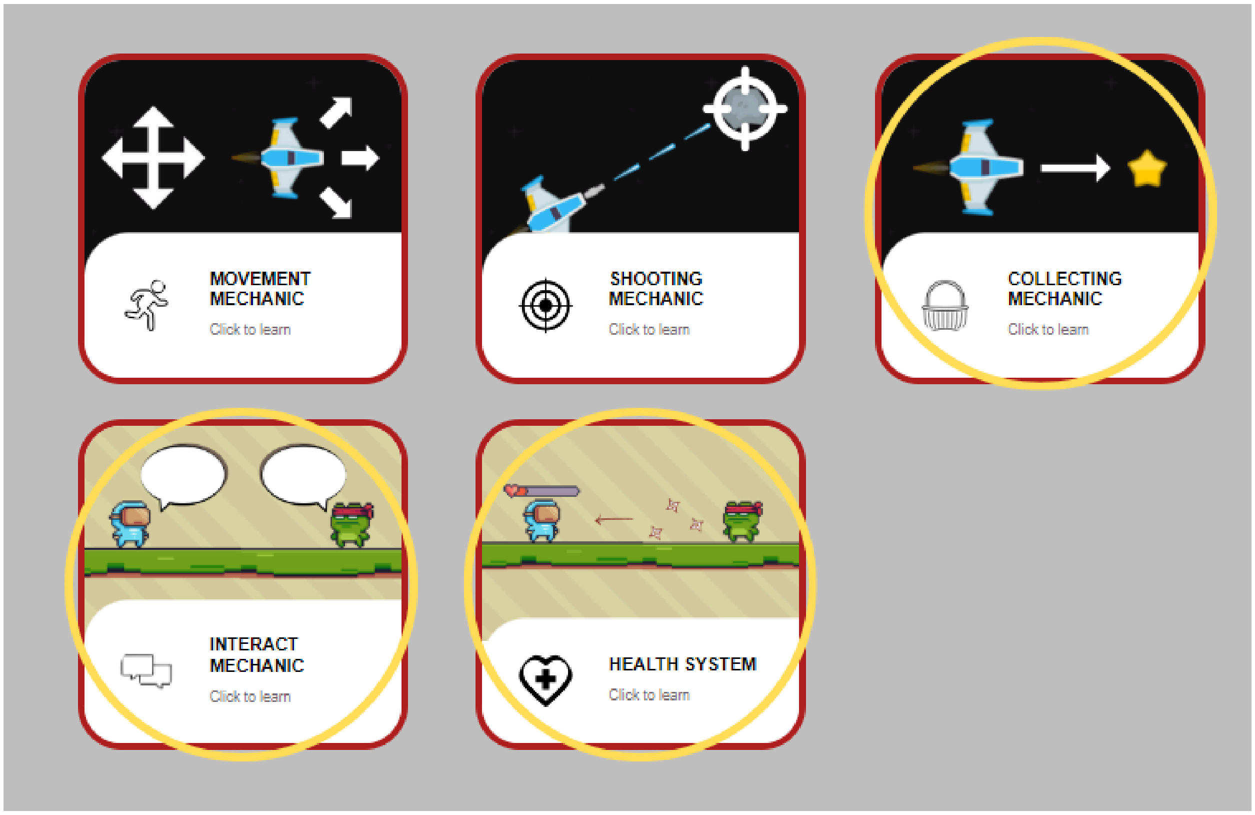
Type your comment here

Pol 1 hour ago
comment 4

Pol 3 minutes ago
@Pol comment 6

@Pol

Once the user log-in their account in VGMech then they are able to access the comment section where they can state their thoughts, questions, and feedback.



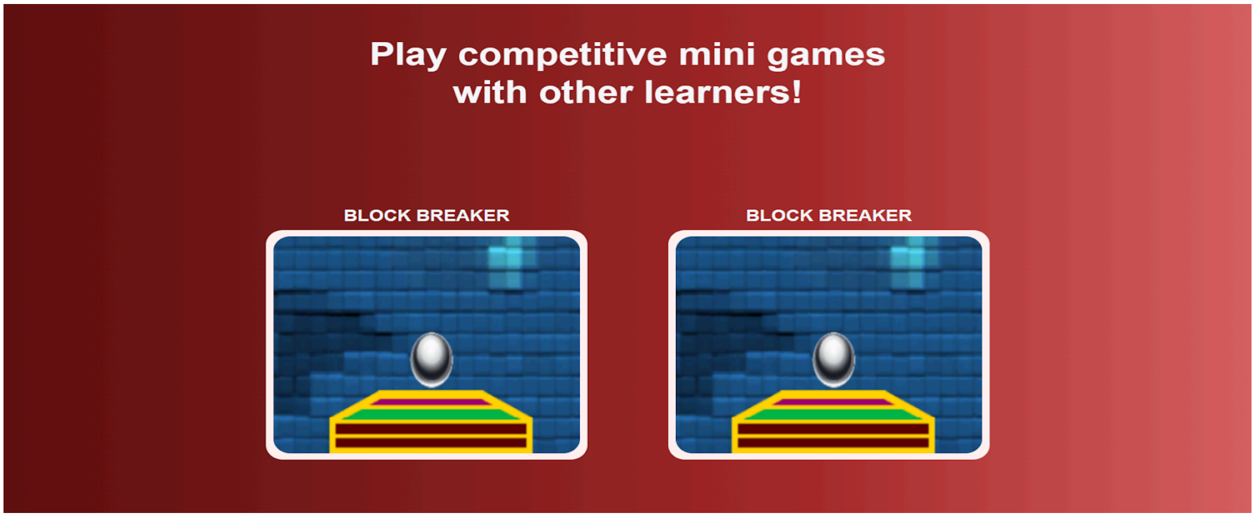
Each mechanic has its own interactive demonstration, coding implementation, description, and comment section with a similar layout and design, but differentiate themselves by their own distinctive mechanic. Further updates shall include additional mechanics for users to learn.

Compete

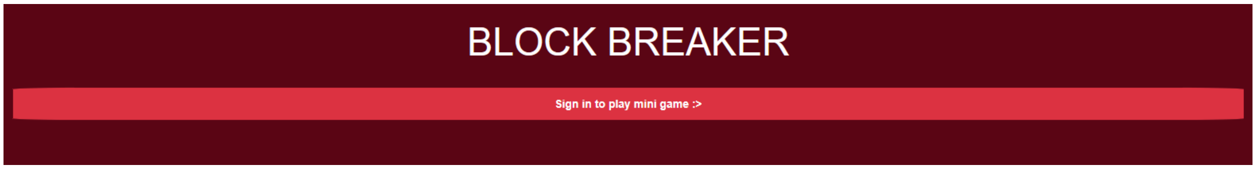
Not only does VGMech help you learn about Game Mechanics, you can also compete with other fellow learners with Block Breaker.

To use the Compete module:
Step 1: Go to the compete section of the homepage.
Step 2: Click the desired minigame to play.
Step 3: Users can view the live leaderboards in the lower part of the minigame page.
Step 4: In order for the Unity game to load, users must first sign in/ sign up their accounts.
Step 5: Once an account is detected, the Unity minigame will load and immediately proceed to the first level.
Step 6: Aim for the highest score and make it to the leaderboard.

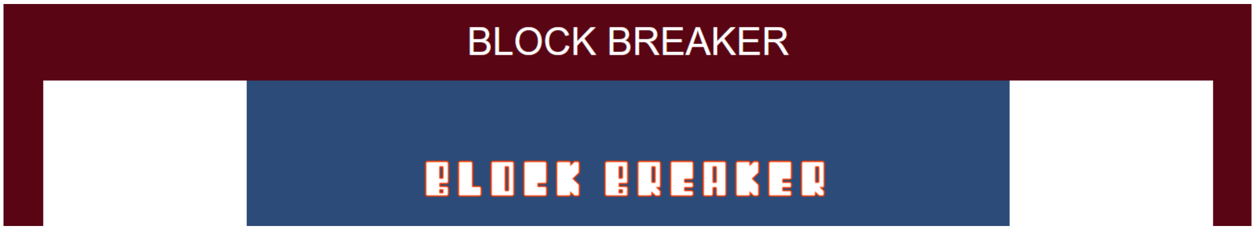
Below are detailed demonstrations

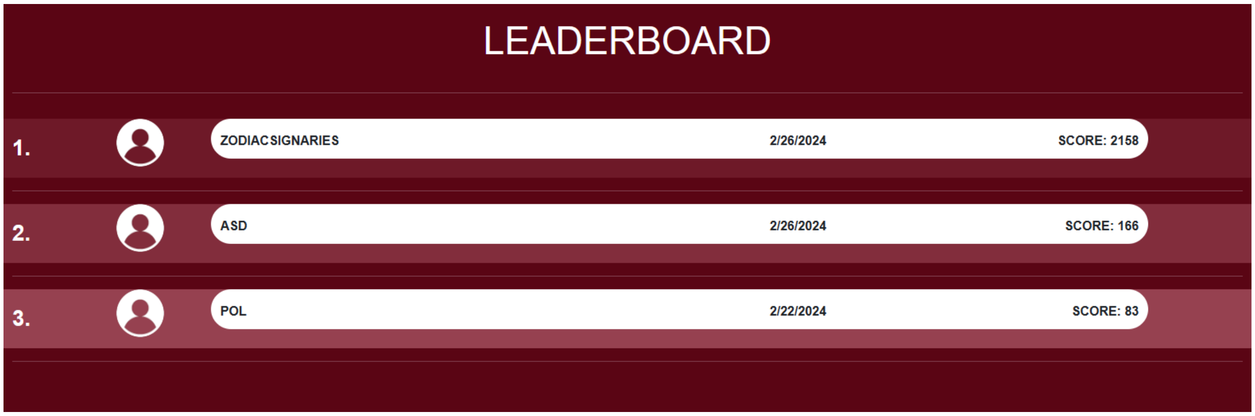


Learners must have an account created before they can play the mini-game.



Once logged in, you can start playing the game. You move the bar using the mouse by sliding from left to right. The goal is to break all the blocks without the ball leaving the bottom part of the screen. Highest scores are listed in the leaderboard.





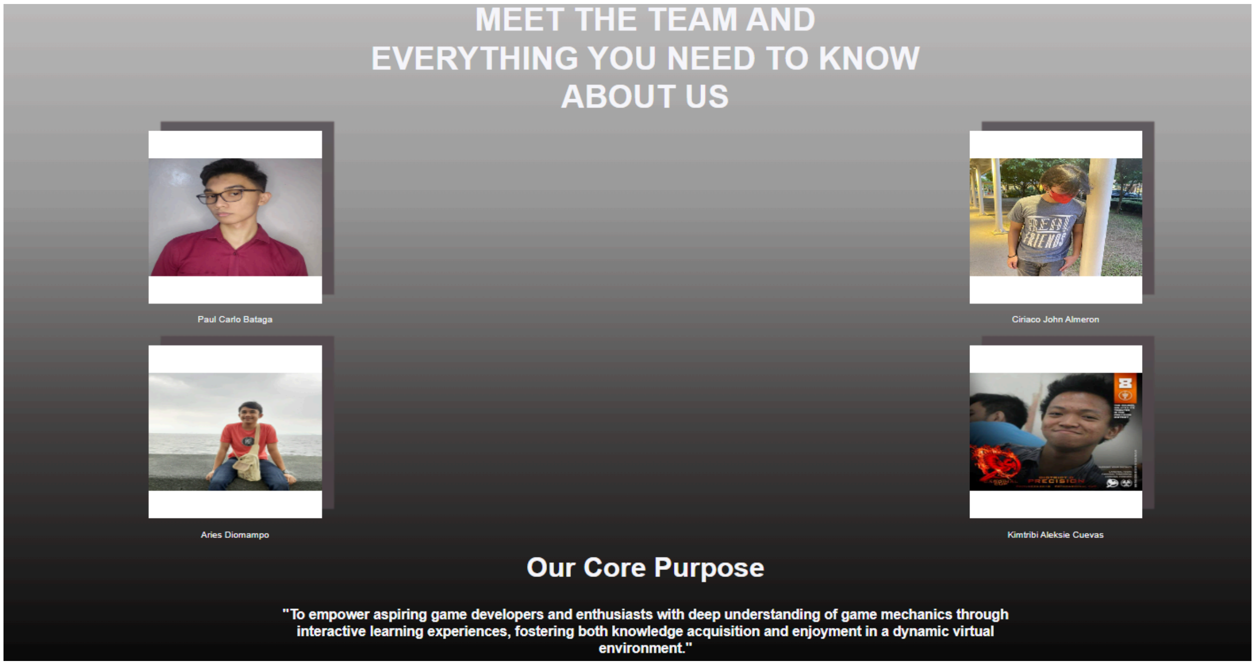
LEADERBOARD

1.	ZODIACSIGNARIES	2/26/2024	SCORE: 2158
2.	ASD	2/26/2024	SCORE: 166
3.	POL	2/22/2024	SCORE: 83

Leaderboard

About Us

Want to know who is responsible for building VGMech? Come see who we are at the about us section. The footer page of the website also offers the following platforms through which users could interact with the developers and the learning community.



MEET THE TEAM AND
EVERYTHING YOU NEED TO KNOW
ABOUT US

Paul Carlo Bataga

Ciriaco John Almeron

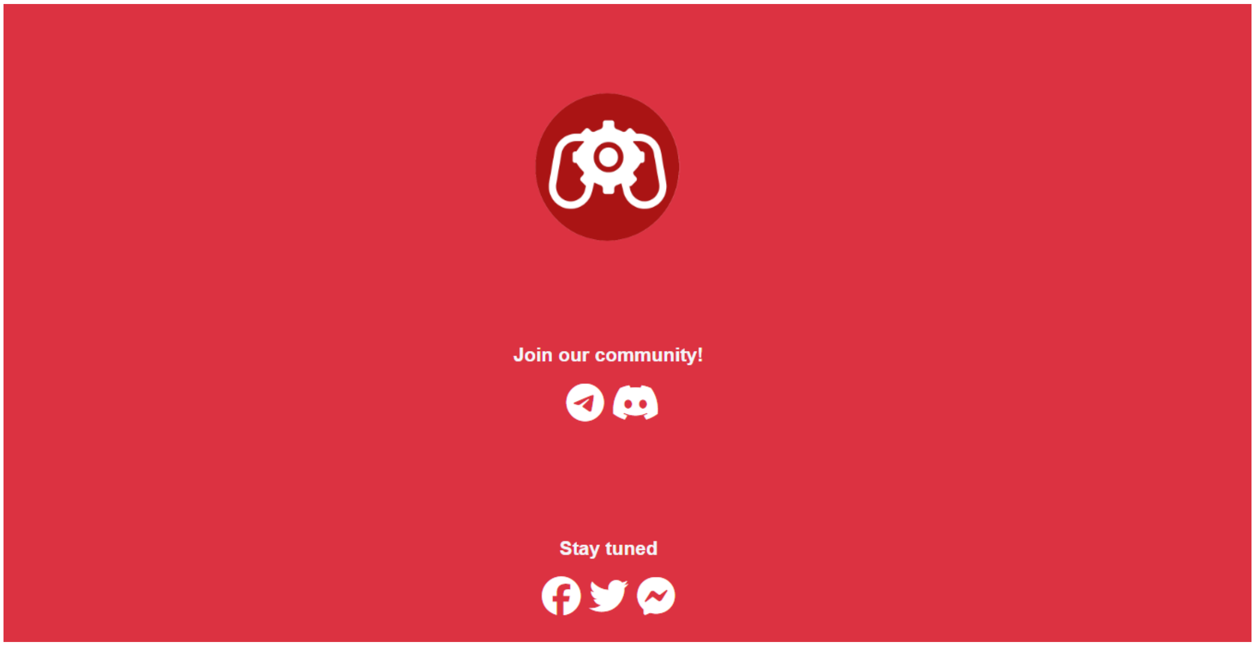
Aries Diomampo

Kimtribi Aleksie Cuevas

Our Core Purpose

"To empower aspiring game developers and enthusiasts with deep understanding of game mechanics through interactive learning experiences, fostering both knowledge acquisition and enjoyment in a dynamic virtual environment."

About Us Section



Join our community!

Stay tuned

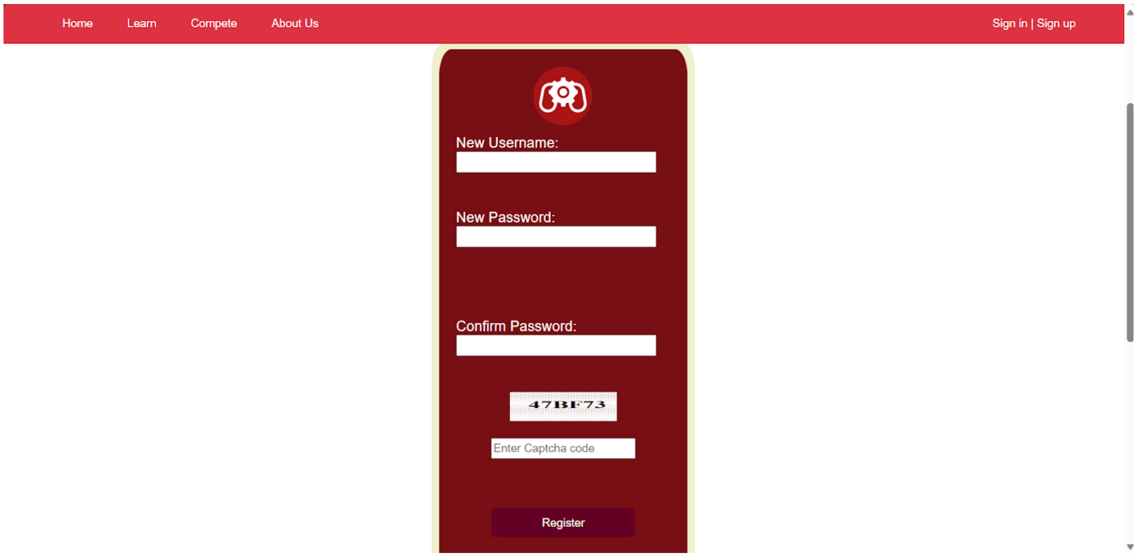
Footer (with Social Media Platforms)

Sign In / Sign Up

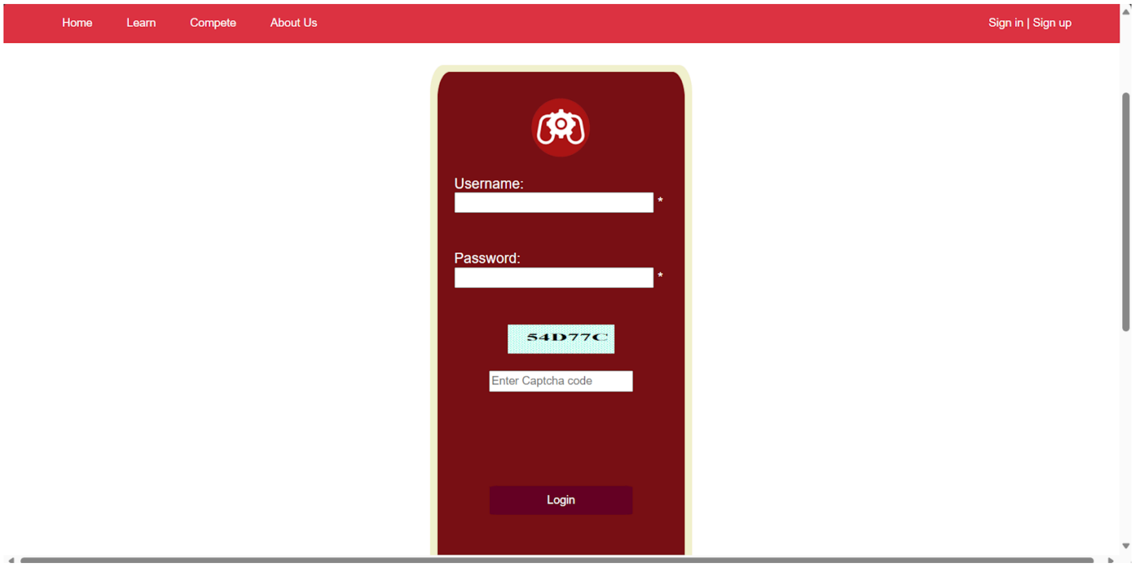
To use the Sign in/ Sign Up module:
Step 1: Go to the sign in/ sign up section using the navigation bar or via the buttons that are directed to the page .
Step 2: If the user needs to create their account, proceed to fill up the required fields following the validations. If not, click the link on the lower right section to access the sign in page.
Step 3: Once done with the filling of information, click register or login and wait for the page response.
Step 4: If the process is successful, the sign in/ sign up link in the navigation bar will turn into the user's name. Once toggled, the user is prompted with an option to log out of their account.

Below are detailed demonstrations

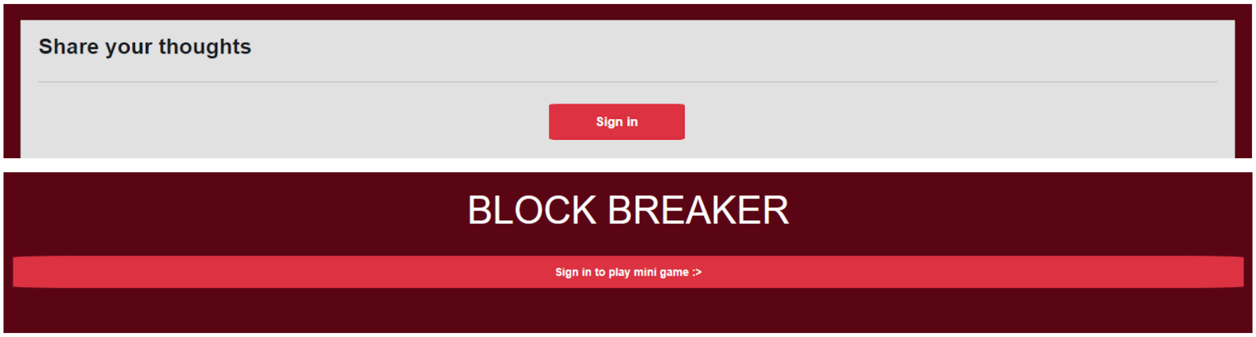
Before the user could participate in playing the minigames or leave their comments on each game mechanic comment section, it is a must for them to sign in/ sign up via the assigned page. Buttons that redirect the user to the sign in page serves as an emphasis for an account. Once done, the hyperlink of sign up/ sign in on the navigation bar turns into the session username button that when toggle prompts an option for the user to log out. Notably, validations are set when creating an account such that username must be 1-15 characters long, and that the password must contain at least one uppercase letter, one lowercase letter, one digit, one special character, and be at least 8 characters long. Also, a confirmation of the password is needed alongside the right captcha entry.



Sign up



Sign In



Buttons that require user sign in

Troubleshooting

Graphics

Slow framerate and/or visual artifacts: With the utilization of Unity WebGL in Learn and Mini sections, the varying quality of the published interactive scenes may be graphically demanding and could induce the said problems. This may occur if your video card drivers are not up to date. Ensure that web browser, operating system, and graphics drivers are up to date to ensure compatibility and optimal performance when running Unity WebGL content.

Graphics Glitches or Rendering Errors:

- **Possible Causes:** Outdated graphics drivers, browser compatibility issues, WebGL rendering errors due to content optimization or complexity, or bugs in the Unity WebGL build.

Content Not Loading or Stuck at Loading Screen:

- **Possible Causes:** Slow or unstable internet connection, browser compatibility issues, server-side issues with content hosting, or browser security settings blocking content loading.

Content Freezing or Crashing:

- **Possible Causes:** Browser memory limitations, browser tab/process crashes due to WebGL content overload or memory leaks, or conflicts with other browser tabs/extensions.

Input Lag or Unresponsive Controls:

- **Possible Causes:** Browser performance issues, input processing delays due to complex content, or conflicts with other browser processes. Notably, users must first click in the Unity WebGL interface for their key inputs to reflect to the interactive system.

Security Warnings or Blocked Content:

- **Possible Causes:** Browser security settings blocking VGMech’s WebGL related content execution, mixed-content warnings due to insecure connections

Browser-specific Issues:

- **Possible Causes:** Browser-specific bugs, limitations or compatibility issues with browser extensions.

Contacting Support

Should an encounter with any technical difficulties or require assistance while interacting with the website's content or Learn section's Unity WebGL content, don't hesitate to reach out to our dedicated support team for prompt and personalized assistance. Below are contacts that you may reach out to:

2021pcbataga@live.mcl.edu.ph
2021cjalmeron@live.mcl.edu.ph
2021asdiomampo@live.mcl.edu.ph
2016kabcuevas@live.mcl.edu.ph