Our Work

Our Work

Games:

The Game Warriors team has over 40 years of combined experience in the gaming industry. Our expertise is dependable and industry backed.

AI:

AI is the cutting edge of technology, and Game Warriors is dedicated to staying at the forefront of tech to ensure the highest possible quality.

Curriculum Software:

Game Warriors doubles as a coding academy, teaching the newest generation to be tomorrows innovators. We built an in-house system to gamify learning and track student progress.

Games

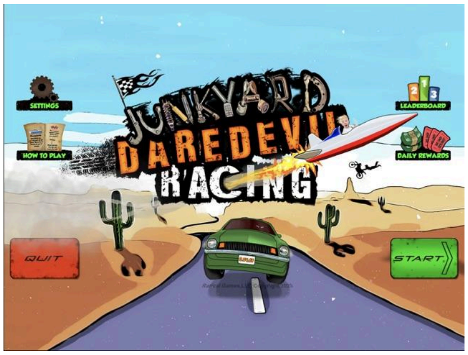
Games

What We’ve Worked On:

Our team has worked on every aspect of game design over the decades, from coding to art direction. Projects we’ve worked on range from IOS, web apps, social media and various consoles.

Here are just a few examples of the over 30+ released titles members of our team has worked on.

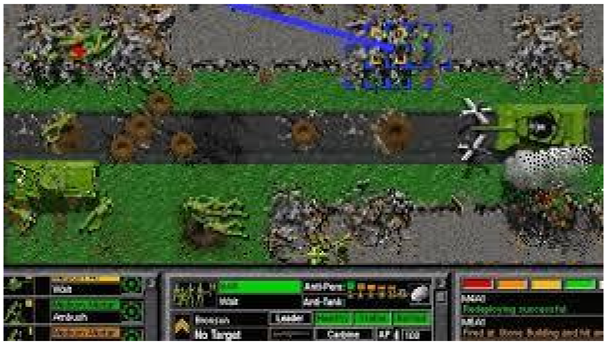
Junkyard Daredevil Racing:
https://apps.apple.com/us/app/junkyard-daredevil-racing/id1533769270



Showdown: Legends of Wrestling:
https://www.imdb.com/title/tt0458478/
https://www.youtube.com/watch?v=Q9IIv9XIYvw

Turok: Evolution:
https://www.imdb.com/title/tt0303164/
https://www.youtube.com/watch?v=MgqmVv_pqtU

Close Combat:
https://store.steampowered.com/app/2916160/Close_Combat/
https://www.youtube.com/watch?v=Bo1i-08WUZo



Redeploying successful

Fired at Stone Building and hit

AI

GAME WARRIORS
CODING AND DIGITAL ART ACADEMY

AI Coding

As AI becomes more commonplace, we remain committed to utilizing AI to its full potential. The following is a sample portion of code for basic AI opponents in a racing game that we utilize to teach our students to understand AI.

```
    void FixedUpdate()
    {
        if (!canMove || waypoints.Count == 0)
            return;

        Vector3 target = waypoints[currentWaypoint].position;
        Vector3 localTarget = transform.InverseTransformPoint(target);
        float distanceToWaypoint = Vector3.Distance(transform.position, target);

        if (distanceToWaypoint < waypointRange)
        {
            currentWaypoint = (currentWaypoint + 1) % waypoints.Count;
            lastDistanceToWaypoint = Mathf.Infinity;
            stuckTimer = 0f;
            reverseTimer = 0f;
            isReversing = false;
            return;
        }

        // Check if stuck
        if (distanceToWaypoint < lastDistanceToWaypoint - 0.1f)
            stuckTimer = 0f;
        else
        stuckTimer += Time.fixedDeltaTime;

    lastDistanceToWaypoint = distanceToWaypoint;

    if (stuckTimer > stuckThreshold && !isReversing)
    {
        isReversing = true;
        reverseTimer = reverseTime;
    }

    steerInput = Mathf.Clamp(localTarget.x / localTarget.magnitude, -1f, 1f);
    currentSpeed = rb.linearVelocity.magnitude * 3.6f;
    float speedRatio = currentSpeed / maxSpeed;

    float forwardGas = Mathf.Clamp01(1f - Mathf.Abs(steerInput) * turningConstant);

    if (isReversing)
    {
        reverseTimer -= Time.fixedDeltaTime;

        if (reverseTimer <= 0f)
        {
            isReversing = false;
            stuckTimer = 0f;
        }

        gasInput = -0.5f;
    }
    else if (isInsideBraking)
    {
        float brakeAmount = Mathf.Clamp01(speedRatio);
        gasInput = -forwardGas * ((brakeAmount * 2f) - 1f);
    }
    else
    {
        gasInput = forwardGas * (1f - speedRatio);
    }

    ApplyControls(gasInput, steerInput);
}

void ApplyControls(float motorInput, float steerInput)
{
    foreach (var wheel in wheels)
```

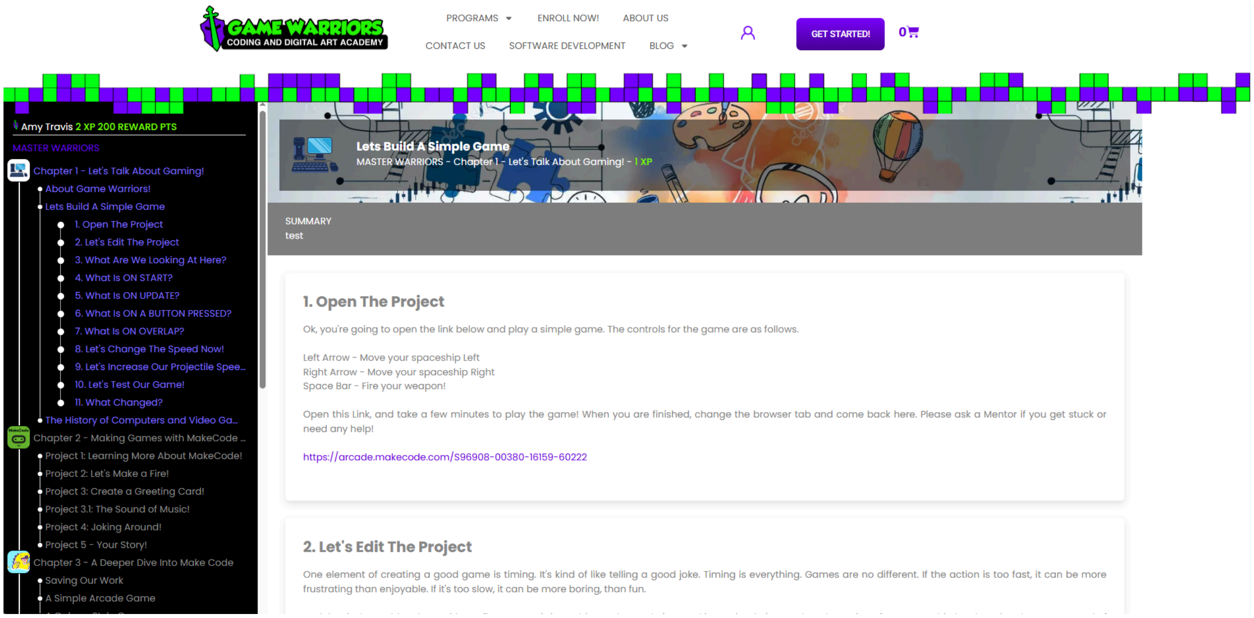
```
        {
            if (wheel.isSteering)
            {
                currentAngle = Mathf.Lerp(currentAngle, steerInput * maxSteerAngle, Time.deltaTime
* 5f);
                wheel.wheelCollider.steerAngle = currentAngle;
            }

            if (wheel.isDriving)
            {
                if (isInsideBraking && currentSpeed > 1f)
                {
                    wheel.wheelCollider.motorTorque = motorInput * maxMotorTorque;
                    wheel.wheelCollider.brakeTorque = 0f;
                }
                else if (!isInsideBraking && currentSpeed < maxSpeed)
                {
                    wheel.wheelCollider.motorTorque = motorInput * maxMotorTorque;
                    wheel.wheelCollider.brakeTorque = 0f;
                }
                else
                {
                    wheel.wheelCollider.motorTorque = 0f;
                    wheel.wheelCollider.brakeTorque = maxBrakeTorque;
                }
            }

            UpdateWheelVisual(wheel);
        }
    }

    void UpdateWheelVisual(Wheel wheel)
    {
        wheel.wheelCollider.GetWorldPose(out Vector3 pos, out Quaternion rot);
        wheel.wheelTransform.position = pos;
        wheel.wheelTransform.rotation = rot;
    }

    public void ApplyDifficultySettings()
    {
        switch (aiDifficulty)
        {
            case Difficulty.Easy:
                maxMotorTorque = baseMaxMotorTorque * 0.6f;
                maxBrakeTorque = baseMaxBrakeTorque * 1.2f;
                maxSteerAngle = baseMaxSteerAngle * 0.8f;
                maxSpeed = baseMaxSpeed * 0.7f;
                turningConstant = baseTurningConstant * 1.5f;
                break;

            case Difficulty.Medium:
                maxMotorTorque = baseMaxMotorTorque;
                maxBrakeTorque = baseMaxBrakeTorque;
                maxSteerAngle = baseMaxSteerAngle;
                maxSpeed = baseMaxSpeed;
                turningConstant = baseTurningConstant;
                break;

            case Difficulty.Hard:
                maxMotorTorque = baseMaxMotorTorque * 1.2f;
                maxBrakeTorque = baseMaxBrakeTorque * 0.8f;
                maxSteerAngle = baseMaxSteerAngle * 1.1f;
                maxSpeed = baseMaxSpeed * 1.1f;
                turningConstant = baseTurningConstant * 0.7f;
                break;
        }
    }
}
```

Curriculum Software

GAME WARRIORS
CODING AND DIGITAL ART ACADEMY

Curriculum Software

Game Warriors thrives as a local coding academy as well as a development team. We created an in-house curriculum system that includes curriculum that we developed ourselves. Our system is optimized for educational structure and the gamification of the learning process. We've blended hands-on learning with skill checks, and a rewards point system to keep students motivated and engaged.



Amy Travis 2 XP 200 REWARD PTS

MASTER WARRIORS

- Chapter 1 - Let's Talk About Gaming!
  - About Game Warriors!
  - Lets Build A Simple Game
    - 1. Open The Project
    - 2. Let's Edit The Project
    - 3. What Are We Looking At Here?
    - 4. What Is ON START?
    - 5. What Is ON UPDATE?
    - 6. What Is ON A BUTTON PRESSED?
    - 7. What Is ON OVERLAP?
    - 8. Let's Change The Speed Now!
    - 9. Let's Increase Our Projectile Spee...
    - 10. Let's Test Our Game!
    - 11. What Changed?
  - The History of Computers and Video Ga...
- Chapter 2 - Making Games with MakeCode ...
  - Project 1: Learning More About MakeCode!
  - Project 2: Let's Make a Fire!
  - Project 3: Create a Greeting Card!
  - Project 3.1: The Sound of Music!
  - Project 4: Joking Around!
  - Project 5 - Your Story!
- Chapter 3 - A Deeper Dive Into Make Code
  - Saving Our Work
  - A Simple Arcade Game

Lets Build A Simple Game
MASTER WARRIORS - Chapter 1 - Let's Talk About Gaming! - 1 XP

SUMMARY
test

1. Open The Project

Ok, you're going to open the link below and play a simple game. The controls for the game are as follows.

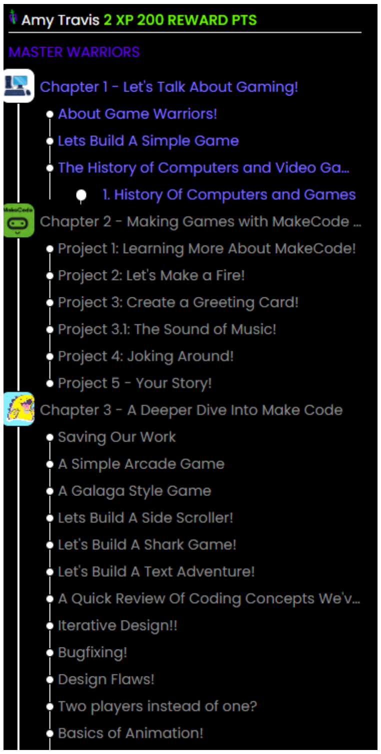
Left Arrow - Move your spaceship Left
Right Arrow - Move your spaceship Right
Space Bar - Fire your weapon!

Open this Link, and take a few minutes to play the game! When you are finished, change the browser tab and come back here. Please ask a Mentor if you get stuck or need any help!

https://arcade.makecode.com/S96908-00380-16159-60222

2. Let's Edit The Project

One element of creating a good game is timing. It's kind of like telling a good joke. Timing is everything. Games are no different. If the action is too fast, it can be more frustrating than enjoyable. If it's too slow, it can be more boring, than fun.



Amy Travis 2 XP 200 REWARD PTS

MASTER WARRIORS

Chapter 1 - Let's Talk About Gaming!
- About Game Warriors!
- Lets Build A Simple Game
- The History of Computers and Video Ga...
  - 1. History Of Computers and Games

Chapter 2 - Making Games with MakeCode ...
- Project 1: Learning More About MakeCode!
- Project 2: Let's Make a Fire!
- Project 3: Create a Greeting Card!
- Project 3.1: The Sound of Music!
- Project 4: Joking Around!
- Project 5 - Your Story!

Chapter 3 - A Deeper Dive Into Make Code
- Saving Our Work
- A Simple Arcade Game
- A Galaga Style Game
- Lets Build A Side Scroller!
- Let's Build A Shark Game!
- Let's Build A Text Adventure!
- A Quick Review Of Coding Concepts We'v...
- Iterative Design!!
- Bugfixing!
- Design Flaws!
- Two players instead of one?
- Basics of Animation!