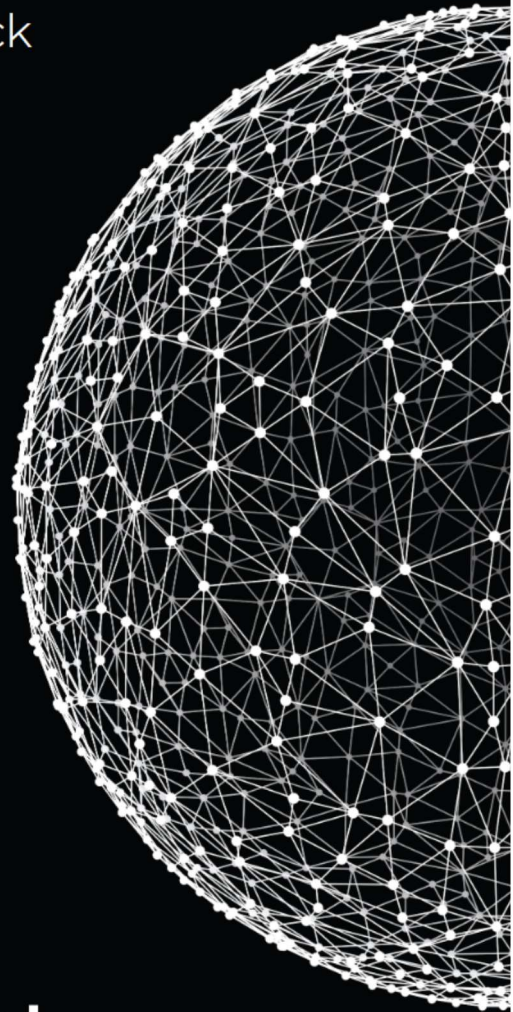




Race01

Half Marathon Full Stack

November 15, 2021



 ucode connect

Contents

Engage 2

Investigate 3

Act: Basic 6

Act: Creative 8

Share 9



Engage

DESCRIPTION

Welcome to the last **Half Marathon Full Stack** challenge - **Race01!**

You are here and this is worth a thousand words.

Achieving goals is not so simple.

If you want to achieve something meaningful in your life, you will have to get out of your comfort zone. That's because meaningful things don't come easily. Still, don't let high expectations slow you down.

Do not expect everything you create to come out perfectly on the first attempt. But you should definitely strive for it!

Think about your ultimate goal, and then do everything you can (and even more) to achieve it. Hard intellectual work is the only way to achieve greatness.

Sorry, we got carried away. We don't want to burden you with all that motivation stuff, which is all over the Internet these days. But we have something interesting for you.

To master large amounts of information, they must be used in practice. During the **Half Marathon**, you must have learned a lot. Not, let's put it together.

We could offer you to create a big complicated web app. But it is boring.

We hope you like games. To create a big game, you need as much skills and diligence as to create a web app. But the process and result of creating games are much funnier!

So, we challenge you to take part in Race01! Here, you will show what you are capable of!

Good luck!

BIG IDEA

Knowledge consolidation.

ESSENTIAL QUESTION

How does achieving small goals lead to great success?

CHALLENGE

Sum up all the knowledge gained during the **Half Marathon**.

Investigate

GUIDING QUESTIONS

We invite you to find answers to the following questions. By researching and answering them, you will gain the knowledge necessary to complete the challenge. To find solutions, ask the students around you and search the internet. We encourage you to ask as many questions as possible. Note down your findings and discuss them with your peers.

- How was your **Race** organized? Were you satisfied with teamwork? Can you improve something this time?
- What are the basics of authorization process?
- What is the difference between website development and game development for the browser?
- What factors affect cross-browser compatibility of a web page?
- How can you check your website for cross-browser compatibility across different operating systems?
- What is the technical background of each team member?
- What technologies has every team member used before? What frameworks have you used?
- What are your expectations for this challenge? What is your deadline for this challenge?
- What tools will you use for teamwork? Jira, Asana, Trello, etc.?
- What each of you would like to do more?
- What makes a game fun or boring?
- What is a complete product?
- What is a product-oriented mindset?
- What needs to be done first when creating a solution?
- What stages do game developers go through when working on their own product?
- What experience do developers receive from this process?
- Which card games do you know?

GUIDING ACTIVITIES

Complete the following activities. Don't forget that you have limited time to overcome the challenge. Use it wisely. Distribute tasks correctly.

- Read the story.
- Meet with your team. Discuss teamwork organization and communication.
- Discuss your experience of working collaboratively on **Race00**.
- Define what was poorly organized and fix it in the new dream team.
- Discuss the terms of the challenge together: game, solution, product, etc.
- Talk with community members on a variety of topics and find out what they like about computer games.
- Decide how to split the work in a way that is beneficial to each team member and the project in general. Listen to each other carefully.

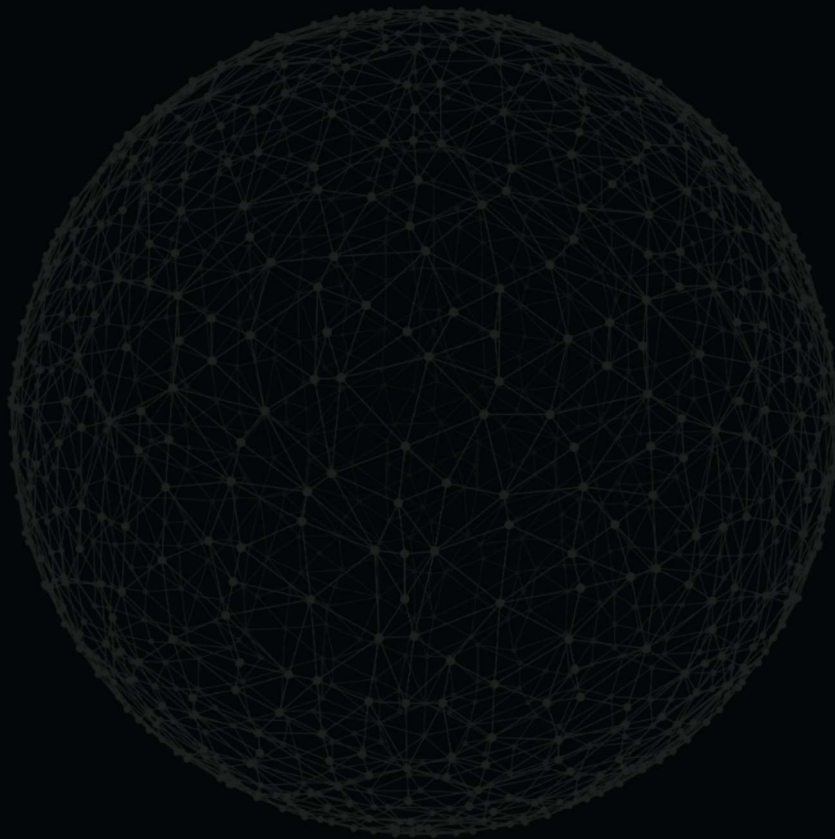
- Find and play popular card games. Not solitaire, or blackjack, but Hearthstone and Gwent. Pay attention to the every little thing.
- Evaluate the main features of the games you played. Find some you want to implement in your game.
- Clone your git repository that is issued on the challenge page.
- Start to develop the solution. Offer improvements. Test your code.
- Remember to test your web app for **cross-browser compatibility**.
- Explore new things. Talk, discuss and make conclusions.

ANALYSIS

Analyze your findings. What conclusions have you made after completing guiding questions and activities? In addition to your thoughts and conclusions, here are some more analysis results.

- Challenge has to be carried out by the entire team.
- Each team member must understand the challenge and realization, and be able to reproduce it individually.
- It is your responsibility to assemble the whole team. Phone calls, SMS, messengers are good ways to stay in touch.
- Be attentive to all statements of the story.
- Analyze all information you have collected during the preparation stages.
- Research the tools and technologies that will help you realize your solution.
- Submit your files using the layout described in the story. Garbage shall not pass.
- Pay attention to what is allowed. Use of forbidden stuff is considered a cheat and your challenge will be failed.
- You can proceed to **Act: Creative** only after you have completed all requirements in **Act: Basic**. But before you begin to complete the challenge, pay attention to the program's architecture. Take into account the fact that many features indicated in the **Act: Creative** require special architecture. And in order not to rewrite all the code somehow, we recommend you initially determine what exactly you will do in the future. Note that the **Act: Basic** part gives the minimum points to validate the challenge.
- Complete tasks according to the rules specified in the following style guides:
 - HTML and CSS: **Google HTML/CSS Style Guide**. As per section 3.1.7 **Optional Tags**, it doesn't apply. Do not omit optional tags, such as `<head>` or `<body>`
 - JavaScript:
 - * **JavaScript Style Guide and Coding Conventions**
 - * **JavaScript Best Practices**
 - * **Node.js Best Practices**
- Do not forget, that the user interface should be easy to use.
- The solution will be checked and graded by students like you. **Peer-to-Peer learning**.

- If you have any questions or don't understand something, ask other students or just Google it.



ucode connect

Race01 | Half Marathon Full Stack > 5

Act: Basic

NAME

Great Battle

DIRECTORY

./

SUBMIT

All files you need for the challenge

ALLOWED

HTML, CSS, JS, MySQL

LEGEND

Who's stronger: the villains or the heroes? What if you bring together some heroes and villains, which combination is better? This challenge can give answers to these and other super-questions.

DESCRIPTION

Create an interesting card game about battles between Marvel characters. It should start with a user logging into an account. Respectively, there must be a registration system. This is what enables the user to take part in the battle. Choose a mechanism of battle creation. It can be the creation of a room and waiting for the enemy, or the players can be paired automatically. It's up to you.

Gameplay:

- each match is a battle between two opponents against each other
- the gameplay must be turn-based, with players taking turns to play cards they have on their hands
- each player is represented by an avatar (photo) and a login (nickname)
- each player has points of **Health** (e.g., 20 points) at the beginning of the match
- each player has **N** cards at the beginning of the match. **N** - number of cards in a hand. It is up to you
- in each round, players receive random cards, so they always have **N** cards in their hands
- once **Health** is reduced to zero, the avatar is destroyed and the corresponding player loses the game
- each player's turn is time-limited (e.g., 30 seconds). It would be cool if the players could see how much time has already expired
- the goal of the game is to leave the opponent without **Health**
- the order of each game should be chosen randomly (you can add visual effects to show this process, e.g., toss a coin)

Cards and Deck:

- implement at least 20 different cards
- decide what will be on the cards. Search for images of Marvel characters
- each card has a few parameters:
 - points of attack
 - points of defense
 - card's cost (the stronger the card the higher its cost)
- all cards should display the avatar and alias of the card's character (Thanos, Iron Man, etc.), points of attack, defense, and cost

Battlefield:

- the **battlefield** (or **game board**) is the board where each player is represented and the action takes place
- you must implement some required UI elements around the **battlefield**, such as:
 - cards of the current player
 - players themselves
- logically arrange players' avatars. For example, the player's avatar can be displayed near the bottom of the **battlefield**, while that of the opponent can be near the top
- it is necessary to divide the board into two parts: for the player's and the opponent's active cards

Gamedesign details are up to you.

SEE ALSO

Hearthstone

The Elder Scrolls: Legends

Act: Creative

DESCRIPTION

It is the place where your imagination and creativity plays a major role. Implement additional features to make the program better and more unique.

You are welcome to implement any feature your team considers valuable, reasonable or just interesting. Be ready to explain your choices.

Your extra work will be evaluated accordingly.



Share

PUBLISHING

Last but not least, the final stage of your work is to publish it. It allows you to share your challenges, solutions, and reflections with local and global audiences. During this stage, you will discover ways of getting external evaluation and feedback on your work. As a result, you will get the most out of the task and better understand your achievements and missteps.

To share your work, you can create:

- a text post, as a summary of your reflection
- charts, infographics or other ways to visualize your information
- a video, either of your work, or a reflection video
- an audio podcast. Record a story about your experience
- a photo report with a small post

Helpful tools:

- **Canva** - a good way to visualize your data
- **QuickTime** - an easy way to capture your screen, record video or audio (macOS)
- **ScreenToGif** - screen, webcam, and sketchboard recorder with an integrated editor (Windows)

Examples of ways to share your experience:

- **Facebook** - create and share a post that will inspire your friends
- **YouTube** - upload an exciting video
- **GitHub** - share and describe your solution
- **Instagram** - share photos and stories from ucode. Don't forget to tag us :)

Share what you've learned and accomplished with your local community and the world. Use **#ucode** and **#CBLWorld** on social media.