

Study notes on Second MileStone project

08 Feb 2021

I started reading the assessment criteria document, I was curious what are the suggested project ideas. My imagination started running wild when I saw “Memory Game”. I had a flash about a boardgame we played a lot in the family. It is about eskimos and animals living on and under the ice.

here is a description about the game: <https://boardgamegeek.com/boardgame/36554/enuk>



picture about the box of the game

I set up the repository here: <https://github.com/ruskipista/cims02-enuk>

The hosted webpage on GitHub Pages: <https://ruskipista.github.io/cims02-enuk/>

Created index.html with starter template for Bootstrap

Created favicon for the browser tab with Paint.Net



Took photos of the board, game tiles, pieces - edited them in Paint.Net

Found a card flipping example here https://www.w3schools.com/howto/howto_css_flip_card.asp

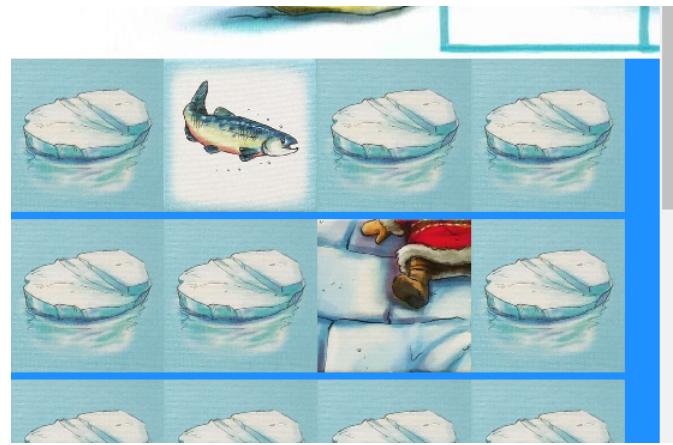
and implemented, because in my mind that was the “biggest” challenge, how I will flip the game tiles...

That example was triggered by mouse hovering over the tile, but I need clicking instead. I studied the CSS code long and hard.

coded random shuffling an array using

<https://stackoverflow.com/questions/2450954/how-to-randomize-shuffle-a-javascript-array>

coded the generation of a series of different tiles, so far only as inline-block, no responsive design yet (not created the wireframes yet)



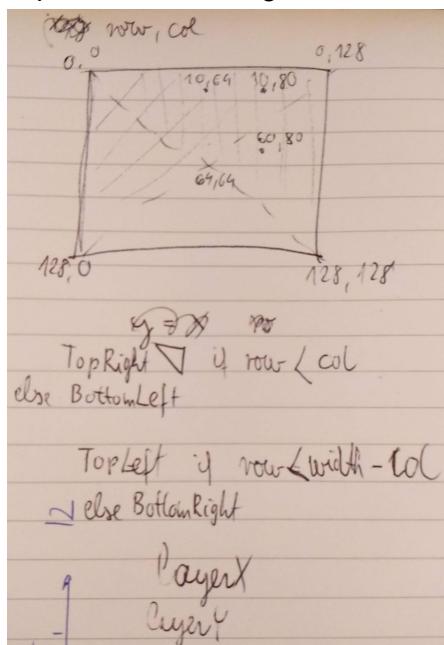
see 2 tiles flipped from the series

09 Feb 2021

the <https://boardgamegeek.com/boardgame/36554/enuk> hosted a game rules document, full of graphics which might come handy for decoration. I disassembled the PDF with Adobe Acrobat Pro - my wife has a study licence on it.

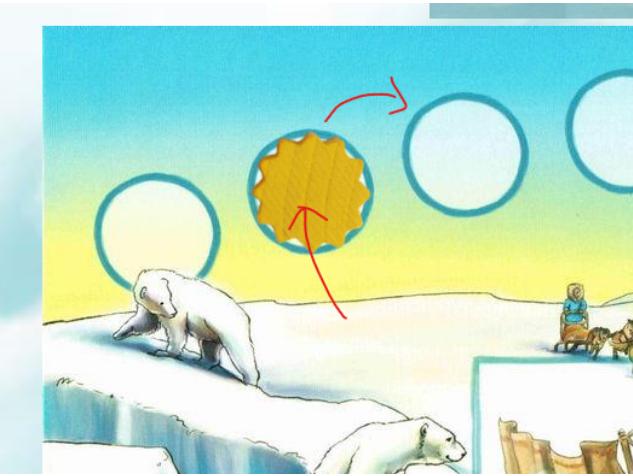
collected game rules and images in the assets/doc folder

coded the click triggered flipping part with recognising which quadrant of a tile was clicked and making the transition dependable on it. e.g. if clicked on the left portion of the tile, it flip to the left, and so on up/down/right.



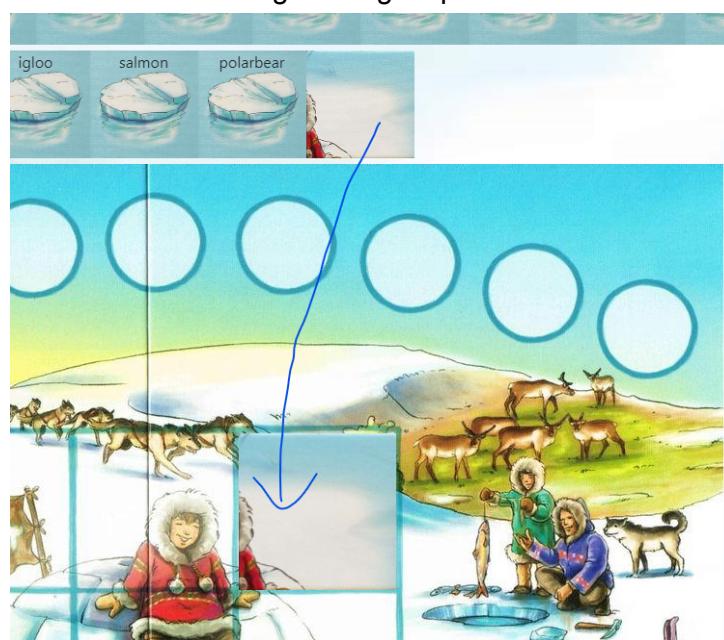
10 Feb 2021

coded to place the sun piece on board and transition to next position when clicked, the movement between two position is animated with rotation

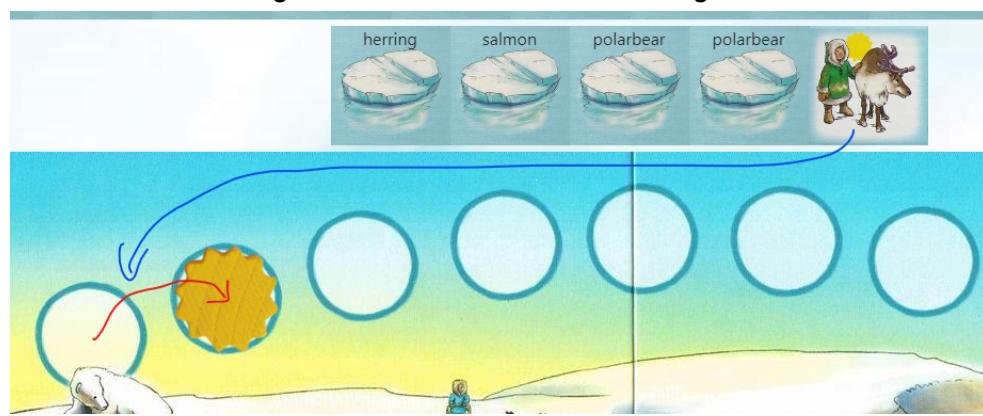


11 Feb 2021

set up the 9 igloo pieces on the board and made the visibility switchable
connected the turning of an igloo piece tile with revealing the same igloo piece on board



connected the turning of a reindeer tile with advancing the sun into the next position



12 Feb 2021

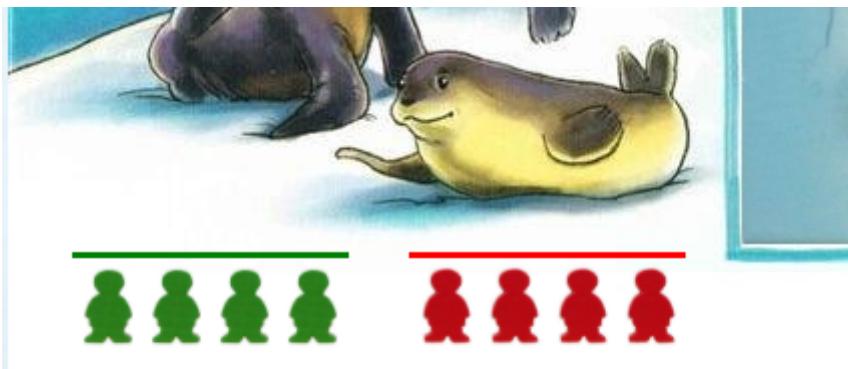
thinking on how the player will select how many players there will be:
human icon <https://icons.getbootstrap.com/icons/person-fill/>
lightning icon <https://icons.getbootstrap.com/icons/lightning/>

cpu icon <https://icons.getbootstrap.com/icons/cpu-fill/>

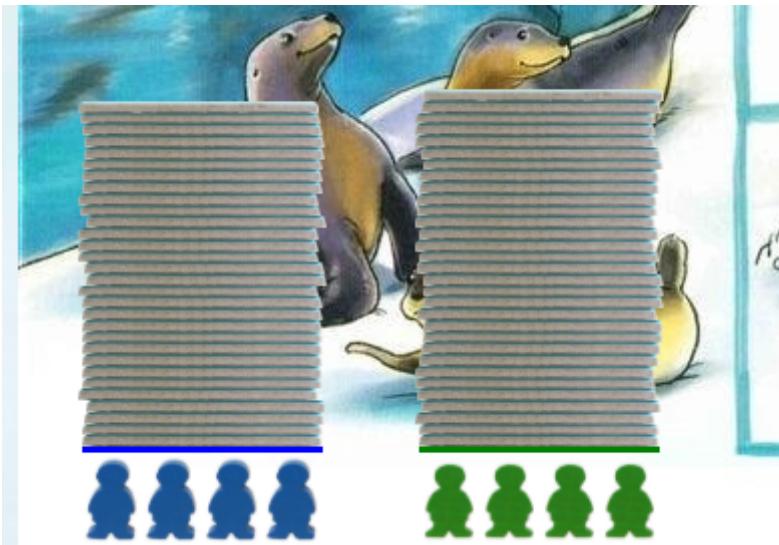
laptop <https://icons.getbootstrap.com/icons/laptop/>



if they click on the human meeple, the color of the meeple gets cycled between the 6 available colors
if user clicks on any of the laptop icons, its color changes from grey to one of the 5 (or 4,3,2) available colors, this represents, that the user chose to play against (0-4) machine(s)
added 4 meeples for each players to the bottom of the board:



added code to represent collected tiles in stacks over the players' base:



the stack has tile edges put on each other, there are two kind of tile edge: one for middle, one for the top. The layering is modified -2..+2px left-right random adjustments to make the impression as it was built by hand.
Each stack is a <div> turned upside down.

refactored the game materials function into an object where each material type is a separate attribute of the object, this freed up from the use of specific literals for claiming each material types.

13 Feb 2021

created the Tile class, separated out code to the new class which deals with single tile

trying to separate from Viewer and Controller code, I can see it is not going to be easy

created CollectTiles icon and connected with action to remove upturned tiles from table and put them onto the board or the player's stack



14 Feb 2021

created laptop meeples to symbolise non-human players



coded randomization of player color and move order

there is a plan to start the game with settings like how many players and human player's choice of color - but we are not there yet

worked on the rules and the definition of End Of Phase 1

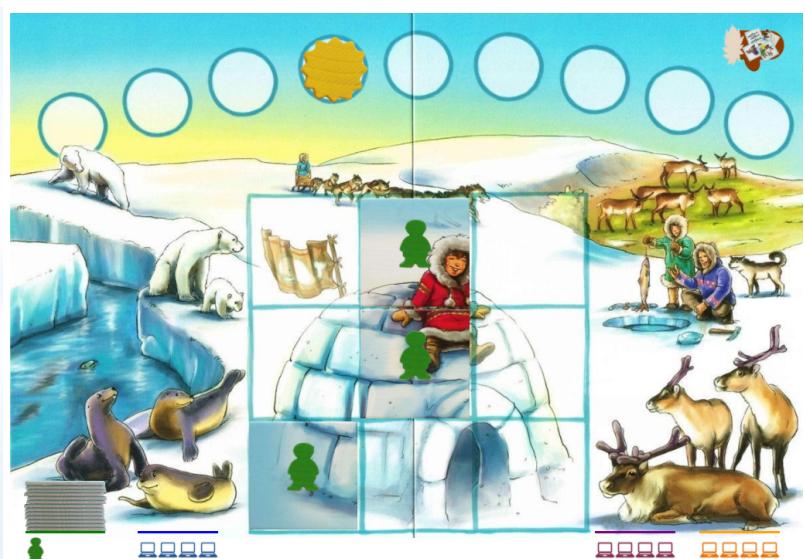
15 Feb 2021

created a yellow background out of the light blue, this yellow will be displayed if the game reaches the end of phase 1, arrive to phase 2

created new class Figure and separated out figure related code

further work went into the evaluation what is executed after each tile turn or at the Collect Tiles action

separated the evaluation and acting on the evaluation result and put a time gap between current status (in test mode):



16 Feb 2021

call with mentor:

- presented development current status

received advice to

- explore grid layout to present tiles
- check out animation.css
- slide game board out of way, slide in on user action
- it is possible to trigger click event programmatically

interesting possibility: generate an event, put an event listener on a function: entry point into the whole code, can be triggered programmatically:

https://developer.mozilla.org/en-US/docs/Web/Guide/Events/Creating_and_triggering_events

I can use this event triggering to build a status machine to handle rounds

20 Feb 2021

tried the jshint.com for the javascript code I wrote so far, corrected semicolon mistakes

21 Feb 2021

replaced the use of word Figure with Meeple in the code

working on the control of the game play with a state machine, this would be the pseudo code:

there are 2 phases in the game:

Phase1: flipping and collecting tiles, building the igloo

Phase2: player declares next tile before flipping it, if correct, collects the tile

A. status **BeforePhase1** (generate game area for Phase 1)

1. generate game table with 74 tiles face down, random order in matrix layout (rows and columns)
 - 9 igloo pieces
 - 14 herring
 - 14 salmon,
 - 14 seal,
 - 14 polarbear,
 - 9 reindeer (with eskimo)
2. generate game board with
 - empty tile stack of player(s),
 - 4 meeples for each player under their stack,
 - empty 3x3 tile holder over the igloo,
 - empty 3x3 meeple holder on top of the 3x3 tile holder
 - place the sun piece on the first of 9 positions
 - place invisible icons of herring, salmon, seal, polarbear, reindeer, igloo
3. determine order of player moves
4. set ActualPlayer to the first player
5. continue to status **InPhase1-BeforeMove**

B. status **InPhase1-BeforeMove** (prepare actual player's move)

1. instruct ActualPlayer to move (flip or collect)
2. wait for request

C. status **InPhase1-ProcessMove** (collecting tiles and building the igloo)

- receive move from ActualPlayer (ClickedTile, Request)
1. If Request is RequestToFlip to flip a face-down tile up, then
 - -> set flag RequestToFlip
 - -> flip the tile face-up
 - If ClickedTile is reindeer -> move the sun piece to the next position. if there is no next position (already on the final), then do not move sun
 2. If Request is CollectTiles to collect face-up tiles from table -> set flag RequestToCollect
 3. If Request is something else -> continue to status **InPhase1-BeforeMove**

4. continue to status **InPhase1-Evaluation**

D. status **InPhase1-Evaluation** (evaluate status after move)

1. clear evaluation flags
2. check face-up tiles on table:
 - o determine the list of animals fleeing from any other animal
(rank order: herring < salmon < seal < polarbear < reindeer.
An animal flees from the next higher ranked animal only, e.g. salmon flees from seal, but not from polarbear or from any others)
 - o If any flees -> set flag AnimalFled
 - o determine the list of igloo tiles to be removed to the board
 - o determine the list of other face-up tiles
3. If NOT flag AnimalFled AND all tiles are face-up -> set flags EndOfPhase2, EndOfPhase1
4. Else If ClickedTile is reindeer AND the sun piece is on the last position -> set flag EndOfPhase1
5. If flag EndOfPhase1
 - OR flag EndOfPhase2
 - OR ClickedTile is igloo
 - OR flag AnimalFled
 - OR flag RequestToCollect
 - > set flag EndOfMove
6. continue to status **InPhase1-Execution**

E. status **InPhase1-Execution** (execute actions based on evaluation)

1. If ClickedTile -> wait some time that each player can memorize the last tile flip
2. turn tiles of fleeing animals back face-down
3. move igloo tiles onto the board's 3x3 igloo and mark each tile with the player's one (of 4) meeples under its stack. If there are no meeples left, do not mark.
4. If EndOfMove -> collect remaining face-up tiles into player's stack
5. wait some time that each player can memorize the actions (if there was)
6. If flag EndOfPhase2 -> continue to status **EndOfGame**
7. If flag EndOfMove -> set ActualPlayer to the next player
8. If flag EndOfPhase1 -> continue to status **BeforePhase2**
9. Else continue to status **InPhase1-BeforeMove**

F. status **BeforePhase2** (set up board for Phase 2)

1. set invisible the CollectTiles icon on board
2. set visible the icons for each tile type on the board for getting tile type declaration from players:
(herring, salmon, seal, polarbear, reindeer, igloo)
3. continue to status **InPhase2-CollectOnIgloo**

G. status **InPhase2-CollectOnIgloo** (collect one tile from the igloo)

1. If ActualPlayer hasn't got meeple on igloo -> continue to status **InPhase2-Evaluation**
2. remove ActualPlayer's one meeple from igloo
3. remove tile underneath the removed meeple and move it to the tile stack of the ActualPlayer
4. continue to status **InPhase2-BeforeDeclaration**

H. status **InPhase2-BeforeDeclaration**

1. instruct ActualPlayer to declare its next flip, choose one of the following:
(herring, salmon, seal, polarbear, reindeer, igloo)
2. wait for request

I. status **InPhase2-BeforeMove**

1. instruct ActualPlayer to flip one tile
2. wait for request

J. status **InPhase2-ProcessMove**

- receive move from player: (ClickedElement (Tile or Icon), Request)
1. If Request is DeclareNextTileType AND ClickedElement is valid:
 - o -> set Declaration
 - o -> mark Declaration on board
 - o -> continue to status **InPhase2-BeforeMove**
 2. If Request is to flip a face-down tile up AND Declaration is set
 - o -> flag RequestToFlip
 - o -> flip the clicked tile face-up
 - o -> continue to status **InPhase2-Evaluation**
 3. If Declaration is set -> continue to status **InPhase2-BeforeMove**
 4. Else -> continue to status **InPhase2-BeforeDeclaration**

K. status **InPhase2-Evaluation** (evaluate status after move)

1. clear evaluation flags
2. If all tiles on table are face-up
OR ClickedElement tile is the last reindeer
OR there is no more meeple on the igloo
-> set flag EndOfPhase2
3. If NOT ClickedElement tile -> set flag EndOfMove
4. Else If ClickedElement tile is the same as Declaration -> set flag CorrectDeclaration
5. Else -> set flag EndOfMove
6. continue to status **InPhase2-Execution**

L. status **InPhase2-Execution** (execute actions based on evaluation)

1. If ClickedElement tile -> wait some time that each player can memorize the last tile flip
2. If flag CorrectDeclaration -> move tile ClickedElement to player's stack
3. wait some time that each player can memorize the actions (if there was)
4. If flag EndOfPhase2 -> continue to status **EndOfGame**
5. Else If flag EndOfMove -> set ActualPlayer to the next player
6. continue to status **InPhase2-CollectOnelgloo**

M. status **EndOfGame** details:

1. Announce winner (most collected tiles)
2. Allow free tile flipping on tiles remaining on the table
3. Offer to restart the game
4. wait for request

N. status **EndOfGame-ProcessMove**

- receive move from player: (ClickedElement, Request)

1. If Request is RequestToRestart -> continue to status **BeforePhase1**

2. If Request is RequestToFlip -> flip ClickedElement

22 Feb 2021

addressed JSHint findings in the code:

- there is no static class property variable in ECMASCIPT 6, but only static methods
- semicolon placement

added skeleton structure to Readme

refined game play control pseudo code and transferred to Readme

23 Feb 2021

Implementing new parts of the controller using the pseudo code

24 Feb 2021

removed the static sun positions from the top of the board



created a single sun placeholder

calculate the positions of all 9 placeholders and put on the top of the window

here is the math for calculating the arch of the sun positions:

$$\frac{b^2}{4} = \left(\frac{b-m}{2}\right)^2 + (a-m)^2$$
$$r^2 = \left(\frac{b-m}{2}\right)^2 + (r-(a-m))^2$$
$$2r(a-m) = \left(\frac{b-m}{2}\right)^2 + (a+m)^2$$
$$r = \frac{\left(\frac{b-m}{2}\right)^2 + (a-m)^2}{2(a-m)}$$
$$r = \frac{(b-m)^2}{4} / 2 / (a-m) + (a-m) / 2$$

endresult:



split the javascript code into 4 files

this will be good for the declaration buttons: Radio Buttons 3D <https://codepen.io/andreasnylin/pen/agiDp>

27 Feb 2021

refactoring

27 Feb 2021

refactored the code again: removed the Classes and distributed the code between the gameController and gameViewer objects

added sound to certain activities, sources:

polar bear: <https://www.youtube.com/watch?v=VbfvQAtKC7w>

seal: <https://www.youtube.com/watch?v=1haAWHO8aCo>

salmon: <https://www.youtube.com/watch?v=SoA1CpHNhTY>

herring: <https://royalsocietypublishing.org/doi/suppl/10.1098/rsbl.2003.0107>

ice: <https://www.youtube.com/watch?v=v3O9vNi-dkA&t=20s>

28 Feb 2021

coded setting the number of tiles dependent on parameters to ease testing

implemented 3D effect on radio button input elements using the idea from here:

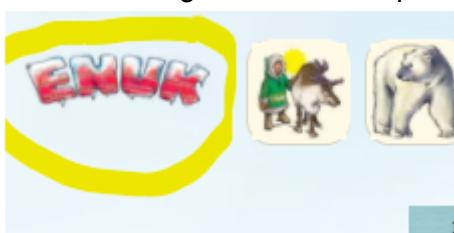
<https://codepen.io/andreasnylin/pen/agiDp>



01 Mar 2021

worked more on the 3D effect of the radio buttons, switched the emphasis to the selected one - selected button pops out

coded title background size and position change for the start of phase 2



coded state handling 2nd phase - tile declaration and tile flip - to do: compare declared and flipped tile

02 Mar 2021

coded state handling 2nd phase - Evaluation

bug fix: during unit test debugging noticed, that clicking on a tile declaration icon fires two click events. Reading on the internet suggested that, as my solution is a label and radio button, clicking on the label fires two click events. One for the label and one for the input element. To avoid this double event, the event listener needs to be placed on the input element directly. See: <https://stackoverflow.com/questions/43174996/javascript-click-event-triggers-twice>

added restart icon to the EndOfGame state:



completed the state machine logic

02 Mar 2021

the bug hunt begins:

- found at one place the Set's length tested, but sets does not have length, but size
- found out the null+1 equals 1, and null>=0, what a shocking development!
- clicking on a radio button while clicking is forbidden still executes the standard radio button event, unless event.preventDefault() is called

started working on the start page, which lists the rules of the game and let set some parameters

03 Mar 2021

working on the Rules page

replaced the ice sound, because my daughter did not like it, it was a sound of flexing ice on a lake, replaced with ice cube dropped into a glass: 1s from here <https://www.youtube.com/watch?v=g6vncdw64BI&t=2017s>

remaining tasks:

- write one player agent

07 May 2021

Corrected responsivity issues on the Rules page, almost all images needed call img-fluid.

Added a left-right infinite animated movement of the green play button on Rules page

08 May 2021

working on the Readme.md file and linked other planning documents

page structure:

