

# **Requirements and Analysis Document (RAD) for Dance Fans**

Authors:

Hedy Pettersson, Jakob Persson, Joar Granström,

David Salmo, Johan Berg, Isabelle Ermeryd Tankred

Date: 24 October

Version: 3

Chalmers Tekniska Högskola 2021

# 1 Introduction

The project aim is to create a fun, top-down, 2D strategy game in which two players play as dancers who compete in gaining the most dance fans on the dance floor.

The game will be a turn-based, two player game played locally on one device. Players take turns making dance moves with their dancer to win over dance fans who join them on the dance floor. The players get to choose dance moves using a card-based system, where each kind of dancer has a specific deck of cards that match the characteristics of that dancer. By playing your cards right you win over more and more dance fans on the dancefloor, until the dancefloor is either full of dancers or enough rounds have passed.

The stakeholders for this project are the members of the group. Joar came up with the initial game idea and he had a thought about creating a game like this before the course.

Since the project is a game the people who would benefit from it would be people who enjoy playing games. For this game in particular maybe people who enjoy strategy games and playing with other people.

## 1.1 Definitions, acronyms, and abbreviations

**Player** = a user of the program and their corresponding representation in-game.

**Main Dancer** = The dancer the user controls to make dance moves in order to gain dance fans and win the game.

**Dance Fan** = One of the dance fans, which join the dance floor inspired by the Main Dancers dance moves. The player with the most dance fans at the end of the game wins.

**Dancer** = A dance fan or a main dancer

**Dance floor** = The world in which a dancer can be located and move around within, in the game.

**Dance move** = A pattern and a set number of steps a player can move across the dance floor. The pattern tells us how the dance floor will change after the dance move is performed. The dance move inspires dance fans to join in on the player's dance team. These dance moves are represented in the game as cards in a deck of cards (a main dancer's repertoire of dance moves).

**Dance pattern** = The pattern part of the dance move.

**Card** = Representation of dance moves in the game, to help the player choose strategy.

**Preview** = A special view of what the dance floor would look like if the player performed the dance move of the currently selected card at the location the selection marker is currently located on the dance floor.

## 2 Requirements

### 2.1 User Stories

#### **Dancefloor**

Implemented: yes

##### **Description:**

As a player I want a window which displays the danceFloor so that I can understand and think about how to play the game.

##### **Functional requirements:**

-When the game starts, the dance floor is on screen.

##### **Non functional requirements:**

-The dance floor should be displayed on screen with 54 tiles or squares.

##### **Acceptance Criteria:**

- DanceFloor is fully visible on the screen.

#### **Main dancer**

Implemented: yes

##### **Description:**

As a player I want a main dancer so that I can tell where I am during the game.

##### **Functional requirements:**

-The games contain two main dancers, one for each player ; both should be displayed once the games started.

##### **Non functional requirements:**

-The main dancers look different in terms of shape and color.

##### **Acceptance Criteria:**

- Both players Main dancers are visible on the dance floor.
- They should look different so that the players don't get confused.

#### **Movement**

Implemented: yes

##### **Description:**

As a player I want to be able to move my main dancer so that I can affect different areas of the dance floor.

##### **Functional requirements:**

-When the user clicks on keyboard arrows, the main dancer moves one step based on the arrow's direction.

##### **Non functional requirements:**

-Main dancer moves using keyboard arrows.

-The main dancer moves only on the dance floor(54 tiles); it must stop when it touches the edges of the dance floor.

##### **Security:**

Can the other player move my main dancer when it's their turn.

**Acceptance Criteria:**

- When a player presses certain buttons the player's main dancer moves.
- A Main dancer can not move outside of the dance floor.
- The buttons used should be commonly used buttons.
- It should not be possible to interact with another players' main dancer.

**Choosing cards**

Implemented:yes

**Description:**

As a player I want to be able to choose between two or more cards so that I can choose what best fits with the situation and improve my chances of winning.

**Functional requirements:**

-When a player chooses a card, the dance fans will be distributed according to the shape that is on that card.

**Non functional requirements:**

-Two or more cards are displayed to choose between so the players can compete.

**Acceptance Criteria:**

- Each player should have their own card deck.
- The cards that the player has "on hand" are visible on the screen.
- More than one card should be "on hand".
- The player can see which card is currently selected.
- New cards should replace the cards as they are used.

**Preview**

Implemented: yes

**Description:**

As a player I want to be able to see how my move would affect the dance floor so that I can compare the different possible options I have in order to make it easier to make the best possible decision.

**Functional requirements:**

-When a player chooses a card, the dance fans will be distributed as transparent occupant tiles based on the shape that is on that card.

-The player can see a transparent representation of the card on the dance floor before deciding how to play.

**Non functional requirements:**

- Preview of card changes when the user changes the card directly.
- The "Preview" should be identical to the card on the dance floor.
- The player should distinguish easily between transparent and existing tiles.
- The main dancer gets back to its original position every time the card is changed.

**Acceptance Criteria:**

- A preview which shows how the dance floor would be affected is visible.
- When the player changes the selected card, the preview is also changed.

## Finished the game

Implemented: yes

### Description:

As a player I want to know when the game is finished and which player has won so that I know if I won or not.

### Functional requirements:

-If the game is finished, a message will be displayed for the player with the most occupant squares on the dance floor.

### Non functional requirements:

-The displayed message should be red/green based on the winner's color or mixed if no one wins(equal occupant tiles).

### Acceptance Criteria:

- When the Dance floor is filled with Dance Fan:s the game ends.
- After a certain amount of rounds the game ends even if there are still empty tiles on the Dance floor.
- When the game ends there is an indicator of who won.

## Help interacting

Implemented: yes

### Description:

As a player I want it to be clear how to interact with the game so that I can be certain that what I intended to happen is what actually happens and therefore make me less confused and frustrated.

### Functional requirements:

-When users start the game, an image with instructions should be displayed to the left of the game.

### Non functional requirements:

-The image contains understandable instructions for playing the game. For example when and how the player wins.

### Acceptance Criteria:

- There should be explanations in the window which shows which buttons are used and what they do.

## Guiding the movement

Implemented: no

### Description:

As a player I want the game to clearly indicate where I can move my main dancer so that I do not get confused or frustrated.

### Functional requirements:

-Indicator shows player where can move the main dancer

### Non functional requirements:

-The indicator should be clear and easy to follow.

**Acceptance Criteria:**

- Main dancer can not move further than the amount of steps given by the dance move.
- When the player changes between cards the main dancer should be moved back to the position it started the round on from its position in the preview.
- Some type of visual indicator of where the Main dancer can move.

**Hiding hand**

Implemented: yes

**Description:**

As one of two players I want my cards on hand to be hidden unless it is my turn so that the other player does not know what my possible moves are, which will make it more difficult for them to win against me.

**Functional requirements:**

-Player's cards are invisible until it is the player's turn

**Non functional requirements:**

-The cards are seen only for when one of the players plays.

**Acceptance Criteria:**

- After a player has made their move the next player's cards on hand are hidden.
- Some type of input is needed to make the cards visible again.

**Resizable**

Implemented: yes

**Description:**

As a user I want to be able to resize the window of the game so that it fits my screen according to my needs.

**Functional requirements:**

-The size of the game will change dynamically when the game window is resized

**Non functional requirements:**

-The game should work correctly once the window is resized.

**Acceptance Criteria:**

- The game scales proportionally when one changes the size of the window.
- The layout should remain the same so the user still knows where everything is.

**Restart**

Implemented: yes

**Description:**

As two players we want to be able to start a new game easily so that we can compete again.

**Functional requirements:**

-When the game is over the player can press a button and the game restarts.

**Non functional requirements:**

-The game should work perfectly every time it is restarted and the main dancers should be on the same tile every time.

**Acceptance Criteria:**

- A new game can be started, by pressing the key "D", without restarting the program.

## **More main dancers**

Implemented: no

**Description:**

As a player I want to be able to choose between different main dancers so that I can choose the one I visually or strategically like the best.

**Functional requirements:**

-The game should have settings where the player can change the color or the image of the main dancer to another preferred one.

**Non functional requirements:**

-In settings there is a button once it is clicked it shows all available main dancers.

**Security:**

Can two players choose the same shape and color in the same game?

**Acceptance Criteria:**

- There should be more than 2 Main dancers to choose from.
- Before the game starts the players get to choose which Main dancer they want to play as.
- Once a color and shape is chosen, those alternatives are locked for the other player.

## **Move cards**

Implemented:yes

**Description:**

As a player I want to have many cards in my deck so that the game is more varied and I get more options.

**Functional requirements:**

-More cards with new patterns.

**Non functional requirements:**

-The patterns and cards should be displayed correctly on screen.

-Both players have the same probability to get the new cards

**Acceptance Criteria:**

- A Card deck should consist of at least 10 cards.

## **Starting screen**

Implemented:no

**Description:**

As a user I want a start screen so that it is clear when the game has started.

**Functional requirements:**

-When the user starts the game, a start screen should appear before starting the game.

**Non functional requirements:**

-The start screen should suit the game and not affect the game.

**Acceptance Criteria:**

- A start screen exists
- It is shown when the program is started
- It is possible to move to the game from this

## UI

Implemented: yes

**Description:**

As a user I want the things that are displayed to be displayed in a way that seems reasonable in order to not be confused.

**Functional requirements:**

-Dance floor, Cards, Deck and instruction image are distributed in a suitable way.

**Non functional requirements:**

-The user should find the game is easy to play and follow game instructions.

**Acceptance Criteria:**

- Reasonable proportions to the things that are displayed
- No overlapping unless there is a good reason

## Settings

Implemented: no

**Description:**

As a user I want to be able to access settings to change the game to my preference.

**Functional requirements:**

-When a user clicks on settings I will be able to change them.

**Non functional requirements:**

-The game has a settings button where users can change some settings .e.g. change the color of the main dancer and dance fans .

**Acceptance Criteria:**

- Settings can be displayed and changed (currently, view help or not could be in the settings, more to add later potentially)
- The user can access the settings during the game.

Link to our trello board:

<https://trello.com/b/AgzbahLo/project>



## 2.2 Definition of Done

All criterias of the user story have been met. Should be well commented so that others can understand. Before accepting a pull request, at least one person who participated in designing the feature needs to review the implementation. Relevant tests have been written.

This definition is the one we would have liked to follow, but we ended up only following it partially. We realized later that the user stories needed to be rewritten when it was difficult to implement things following them, (being too large / unspecific, also depending on each other a too much). We also had a lot of problems getting the project started, and had to rush many parts, which meant that a lot of documentation and testing had to be postponed to later. We have mostly followed the pull request criteria where another person has to look at the code before approving it, but it is possible for one person to push directly to main and also to both create a pull request and approve it by themselves. We should change the settings in github to solve any future mistakes.

## 2.3 User interface

These images represented the initial idea of the game. Some rules changed during development.

See Appendix 1.1

The player chooses between different cards representing dance moves that the main dancer can perform. In Appendix 1.1 you can see how the preview of the dancefloor changes depending on which dance move card is selected.

See Appendix 1.2

When the player decides to use a card and confirms it, those tiles on the floor are confirmed in the color of the main dancer that performed the dance move. In the theme of the game this means that more dancers came up on the dance floor because they got inspired by the main dancer's dance moves!

See Appendix 1.3

Their turn is now over and the dance move card used is placed in the discard pile. Thematically this makes sense since the main dancer wants to vary dance moves to impress potential dance fans more. The cards come back a few turns later however! Later when it is that player's turn again, it draws new cards from its main dancer's drawpile, a deck of dance move cards.

See Appendix 1.4

Now it is the next main dancer's turn (the red main dancer) who has different kinds of cards (shown at the top of the image) representing that dancer's dance style.

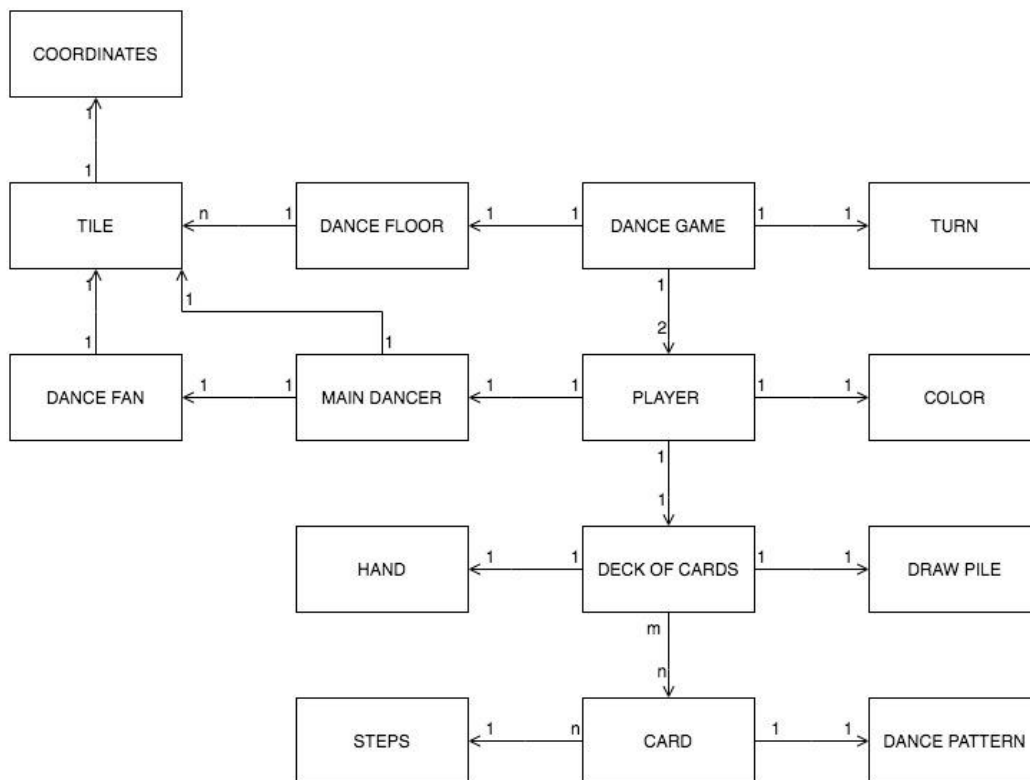
See Appendix 1.5

When the red main dancer does its dance move with a pattern covering the green dance fans, they are overtaken to the red players team and color. This has similarities to games like Othello and Go.

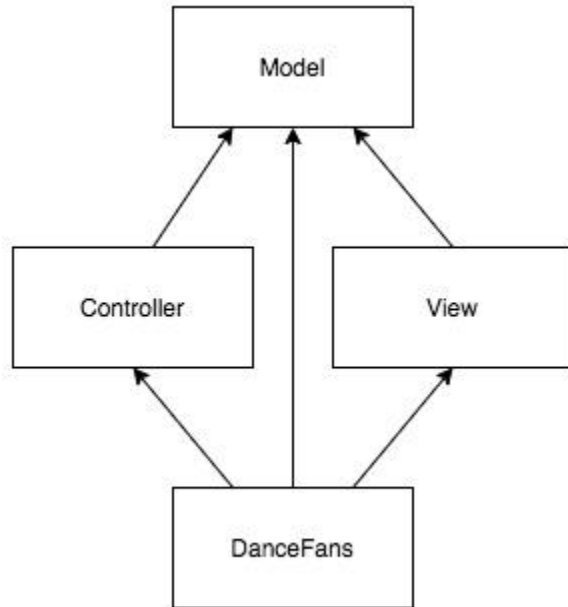
See Appendix 1.6

When a certain amount of turns has passed (thematically that the song has ended) or if the whole dance floor is filled with dancers, the player who has the most dancers of her color on the dance floor wins!

### 3 Domain Model



### 3.1 Class Responsibilities



See appendix 1.7 for an UML class diagram.

DanceFloor contains a matrix of tiles. It is also responsible for changing the details of the specific tiles. Also handles counting what is on the tiles.

DanceFloorTiles contains information about what is currently standing on the tile, and can tell us more details about the object on it.

Coordinates are used to navigate and keep track of locations in the danceFloor and consist of two ints.

CardDeck contains multiple lists of card:s, and in what order they are in. The different lists are the currently opened cards, the used cards, the unused cards, and a copy of the original list which was used to create the deck in the first place. It can shuffle card:s in a list, and also handles moving the cards between the different lists.

Card contains a matrix which is used to represent the dance pattern, an int for the amount of steps and an image representation of the card. As of now the cards in the game have been manually added in, but automating the process is something we would like to do in the future.

FloorObject has coordinates and keeps track of different aspects of the object for now. (Like color and type)

Dancer extends FloorObject but does not do anything special as of now, but we think they might have some more specific behaviour later.

MainDancer extends Dancer and also has an extra set of coordinates, which are for its temporary position. Representation of the player on the dancefloor.

DanceFan extends Dancer, more functionality seems likely to be added in the future.

Player has a CardDeck, a MainDancer, and two DanceFans as of now. Is what ties together many of the parts.

## 4 References

List all references to external tools, platforms, libraries, papers, etc.

<https://github.com/libgdx/libgdx>

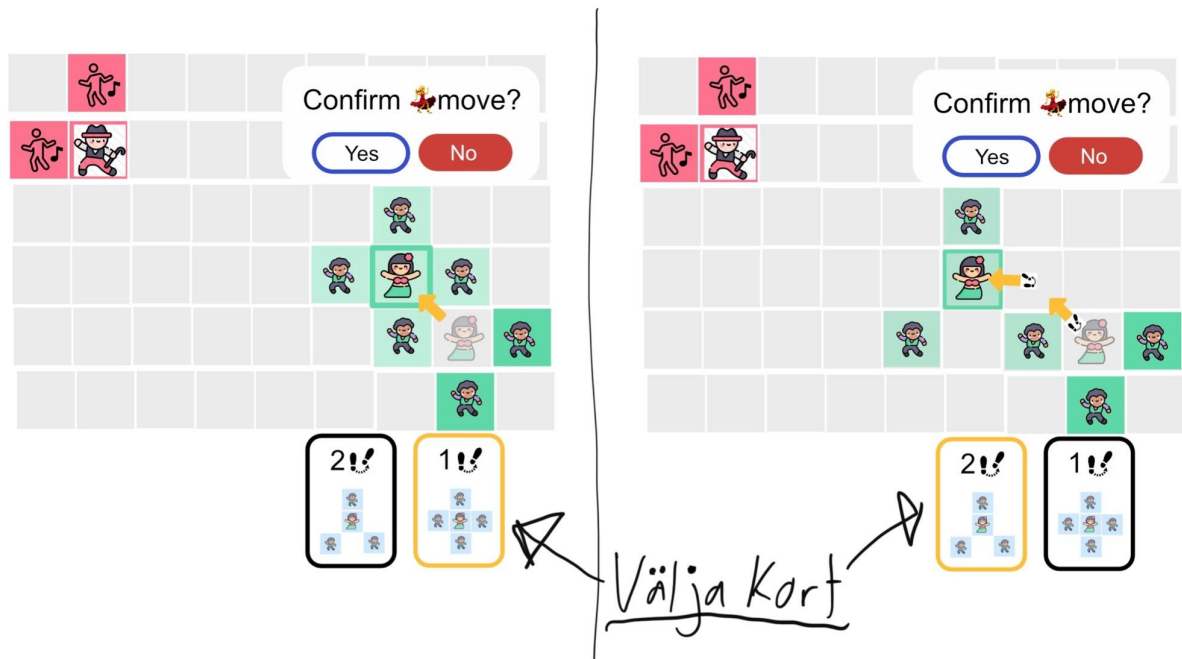
<https://www.figma.com/>

<https://www.codeandweb.com/texturepacker>

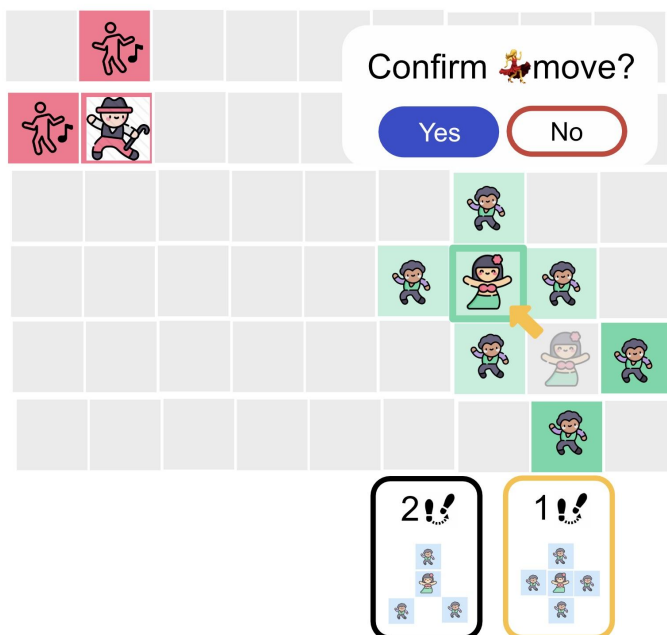
Dancers images made by Freepik from [www.flaticon.com](http://www.flaticon.com)

# Appendix

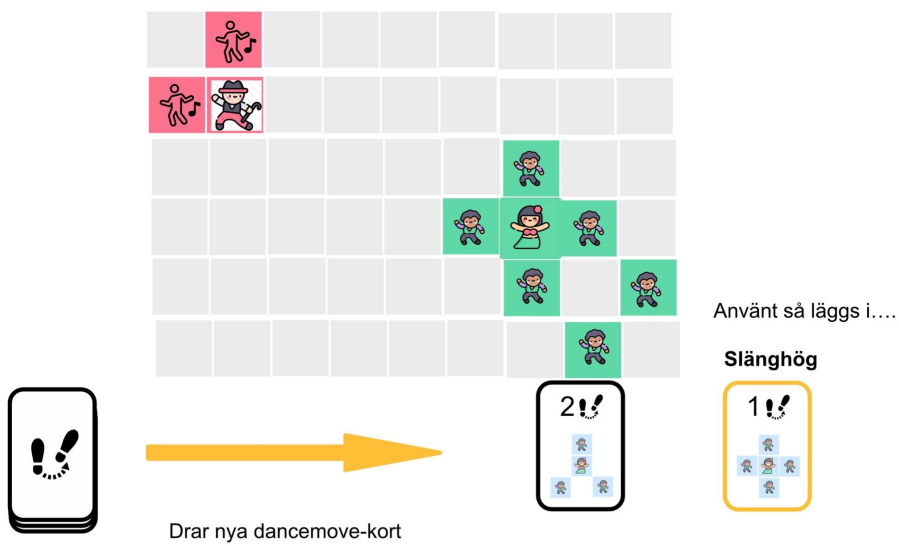
## 1.1



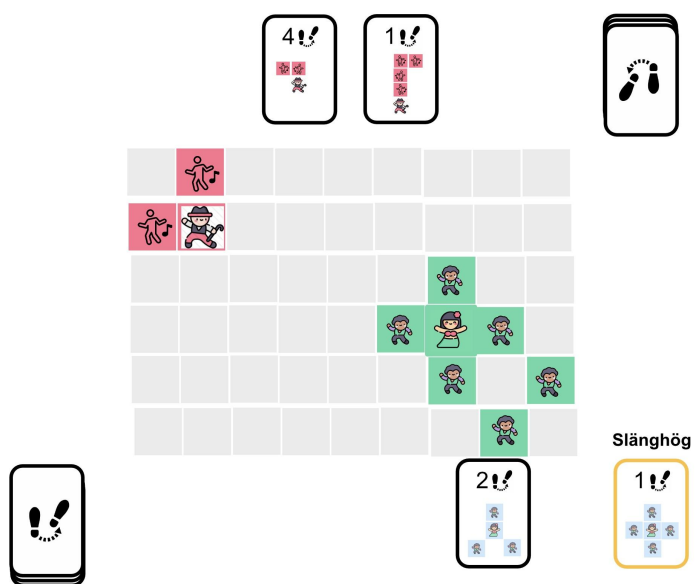
## 1.2



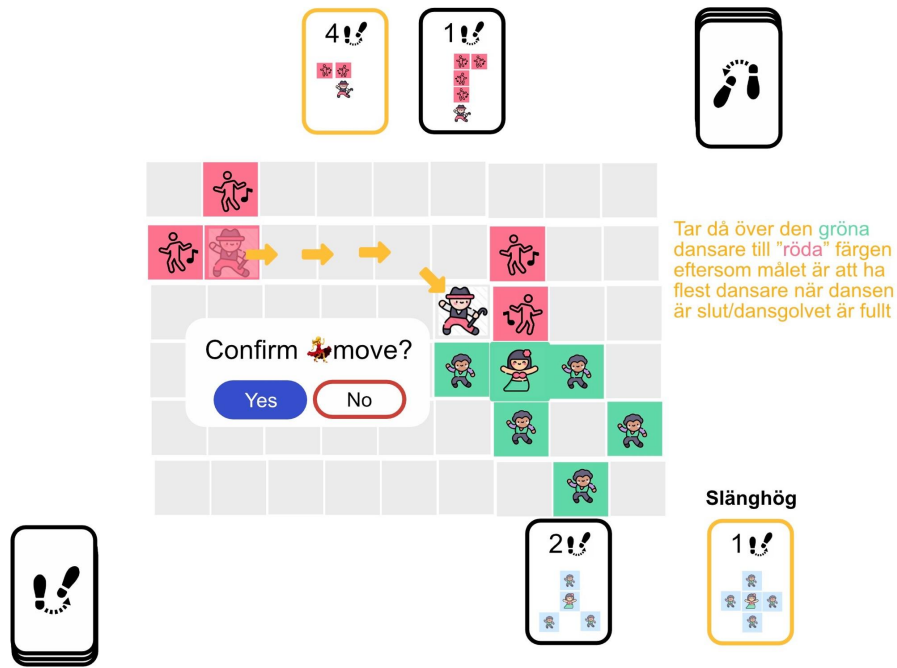
1.3



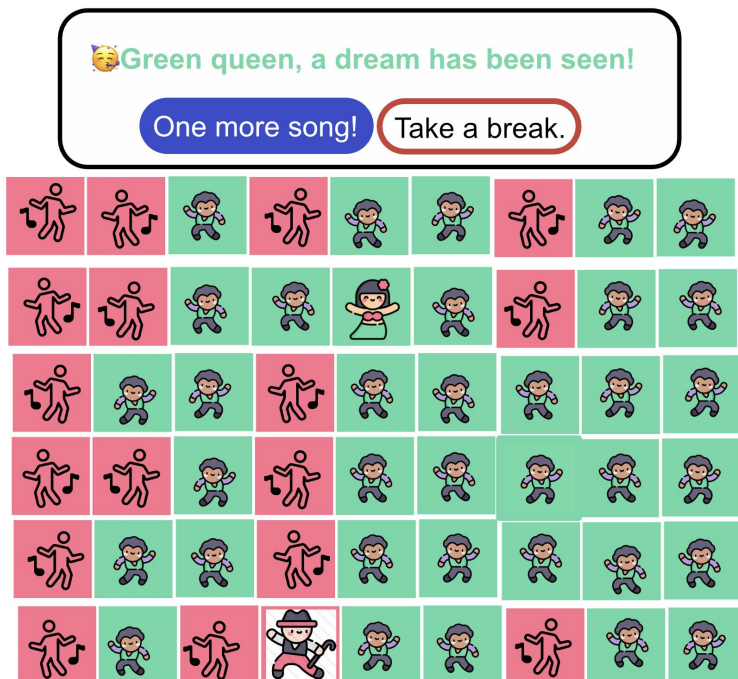
1.4



1.5



1.6



# 1.7

