Initial Research

* consist of **several different views** of the same model
  + View of the cards when they are auctioned
  + Views of the cards when they are owned by one player
  + Selection of game mode per round (e.g. painting auction or player auction, maybe additional modes like that a random painting can be sold for a random price at one point)
* have **interaction** on the model data including the possibility to change the data (not only display it)
  + When auctioning, drag and drop money to the auction table/picture to bet money
  + Settings where the standard values or images (e.g. art or picture library) can be changed
  + Additional features: Select own images from the phone
* use a front-end framework (e.g. [Angular](https://angularjs.org/), [Backbone](http://backbonejs.org/), [Ember](http://emberjs.com/), [Meteor](https://www.meteor.com/)) in case of web solution that supports some MVC (or similar) architecture, or Java FX in case of Java. Android by default supports MVC
  + We use Android, so we have MVC
* use some external data through REST API (in case you are working on proposed ideas, you will have to extend the ideas to support this)
  + Pixabay free images API
    - <https://pixabay.com/api/docs/>
    - Include logo Creative Commons
  + Bing / Google Image Search API
    - Google custom search API <https://developers.google.com/custom-search/docs/overview>
    - <https://www.google.se/search?as_st=y&tbm=isch&hl=de&as_q=famous+artwork&as_epq=&as_oq=&as_eq=&cr=&as_sitesearch=&tbs=isz:lt,islt:svga,iar:t>
    - <http://stackoverflow.com/questions/26732530/how-do-i-return-json-results-from-bing-search-engine-api>
  + BETA: Artworks API <https://developers.artsy.net/docs/artworks>
* persist the data either by using an API that supports it (e.g. [Firebase](https://www.firebase.com/)) or by having a minimal backend (we will not evaluate the architecture of the backend)
  + Firebase for syncing the player/game status on the web - can work with a game code to join friends
  + Ensure that application does not end/crash when minimized/paused
* in case of your own project idea, make sure it is complex enough so that the work can be split between the group members

Agenda:

* Define MVP
  + Brainstorm features
  + Draft technical setup including architecture
* Sketch MVP together (including screens)
* Discuss task packages and how to distribute it
  + Firebase Setup
  + Game Design
  + UX Design

Question:

* Hosting?
* model imports no graphics??? Is our setup ok?