Plans

- Check games Angrybird Dream Blask & Empires and Puzzles
- Plan prototype & think through questions 1-4
- Game ideation
    - blast game core prototype like Angrybird Dream Blask
    - pinch-zoom panning menu like base building UI in Empires and Puzzles
- Setup environment (latest Unity, version control, ide)
- Implement prototype
- Synch prototype final state with plans
- Deliver zip file or git repo

Planning of Features & tasks

- Basic graphic
    - Tiles
    - Level
    - powerup/combined tiles
    - UI elements
        - Level goal
        - Moves
        - Victory-out of moves popup
        - Ingame currency
    - Progressing main screen
- Basic level
- Basic tiles with collision & physics
- Collectible tiles
- Tile spawner
- Level goal
- Level goal target collection fx
- Moves
- Reshuffle on stuck
- Powerup/combined tile (2-3 combination)
    - Combine more collected tiles
    - Combine powerup/combined tiles
    - Blast effect on tiles no collection
- Victory & out of moves popup
- Create levels and give it randomly/sequential
- Main screen
- Pinch-zoom panning main screen
- Ingame currency
    - Store state in player prefs
- Progress main screen by currency
    - Add extra element on enough gold on enter the main screen
        - Currency spend on extra element fx
    - Store state in player prefs

First iteration

- Setup environment
- Basic graphic
- Basic level
- Basic tiles with collision & physics
- Collectible tiles
- Tile spawner
    - Limit by level
    - Blockable by tiles
    - Kick effect on spawned elements (configurable in prefab with direction & multiplier)

Second iteration

- Level goal gui
- Moves gui
- Clusterize - combine vegetables
    - 1. level combination - 4 vegetables
    - 2. level combination - 5 vegetables
    - 3. level combination - 6 vegetables
- Powerup/combined tile (2-3 combination)
    - Combine more collected tiles
    - Combine powerup/combined tiles
    - Blast effect on tiles
- Polishes
    - Animations
    - Music
    - Levels
    - Bug fixes
    - Tile bouncinesses
    - Colliders
    - Spawner configurations

For any further questions or explanation please contact me as I didn’t have time to document the code.

I can send some code explanation later if needed or we can discuss it during a meeting.

Answering the questions (Empires and Puzzle)

1. What could be the best backend solution for this kind of game? And why? Please be aware you should consider cost, time, resources, and all other things as a part of a small team.
    - I think Firebase could be a good choice.
      It's widely used so it's been tested by the time, it's reliable, well documented and there's a huge community to ask help for during the development.
      It's easy to integrate into Unity projects and to develop.
      It provides a safe platform to handle users, store database or collect data & analyse it.
2. List possible problems and solutions for game security.
    - Hacking (steal code, manipulate data or network packets)
        - Obfuscate the codes
        - Implement business logic on server
        - Validate client requests
        - Use secured backend solution
    - Cheating (change local data or mock phone settings, cheating bot / software)
        - Validate and sync with server
        - Anti cheat script to detect bot or software help
    - Bugs on purchases or collecting in game items
        - Identify and fix such bugs
        - Validate such actions on server (or both on client) side
3. Which action needs to be verified with the server? And why
    - Changes that are related to game state / player inventory change like currency spent, progress or usage of in game consumables.
      The local client can be cheated or have incorrect data so the server needs to sync with it and verify it as the server is trusted.
    - Actions which might have conditions where the server should decide same for sync and security reasons.
    - Any authentication or purchase.
4. Hard currency or inventory item used by the player! If you want to support this kind of action not only in online mode but also in offline mode, what could be a solution?
    - Then there's a need to store the same data on the client side and sync it each time with the server as soon as possible. For example, store the hard currency and inventory state.

Rough time spent estimation

- Documentation (analyse games & questions, plan prototype, document the process) ~2h
- Implement, test & polish prototype
  ~14h-16h

Sándor Trombitás