Game Design Document (GDD) Template

April 2022

GDD? What’s the purpose?

- Before starting development for a playable ad there are alot of unknowns, this document will help you as a Developer/Game designer/UA to ask yourself and your team what are those unknowns and how should they be dealt with

- Once the unknowns are flushed out and discussed in this document the production plan will be clearer for whomever is involved

- When this document is full with details, production kickoff can commence and it will be the goto document for anyone who works on this playable

- The content of this GDD is vast and not necessarily needed in its entirety - You should feel free to just use the slides you feel are a good fit for your team and projects

➕ Add app store links for the specific game!
⬇ The team should Download & play the game!

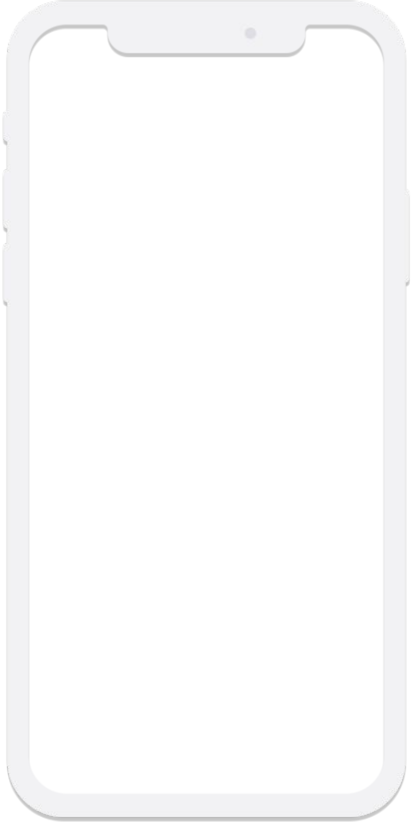
Apple Store

Android Store

Reference

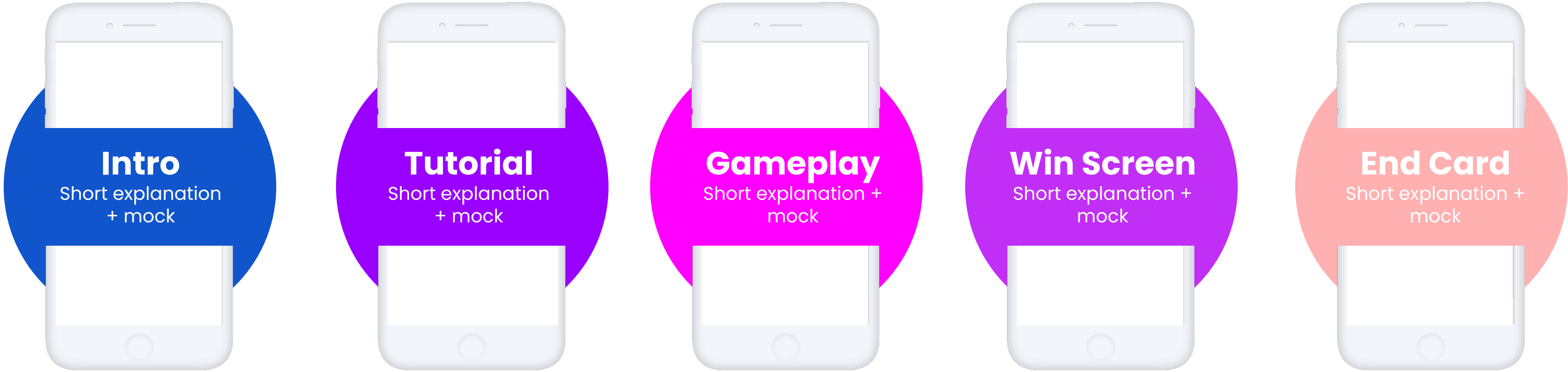
- Add specific references that help the team understand the inspiration for this playable

- Design style, specific layout, game mechanic, or concept are all great starting points as references



Gameflow mocks

Show the entire flow of the game with mocks - this should act as a summary for designers & developers



Intro
Short explanation + mock

Tutorial
Short explanation + mock

Gameplay
Short explanation + mock

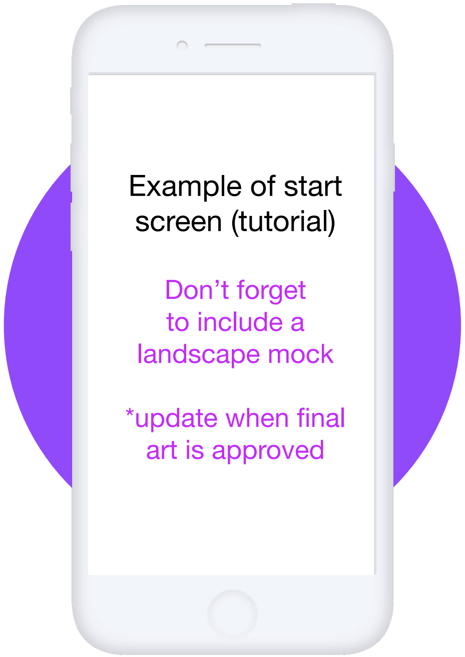
Win Screen
Short explanation + mock

End Card
Short explanation + mock

Asset folder (link here) | Approved art (link here)

*make sure to include fonts and loading screen assets

Scenario - Start screen (intro/tutorial)



Example of start screen (tutorial)

Don’t forget to include a landscape mock

*update when final art is approved

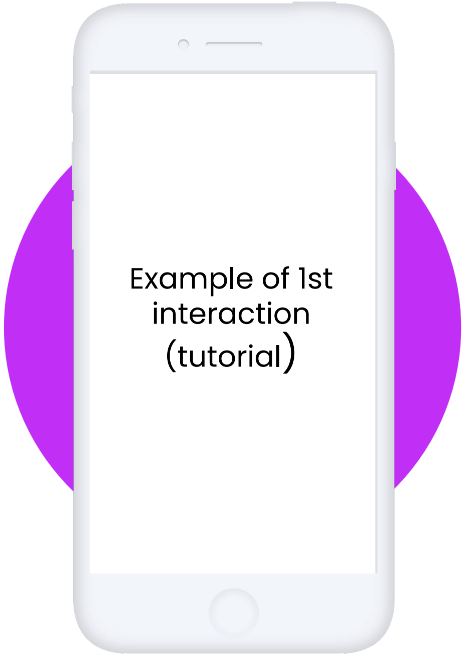
- The first screen of the playable starts with this screen
- The playable does not start before the user interacts with it
- Transition to the next screen should occur this way:
  - A
  - B
  - C

Engagement settings (different settings that can be changed in Playground to create new variations of the playable):

- Intro (start screen): on/off
- Character: on/off
- Timer: on/off
- Dark overlay: on/off

Friendly tip: Keep the intro between 1 - 1.5 seconds and always test a version without an intro

Scenario - 1st interaction



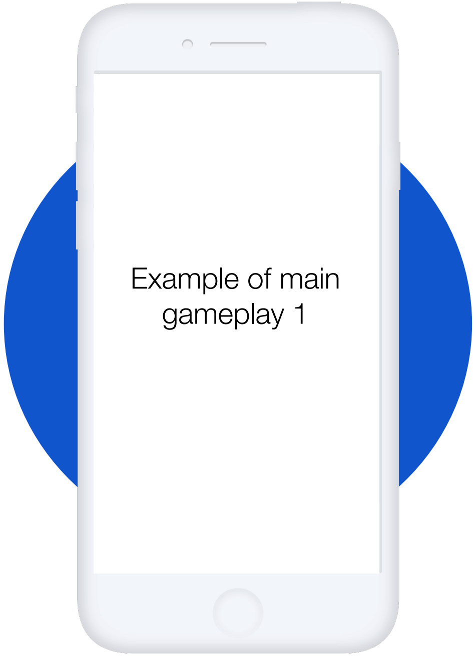
- CTA text “lorem ipsum”
- Hand gesture animation (reference)
- When the user clicks, the playable starts
- Timer starts (if it exists)
- Playable logic (here is the place to add timings and defaults):
  - By default a hint should appear after xx seconds
  - By default, if player is idle for more than xx seconds, playable autocompletes
  - By default, if player made a wrong move, show a shake animation of the camera or do not allow illegal moves

Settings:

- Hand at the start: on / off
- Timer duration: on / off (timer duration = time to end playable)
- End the playable after 1 click at this screen: on / off
- Number of blocks between bottom shape and fail line

💡 **Friendly tip:** Make sure this screen acts as a good hook so players want to interact with it

Scenario - main gameplay #1

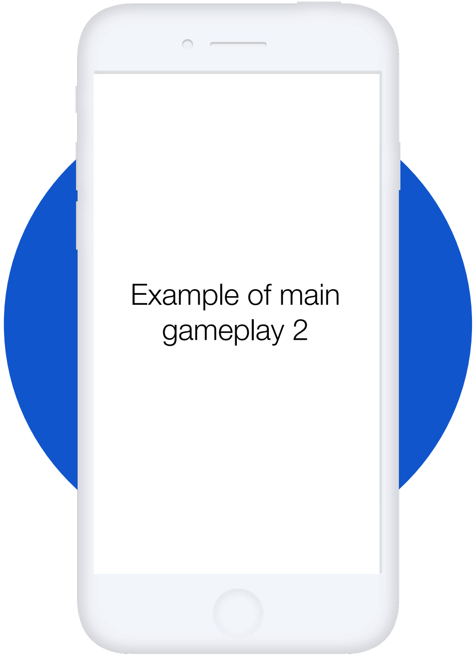


Example description of gameplay for a runner game:

- When player drags finger, character moves accordingly from left to right and automatically runs forward (runner mechanic)
- There are several obstacles along the way
- If the player runs into an obstacle in the first 7 seconds, gets an extra life (character blinks opaque and continues to run)
- If the player hits an obstacle after 7 seconds, sees the fail screen

The better you describe and show in the mocks what your vision is, the easier it is for designers and developers to execute

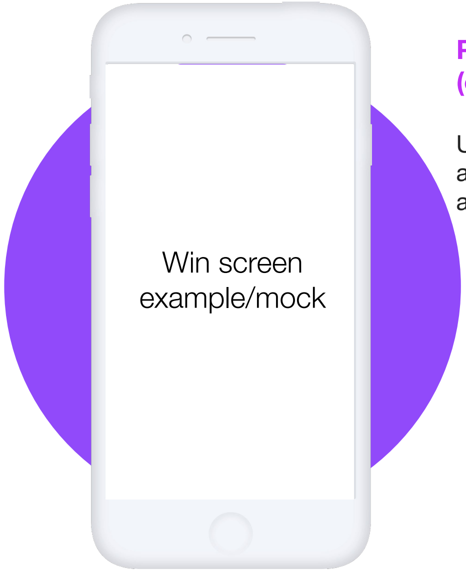
Scenario - main gameplay #2



Gameplay Settings:

- Hand at the start: on / off
- Timer duration: on / off (timer duration = time to end playable)
- End the playable after 1 click at this screen: on / off
- Number of blocks between bottom shape and fail line
- If player is idle for xx seconds, autocomplete the playable and show the end card

Success



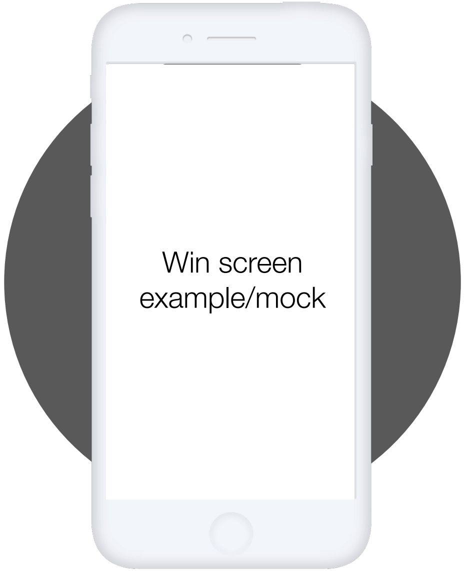
Playable can be “won” by users if (example):

Users fill all the empty cells before another one appears => endcard appears

Settings:

- Change “Play Now” text
- Retry redirects to store: on / off

Fail



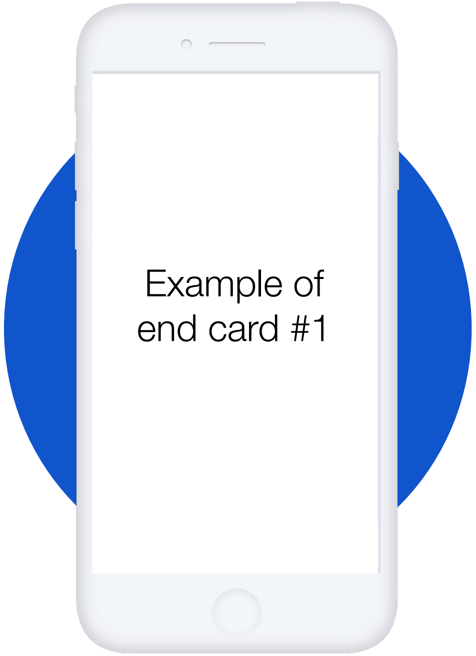
Playable is “lost” if (example):

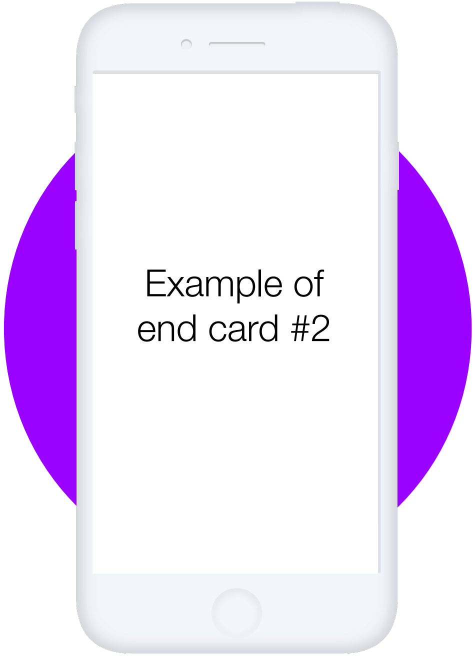
- Timer finishes and no empty cell reached the fail line => fail screen appears
- If the user fails a level, the background darkens (we still need to see the failed level in background)
- A fail message appears
- Play Now = redirects to store
- By default retry = the user can retry the failed level

Settings:

- Change “Play Now” text
- Retry redirects to store: on / off

Scenario - Endcard





End card #1

- Game logo
- Background #2

End card #2

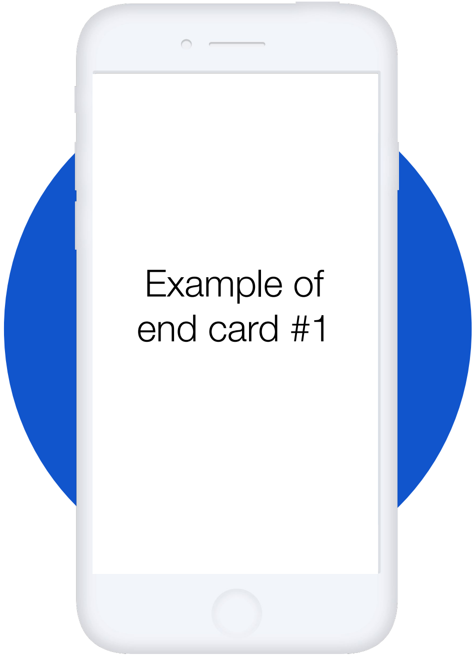
- Branded end card - match the design with the first image in the store

Conversion settings:

- Change “Play Now” text
- Change “Collect” text
- Change sum “100$”
- Disclaimer on - only on cash version
- Change avatar text and image (edit text, upload different image)
- Back from store show end card “X”

💡 **Friendly tip:** In Hyper Casual games - It may be better to test sending to store during gameplay (on networks that allow it)

Playable EndCard - PEC



We recommend creating a short version of the playable that can run after a video and not independently

- All playables must have a 1- or 2-tap PEC
- If 2-tap:
    - 1st tap on gameplay
    - 2nd tap on END CARD (should be defined as yes/no for less/more aggressive approach)

Why should you breakdown the art/animations tasks?

Breaking down the art/animation tasks into a list can benefit you, the artists, and the developers.
Each team can check the progress according to these breakdowns and make sure nothing is missing.

Note: this doesn't replace the need for a full list each team should create - instead it will serve as a high-level list for them to reference along the way

Animation tasks example:

- Particles fx after each move
- Character idle animation
- Character success animation
- Character fail animation
- Elements appear fx
- Transition to end card
- Timer
- Red vignette flashing

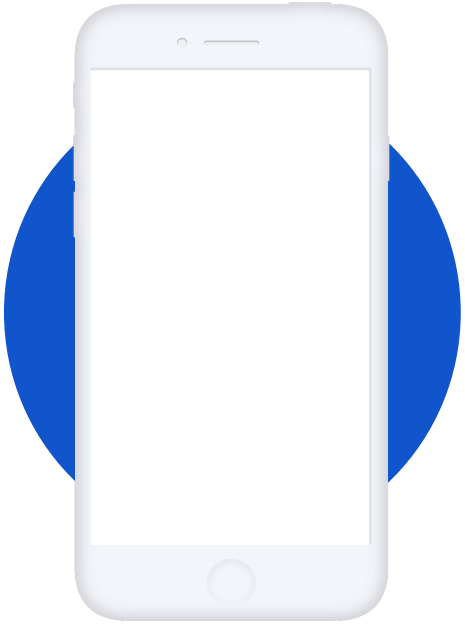
Art Tasks example:

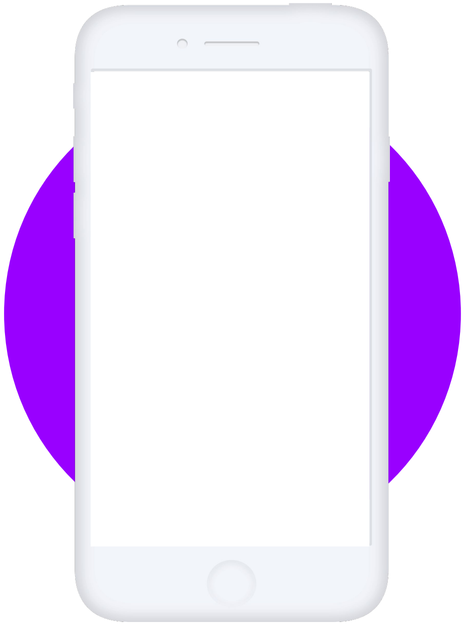
- Intro - 3 characters
- Card highlight
- Bid screen
- Main game field
- Elements and layout
- Player avatars, player cards, dark overlay, hand, bid counter
- CTA text: "your turn"
- Win screen
- "You won!" text

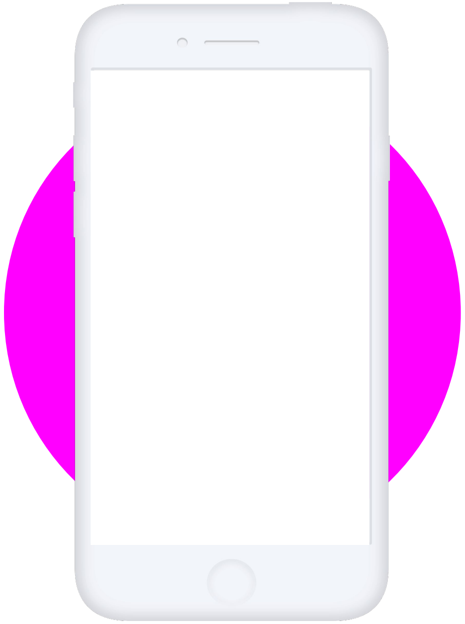
💡 **Friendly tip: Depending on your coding skills, you can also create the same breakdown for the developers - or with them**

Animation References

For example: card shaking animation, amazing screen particle systems, tutorial hand movement







Edge cases

- If player does wrong move (not according to hint)
- Lock first interaction to predefined or random

Defaults

- When users come back from store, they should see the end card (changes available - see settings here)
- ‘Move to store; setting should be after reaction by default (changes available - see settings here)
- If there’s no intro and playable starts with tutorial, hint should be present immediately. After the tutorial interaction, next hint should appear after 1 second - same for following hints
- If the user is idle for 6 seconds, autocomplete the playable and show the end card

Mid-version (Mvp)

Can be discussed during the planning process

- Main mechanic implemented (discuss with developer to understand the cost/effort)
- Main gameplay (static design) implemented
- Tutorial implemented
- Fail/win screens implemented
- Custom events implemented

Skippable for mid-version:

- Tutorial
- Final animations
- Auto complete

Date to deliver:

Sound table

Define after animations completed

Add link to sound folder here

Sound file	Event for trigger	Comment
BGM.mp3	Background music	
Place_bid.mp3	Bid chosen	
Get_trick.mp3	Cards fly towards the winner of the trick	
Place_card.mp3	Card put in place	
Spades_broken.mp3	Spades broken	
You_win.mp3	User won (need to check length) - confirm these sounds can run when end card appears	

Configurations - versions

It is best practice to set up a meeting with the performance manager and decide together what will the different versions include/exclude

Configs: (please make sure the 1st config is the priority. Development should first focus on this config for the mid-version) make sure each config has a descriptive name - see example on config 1 (version 1)		Version1_withintro_with_coins_3taps_no_Reaction	2_relevant_meaningful_name	3_relevant_meaningful_name	Version4_FG	5 (just pec)
Tutorial						
(intro)		yes	yes	no	yes	no
Game Play						
With coins		yes	no	yes	yes	yes
Conversion						
Send to store	Amount of taps	3	3	3	4	1
	reaction	no	yes	yes	yes	yes
Back from store		EC	EC	Last gameplay frame	EC	EC

Legend:

Reaction - after an interaction, show the animation before sending to store

FG - Full Game (user is sent to store only from end card) - will usually appear in version name at the top of the table

EC - End Card

PEC - Playable End Card

*no high ctr for all configs (except PEC configs)

Test ideas

The best way to divide your tests is into 3 categories:

1. Tests that check the engagement rate (start of the playable)

2. Tests that check the performance of the gameplay itself (amount of interactions, different layout, different color palette, different skin/theme, different mechanic)

3. Tests that check conversions (auto-send to store, different end card, fake next level, fake collect screen)

General settings that must be available

Engagement (add settings that should affect the 1st engagement rate)

- Autocomplete when idle 5 sec (on/off) (tutorial)
- Intro screen (on/off)
- Overlay on cards (on/off)
- Dark overlay entire screen (on off)

Gameplay (add settings that affect gameplay)

- Autocomplete when idle XX sec (gameplay)
- With bids (on/off) - when bids are off gameplay length should be according to bid “2” and bid counter should be turned off
- Bid screen pointer (on/off)
- Overlay on cards (on/off)

Conversion (add settings that affect conversion)

- Change end card button text
- Move to store after XX Amount of taps
- Move to store after reaction (on/off)
- Retry text (edit text)
- Retry redirects to store (on/off) - If retry redirects to the store is off, the retry button should let users replay the last move

Custom Events

Performance manager must approve this slide

- **interaction_1** - add specific explanation on what triggers this custom event (when user first taps on screen/when user first successfully taps on character etc…)

  OnTap/OnRelease/OnFingerLift…)
- **interaction_2**
- **interaction_3**
- **win/lose**
- **end card shown**

THANK YOU