

Variant Solitaire

CS327 Interdisciplinary Game Project, Fall 2019

Professors Ira Fay and Bassam Kurdali

Revision History

Date	Changes
9/4/19	Initial Version
9/17/19	Updated to Reflect Reactor
9/20/19	Updated to Reflect LNT
9/22/19	Updated terminology for programmers.
9/23/19	Added Touch Interactions section
9/24/19	Added screen flowchart.
10/20/19	Added rundown
10/21/19	Updated screen flow, added scoring, updated options screen, added move counter, removed flipping the deck causing ca
10/23/19	Updated screen flow again, updated UI, updated tuning variables
10/28/19	Added more detailed scoring
11/2/19	Updated images and assets through full GDD
11/18/19	Updated all art examples, mock ups, and descriptions

Project Overview

Release Target	11/25/19
Platforms	iPhone, Android Phone
Business Model	None (Free game with no revenue intended)

Game Engine	Unity using C#
Game Type	Single player
Genre	Solitaire game
Target Audience	Casual puzzle game players

Current Schedule

Please note that the schedule below is an informed estimate, but will likely change.

Milestone	Target Date
Production Starts	9/9/19
First Playable	9/23/19
Alpha	11/04/19
Beta	12/02/19
Final	12/11/19

Production is when the artists and engineers actively start building the game, based on the design documentation. This is an iterative and agile process so that design changes can be integrated throughout production, based on feedback and playtest results.

First Playable is the first time the essence of the game can be played. Major art, audio, and design features will be missing, and there may be many bugs. Nonetheless, it's an opportunity to playtest the game with players for the first time.

Alpha is when there are no major art, audio, or design features missing from the game, though there may be many bugs. Minor art, audio, or design changes can still be pending.

Beta is when there are no remaining art, audio, or design changes remaining, and only bug fixing remains, and perhaps a few low-risk polish items.

Final is when all bugs are resolved (fixed or deemed minor enough) and the game is ready for release.

Project Scope

Because this game will be designed and developed in a compressed timeline with a team that has never worked together before, we will choose a project scope that will allow us to actually publish a reasonably fun and reasonably bug-free game to the Apple and Android app stores within the Fall 2019 semester.

External References

TBD by Game Designer(s). Lots of solitaire games!

https://en.wikipedia.org/wiki/List_of_patience_games

Card Crawl: <https://www.youtube.com/watch?v=fHwDq6JHl24&feature=youtu.be&t=263>

FlipFlop Solitaire: <http://flipflopsolitaire.com>

Target Audience

The target audience for *Reactor* are casual puzzle game players. More details TBD.

Game Overview

Reactor is a single-player solitaire card game, playable on mobile devices. Players attempt to prevent their four suit-based "Reactors" from reaching or exceeding [insert number here] when cards start flowing into them. Players aim for the highest score they can get by matching cards of the same rank and colour while managing this ever-present threat.

Step-by-Step Game Outline

1. When the player begins the game, they start at the main menu
2. When they click the play button, they are brought to a screen with three difficulties (easy, medium, hard)
3. After choosing a difficulty, they are brought to the play screen
4. The play screen contains four foundations of 7 cards with the top card in each foundation revealed
 - a. Tokens drop down from the top of the screen and slam into place
5. There should be four empty reactors with their assigned suits (Clubs, spades, diamonds, hearts equivalents). These have numbers showing how full the reactor is (0/18)
6. A conveyor belt will cycle in three new tokens and display the number of tokens remaining in the deck
7. The player clicks an 8 of diamonds and stacks it onto the 9 of clubs
8. The player moves a 3 of clubs into the Clubs reactor
9. The reactor responds
10. The player matches a 5 of hearts with a 5 of diamonds
11. The tokens project a hologram that manifests into food which then launches itself towards space baby
12. The player requests more tokens from the conveyor belt, and 3 more pop out
13. If the player fills any reactors beyond their limit, they explode with food bits over the screen
14. If the player clears all the stacks, "the space baby gets super happy and has a party and you win" Alex McGrath

Terminology

Foundation = The cards laid face-down at the beginning of the game which slowly deplete over time and fill your reactors.

Reactor = The four aces are removed from the deck and symbolize your four reactors. The suits on each ace correspond to where the cards of each suit go.

Deck = Your side deck of cards not currently on the board, which are currently not in use.

Waste Pile = The pile of cards turned face-up from the deck, which can be used for matching and stacking.

Matched Pile = The zone which contains all matched pairs removed from the board.

Overload = A reactor reaching or exceeding [number] in total value; equivalent to “game over”.

Theme and Art Style

Will be entered by art team.

In this game, players are attempting to feed a Alien Space baby by combining various ingredient tokens to create meals. It takes place inside the kitchen/engine room of a ship, when the player then sends the meals from either into a tank or into space where the space baby is. The interior aesthetic is a sort of industrial sci fi, focusing on machines and structure with thinner lines. The alien elements such as the food or the baby being cute and weird, utilizing saturated colors and bold lines.

Most recent mock up for game screens:

<https://docs.google.com/presentation/d/1ER3ZZ6KVvCveBleduOTnWSYrFDnC7T-jPW82LuK4SAs/edit?usp=sharing>

Game Size and Dimensions

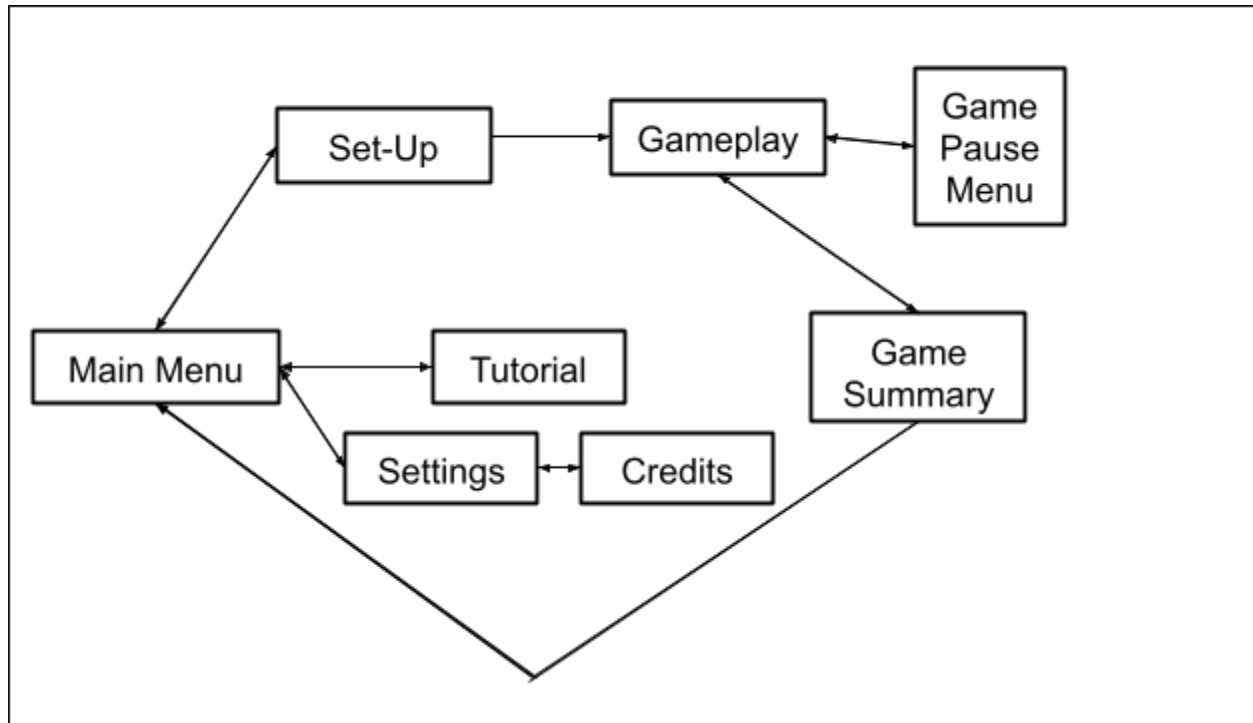
Since this game will be downloaded via the App Store (and possibly Android Marketplace), file size is a key factor. The final package must remain under the download-over-cellular limit, which most recently is 200 MB. When considering overall performance, fast game load times must be balanced with a small file size. Since the game is standalone, memory usage during gameplay isn't a major consideration, but we will strive to minimize memory usage to play nicely in a multitasking ecosystem.

Art assets will be created at the largest needed size, which is likely an iPad retina display. The assets will be scaled as needed for the smaller screens and different aspect ratios. All assets and implementations will be developed for cross-platform compatibility.

The game will be played in Portrait mode.

Screenshot: TBD

Game Flow



1. Loading screen / Splash screen

2. Main Menu

- a. Play Button (moves to set-up screen)
- b. Tutorial Button (moves to tutorial screen)
- c. Options/Settings (moves to options screen)

3. Set-Up

- a. Easy (moves to gameplay screen)
- b. Medium (moves to gameplay screen)
- c. Hard (moves to gameplay screen)
- d. Return (moves to main menu)

4. Gameplay

- a. Pause (moves to same Gameplay state, restarted gameplay state, options, or main menu)

5. Tutorial

- a. Return (moves to main menu)

6. Options

- a. Credits (moves to credits screen)
- b. Return (moves to main menu)

7. Game Summary

- a. Main Menu (moves to main menu)
- b. Restart (restarts gameplay)

Overall Asset list (Art/Audio)

https://docs.google.com/spreadsheets/d/1JIUINPbXkqWQVUCUYbMVmTmfC2eHJ8VgK5UI_EI1cWE/edit?usp=sharing

Analytics List

TBD, but likely:

- Score
- Playtime

Game Screens

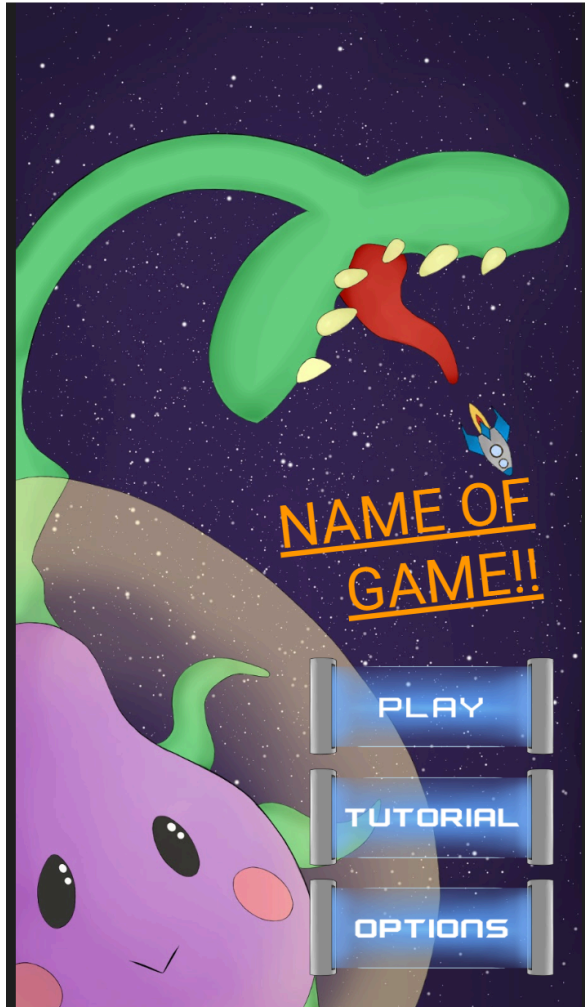
Main Menu

There are three buttons on the Main Menu screen. [Play], [Options]. Touching [Play] takes the player to the Set-Up screen (see below for details). Touching [Options] takes the player to the options menu (see below for details).

Art

Menu shows space baby, asymmetrically arranged on screen, chasing the ship the game takes place on. In the future, there will be food floating in the space behind the ship.

The buttons are located to the lower right, between the ship and the bottom of the screen. The wording and font of the title is currently placeholder.



General menus have this aesthetic, focusing on holograms (such as the one below):



Audio

Music

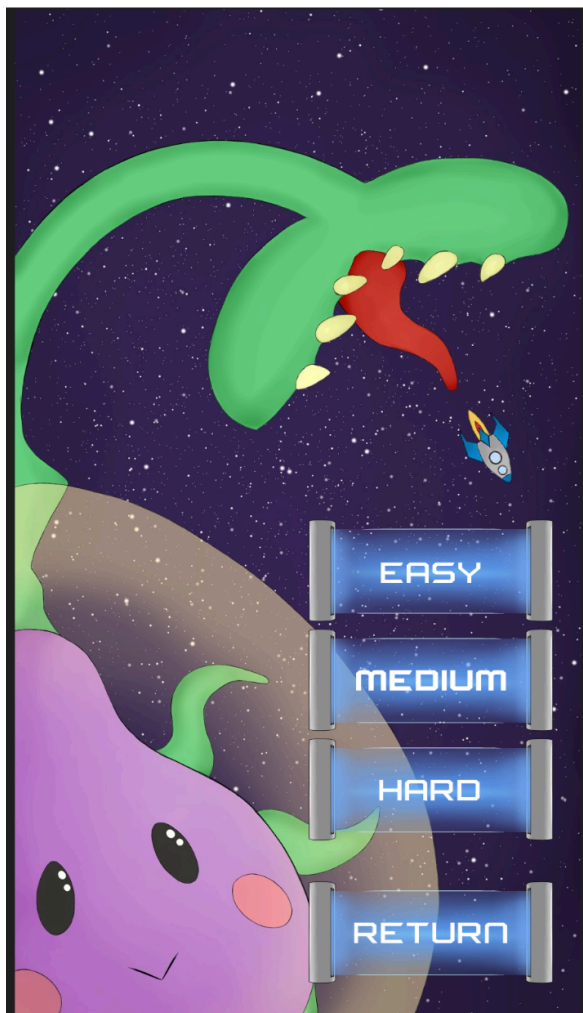
https://drive.google.com/open?id=1N4VQyKoA2UujOg7hp_cOLefAtpgGDmuH

Set-Up Screen

[Easy], [Medium], [Hard] play buttons should move the player to the gameplay state of the corresponding difficulty. [Return] button should return the player to the main menu screen.

Art

The current swt up screen uses the same background image as the main menu. This may change to a specific background for the set up, such as mechanical holder for the buttons, with the sarts behind them, dependant on if there is time.



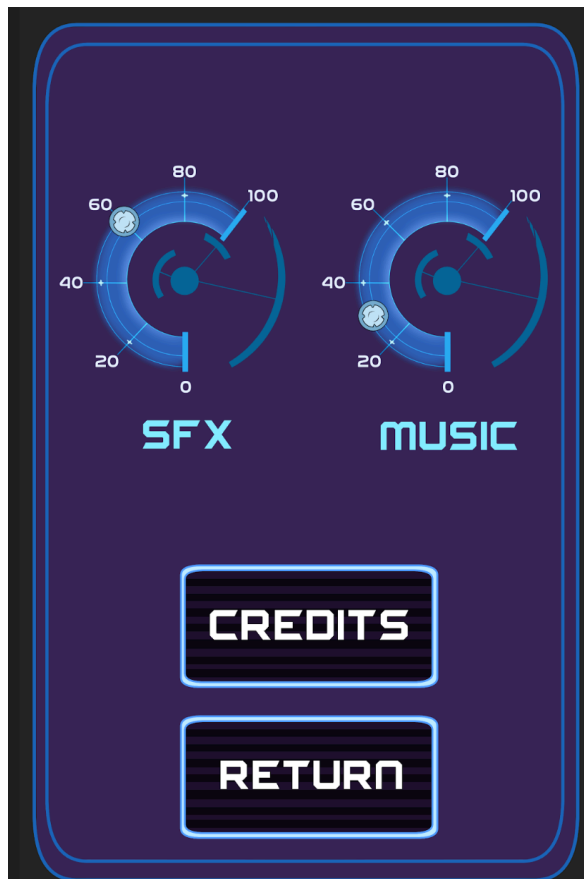
Audio

Options Menu

Options menu should have numbers or sliders to alter the volume for Sound Effects (buttons, matches, clicking noises, etc) and Music (background soundtracks). The options menu should have a [Credits] button that moves players to the Credit screen.

Art

Options menu appears holographic, and will eventually have a mock state of the game in the background. The sound controls are dials.



Audio

none

Credits

Should include participants in the course and link to a website.

Game Design

Alex McGrath - Lead Game Designer

Doug Levey - Game Designer

Sam Dormer - Game Designer

Sam Fioretti - Game Designer

Matt Regan - Game Designer

Art

Nat Mongilio - Lead Artist

Chhavi Kumar - 2D Animator and TA

Anna Christensen-Goodfellow - 3D Artist

Dharam Bir Khalsa - 3D Artist

Fang Chen - 3D Artist

Connor Ryan - 2D Artist

Peter Watko - 2D Artist

Wolfie Smith - 2D Artist and UI Design

Audio

Richie Olivas-Knapton - Lead Audio Designer

Granger Smith-Massa - Audio Designer

Zofia Shura - Audio Designer

Programing

Shan Jiang - Lead Programmer

Ian Macpherson - Programmer

Jona Lawrence - Programmer

Max Mark - Programmer

Noah Brinton - Programmer

Per Van Dyke - Programmer

Production

Armon Walker - Lead Producer

Luke Townsend - Game Design Producer

Alexandra Webb - Audio Design Producer

Lucas Kohn - Art Producer

Corbin Nelson - Programming Producer

Executive Producers

Bassam Kurdali - Executive Producer of Art

Ira Fay - Executive Producer of Game Design

Art

The art team would prefer the first credits image, though the second is acceptable if the sizing of the borders are too difficult to fit into unity.

The border and background have been uploaded as separate assets, along with the font, for the text to be up in on the unity side.

Credits

Executive Producers

Art - Bassam Kurdali
Game Design - Ira Fay

Producers

Lead - Armon Walker
Game Design - Luke
Townsend
Art - Lucas Kohn
Audio - Alli Webb
Programming - Corbin
Nelson

Game Design

Lead - Alex McGrath
Game Designers
Doug Levey
Sam Dormer
Sam Fioretti
Matt Regan

Art

Lead - Nat Mongilio
2D Animator & TA
Chhavi Kumar
3D Artists

Credits

Executive Producers

Art - Bassam Kurdali
Game Design - Ira Fay

Producers

Lead - Armon Walker
Game Design - Luke
Townsend
Art - Lucas Kohn
Audio - Alli Webb
Programming - Corbin
Nelson

Game Design

Lead - Alex McGrath
Game Designers
Doug Levey
Sam Dormer
Sam Fioretti
Matt Regan

Art

Lead - Nat Mongilio
2D Animator & TA
Chhavi Kumar
3D Artists
Anna Christensen-Goodfellow

Scoring

Base Scoring:

- Each match made rewards the player with 100 points.
- At the end of the game (when all the stacks are empty), the player receives 600 points for each reactor with no tokens inside of it.
- This means the maximum score is 5000.

Bonuses (stretch, low priority)

- At the end of the game, the player should get 1000 points if they finish under one minute, 800 points if they finished under two, 600 points if they finish under three, 400 if they finish under four, 300 points if they finish under five, 200 points if they finish under six, 100 points if they finish in over six minutes.
- After all the points are added up, the player multiplies their points for difficulty. Easy difficulty multiplies it by 1 (basically no change), Medium difficulty multiplies the score by 2, and hard multiplies the difficulty by 3.

Multiplier (stretch goal, low priority):

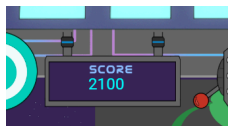
- Whenever the player makes a match, multiply the points they earn by the number of matches they've made without taking any other actions (stacking cards, revealing cards, moving things into the reactors, etc.)

Art

Pause Menu Score



Main Game Score



Audio

<https://drive.google.com/open?id=1RhvkIIYGHcaboljYbW3pCEJu7YYk6sGI>

Game Summary

Details TBD by Design

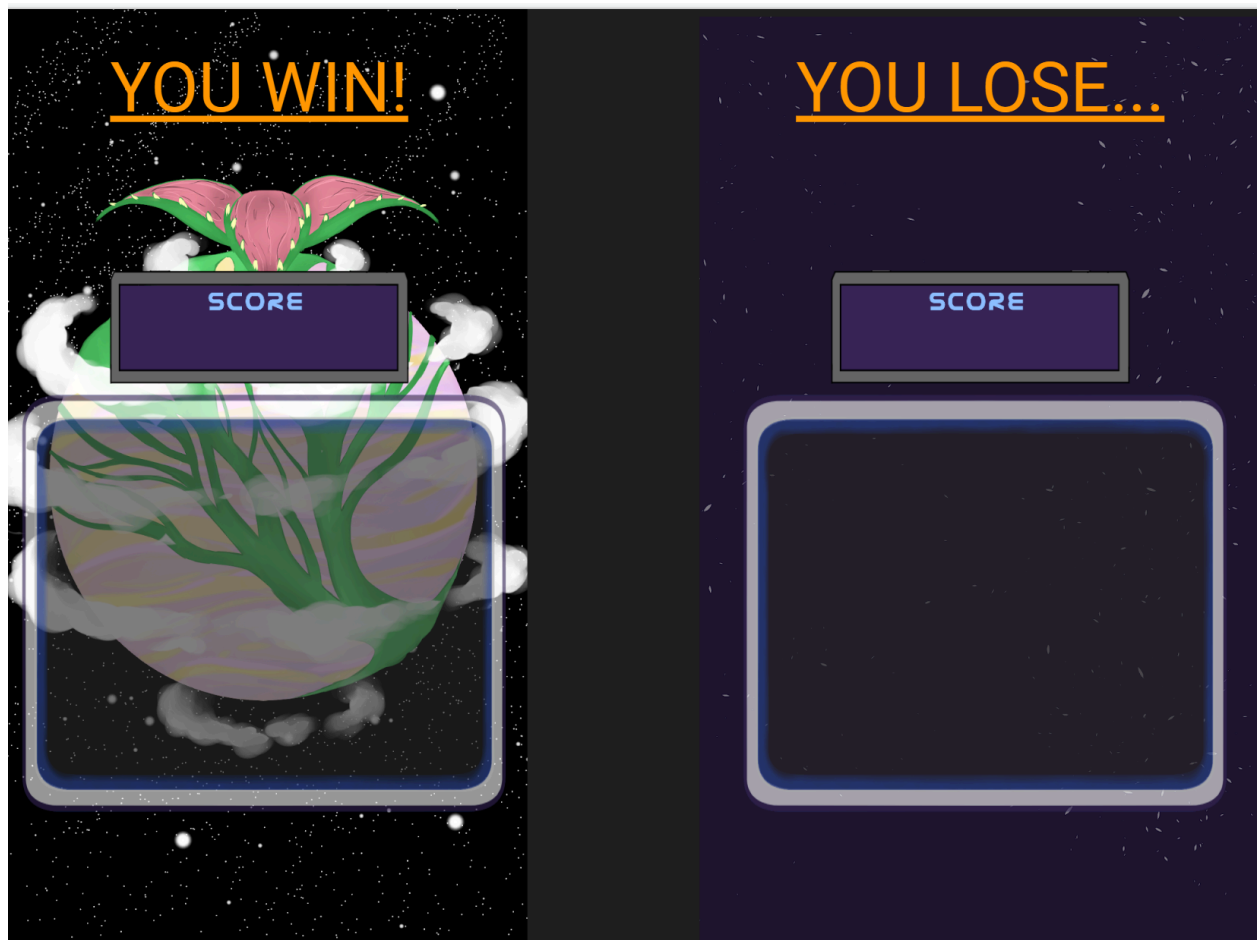
Art

There will be a win screen and a lose screen. In terms of format, they have the same format (specific formatting and font TBD), but each will have a different background.

As a stretch goal, Art would like a short animation for each.

For winning (as defined by game design), We would like the baby to show the happy animation, gradient fade to white from the baby, then shows the planet.

For losing (as defined by game design), The reactor begins to spark (2D art assets) and shows error messages. A loud clunk and sputtering is heard, maybe some explosion sounds, and the reactor tilts/collapses, showing it's clearly broken. The lights flicker, and go out. Then the space baby, slowly starts floating closer in the background. Snap to black, and there's the lose screen stats.



Audio

Lose: Reactor falls sound

Win: trumpet happy sound

UI

Touch Interactions

Players can move cards and make matches via touch-drag or single touch.

Touch-Drag

Touch-drag is fairly straightforward. If a player tries to drag an immovable card (e.g. anything not on top of a reactor, anything not on the top of the waste pile), wiggle the card that can be moved, give a little haptic feedback, and play a sound effect. See <http://flipflopsolitaire.com> for a great example of this.

If a player tries to drag a card to someplace it can't go, the card (and anything being moved with it) should snap back to its original position.

Implement tuning variables that will allow the game to detect if the player is attempting a touch-drag, or a single touch. The two relevant details are:

- How long has the player been holding down the touch?
 - `touchDrag_HoldThreshold`
- How far has the finger moved on the screen since the start of the touch?
 - `touchDrag_MoveThreshold`

If either threshold is exceeded, then the interaction is a touch-drag. While both thresholds haven't been exceeded yet, the interaction should be treated as a single touch.

Touch-Touch

Whenever the player clicks on a usable token, it should have some haptic feedback such as enlarging to show that is selected. If the player then clicks a token of the same color and value, the tokens should match. If the player touches a token with one higher value, the first selected token should move to stack on that one. If the player touches a reactor of the same suit, it should move into that reactor. If the player clicks a token or reactor that isn't valid, the originally selected token should wiggle and deselect.

Reactor

Touch a card on top of a reactor that can be matched: match it!

Touch a card on top of a reactor that can't be matched: wiggle that card, haptic, SFX

Touch a card not on top of a reactor: wiggle top card of reactor, haptic, SFX

Art

3D asset, exported in 2D images.

Reactors are a mix between chemical reactors and food storage areas like fridges.

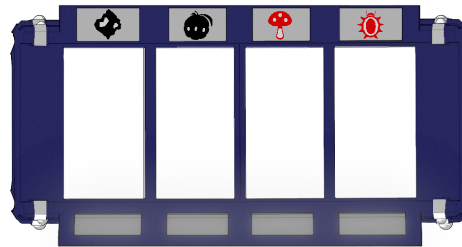
Tokens (cards) fit into reactors and stack to show what is on top.

Reactors display a number on a screen showing the current reactor limit and amount filled.

Back



Front



Example from mockup.



Example from game:



Audio

Tokens stacked into the reactor clang

https://drive.google.com/open?id=1mNvmL2HFvAxib7YPNG_FQeqZno7Eqvsb

Foundations (Stacks)

Touch a single card:

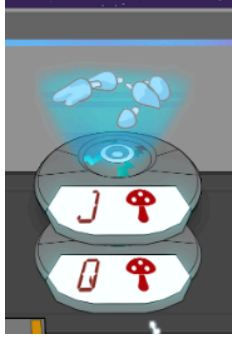
- If it can be matched, match it.
- Else if it can be moved to another stack, move to leftmost valid stack
- Else if it can be moved to an empty stack, move to leftmost empty stack
- Else if it can be put into reactor without exploding, move to reactor
- Else: wiggle that card, haptic, SFX, and highlight the reactor that would blow up

Touch a card with more cards under it:

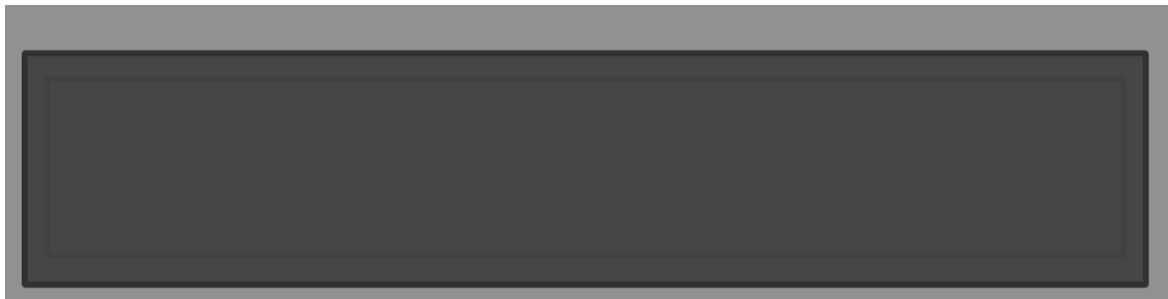
- If it can be moved to another stack, move to leftmost valid stack
- Else if it can be moved to an empty stack, move to leftmost empty stack
- Else: wiggle that card and all cards that would move with it, haptic, SFX (see <http://flipflopsolitaire.com> for an example if needed)

Art

Stacked tokens, with only the top token showing the hologram



The foundations themselves will have a base, which has not yet been created by art, but will be placed along this space of the background.



Audio

Token_stack.mp3

<https://drive.google.com/open?id=1D5F-XMbYqWToiHhbJYMwBqS7LKanGkry>

Waste Pile

Touch a card not on top of waste pile: wiggle top card of waste pile, haptic, SFX

Touch card on top of waste pile:

- If it can be matched, match it.
- Else if it can be moved to a non-empty stack, move to leftmost valid stack
- Else if it can be moved to an empty stack, move to leftmost empty stack
- Else if it can be put into reactor without exploding, move to reactor
- Else: wiggle that card, haptic, SFX, and highlight the reactor that would blow up

Art

Tokens be displayed on conveyor belt.



The conveyor itself is explained in further detail in the Deck section below.

Audio

https://drive.google.com/open?id=1WrK9wptbfowN-_diEr8Uu9nzKJV42n_3
[Match_sound](#)

Buttons

Button interactions/feedback? How will buttons react to being touched?

Art

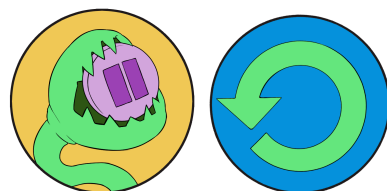
Below: main menu buttons



Below: ingame pause menu buttons



Below: Ingame pause and undo buttons



Audio

<https://drive.google.com/file/d/1YA9-OLQuSk54p2YLeAnp74xM9GsQ7Z8-/view?usp=sharing>

Gameplay

Thin Screen Mockup



Wide Screen Mock up.



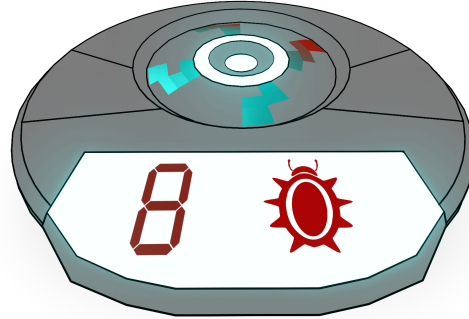
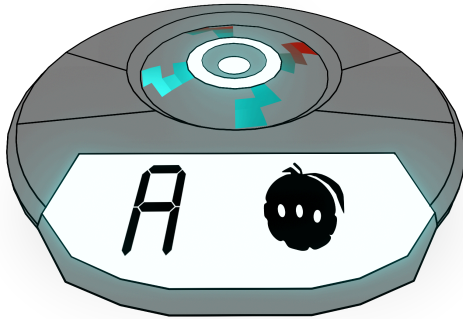
Gameplay Elements

Cards

Contain a Number and a suit, further detail by Game Design

Art

Cards are holographic chips (tokens), with pictures/holograms of ingredients that overlay with their match to create a meal. Their back is a uniform grey, with their front colored red or black dependant on suit.



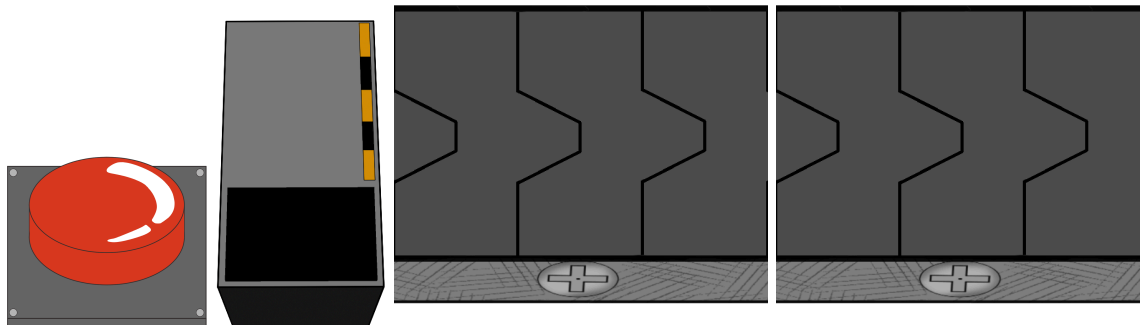
Zones

Deck

This is the pile where cards remain face down. Face down cards here cannot be interacted with except by using the Flip action.

Art

This is represented by a conveyor belt with the number of tokens left displayed on it. It has a button that can be tapped to deal 3 more tokens. The conveyor belt is broken into squares so it can be extended as long as needed to fit on the phone.



Waste Pile

This is a pile of revealed cards. A player may only interact with the top card. Optimally, they should be able to easily see the top three stacked in a way that demonstrates only the 'top' card is easily interacted with and perhaps click the pile to be able to temporarily view the entire pile.

Art

Conveyor Belt: Cards are drawn along a conveyor belt, with one space open to take from.

- 6 drawn cards can show at a time.
 - A glass cover stops players from grabbing any but the "top" card.

- By tapping the food chute to the right, 3 new cards roll out.
 - The chute shows the number of cards remaining not yet drawn. It could also show this number over the number of cards in the deck.
 - When it runs out, it shows a different symbol, and tapping it again will return all cards back to the chute.
- By swiping a finger across the conveyer belt, players can view old cards.
 - While they do so, the glass slides to cover all the cards, so they can't select old cards.

Audio

Conveyor belt sounds.

<https://drive.google.com/open?id=1UIU1pComwrLcktDgHGtTPRqjNuBnGvJT>

Matched Pile

This is a pile of revealed cards that have been paired together. These cards cannot be interacted with for gameplay, but should be viewable upon clicking the pile. Only the top card needs to be easily viewable.

Art

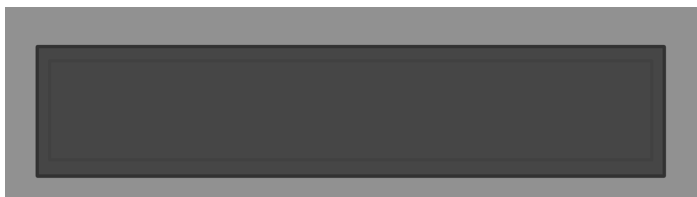
Game Design has been decided that matched cards will not be shown.

Foundations

The top cards of these piles should always be revealed. If the top card of a foundation is removed, after the removed card is placed somewhere else, the next card in the foundation should become revealed. The top revealed cards of the foundations are the cards often interacted with. If all four stacks are ever empty, the game ends.

Art

- There are 4 foundations, aligned to the bottom and stacking up. They will each have a base that shows where the foundation is and 'holds' the cards, as of yet not finished by art. The base will go on this space of the background.



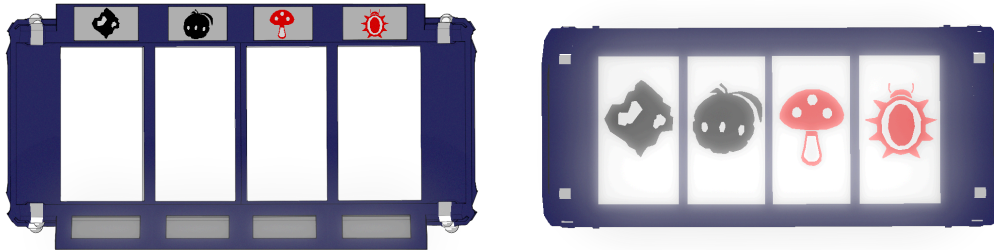
Reactors

Name pending based off flavor. Each Reactor has an affiliated suit. Cards not of that suit should not end up in that Reactor. Cards in this stack are always face up. The top card

cannot be Moved, only Paired. If the total value of the cards in a single Reactor total above [**Reactor Limit** (18?)], the game ends. (Face cards are all worth 10. Aces are worth 1.)

Art

Below: reactor front (left) and back (right)



Reactors glow blue when a token is picked up and can fit in the space and red when they are almost full.

They also have numbers in the space below where the tokens go, listing how full they are.

Set-Up

A standard deck of 52 playing cards (4 suits, ace - king) begins in the Face-Down Deck area in a shuffled, face-down pile. Each of the four stacks are filled with [**Stack Start Size** (7)] cards from the top of the Face-Down Deck. The top card of each stack becomes revealed to the player.

Art

There is not currently an animation planned for set up.

Audio

Set up sounds?

The Move Counter

Adjacent to the reactions, there should be a counter that counts down from a number determined by the difficulty.

- Easy: 21
- Medium: 18
- Hard: 15

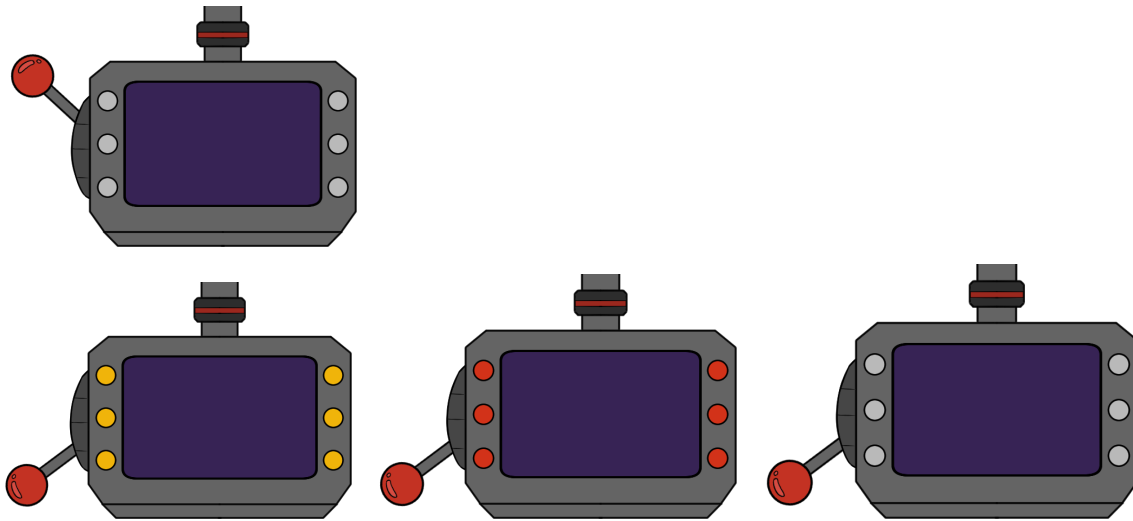
After the player takes any Action (see details below), the counter should be reduced by one. Once it reaches zero, the following happens. From left to right, the top card of

each Foundation will be moved into the Reactor of the matching suit. Any Foundation without a revealed card on top will reveal the top card. The counter should be reset.

In addition, the counter that counts down should be a button. If clicked, it should shift the top cards of foundations up into the reactor and reset the counter as seen above.

Art

The move timer is a screen with a lever at the right. When pressed, the lever animates down. The box should have “moves” written across the top, in the nasa font. The buttons glow specific colors when activated (yellow) or in danger (flashing red)



Actions

Once gameplay has begun, the player may take the following actions, assuming there isn't something else currently happening, like a Deck being shuffled.

Move A Card

The player may drag any revealed card not in the Reactor Zones. A Card may be moved around the stacks if;

1. If Card A (the one being moved) is one numerical value lower than Card B (a card on top of a stack). This includes, 10 to Jacks, Jacks to Queens, Queens to Kings. Aces are low. Color doesn't matter.
2. If Card A is being moved onto an empty Stack.

In both of these circumstances, Card A might not be the top of the Stack. If it isn't, the cards above it must be of a lower numerical value, making a chain.

Additionally, the top revealed card of any Stack or Face-Up pile may be moved into the Reactor. It must match the Reactor's suit, but its number doesn't need to create a chain. This can cause game overs.

Art

Copy of token remains in space in a semi-transparent image, while a larger version of the token with full opacity is dragged by the player.

The places where the token can be put glow with a highlight. The glow is yellow if the token can be placed there, and green if it can match. The glow lightens when the token is hovering over the area.



Audio

https://drive.google.com/open?id=1AZbFTOoejJHDC7AkP7JKoIVSjL_YMzLm

Pair A Card (Matching)

The player may drag any revealed card that is on the top of it's pile and move it onto a card that matches it in Number (or face, King to King, Queen to Queen) and Color. Both these cards are placed in the Matched Pile immediately.

Art

Holograms of each food overlay, and together form the full meal. They then flash, and gain full color, then fly off screen for the space baby to eat.



Audio

<https://drive.google.com/open?id=1-AuqmLzP9a1A9UHWYVshGEAWVLXL-fau>

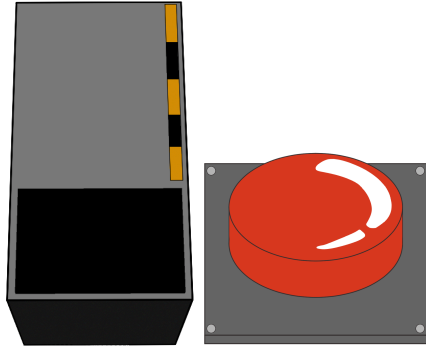
Flip

There should be a button the player can press that will do one of two things. If there are cards remaining in the Deck, the button will add the top three cards from the deck into the Waste Pile, face-up. If there are less than three cards in the Waste Pile, it will place as many cards as there are.

If the player uses the Flip button while there are no cards in the Deck, the Waste Pile is instead flipped back into the Deck face down and the top three cards of the now flipped Deck should be placed into the Waste Pile face up. This counts as a single action. Once the player has flipped the deck like this once, they should be allowed to hold down the Flip button to see a list of all the incoming cards.

Art

This is a button on the conveyor belt. When clicked, it animates as being pressed, and three new tokens leave the conveyor belt's chute. Once the entire deck is moved through, the conveyor belt brings all the tokens back into it, and then deals out 3 new ones.



For the animation, the token should slide out from the left, then 'drop' by getting smaller onto the conveyor belt, which will be rolling away.

Audio

Conveyer belt

<https://drive.google.com/open?id=1UIU1pComwrLcktDgHGiTPRqjNuBnGvJT>

Pause Menu

Options TBD by game design.

Art



- The menu can be accessed in game by tapping a button with the space baby. This will list the space baby, along with the 4 buttons, and the score.

Audio

See above

Tuning Variables

There are many elements of the game that we will need to tune later for fun and game balance. (For "tuning" think about tuning a radio knob. We want to get it just right. Oh wait, that analogy probably isn't helpful to you. The internet can tell you what a radio is.)

In general, no value in the game should be hard-coded. Instead, a tuning variable should be used, and all tuning variables should be kept in a separate file (presumably .json) that a game designer can easily modify without bothering a programmer for help.

The Game Designers should create and maintain this list. Entries in the list should include a description of the variable, the name of the variable, and the game designer's initial guess at the values. This can include things like score-related variables, the text of

randomized tips that will appear, or the amount of time it takes a card to be dealt (in milliseconds).

ExampleVariable: Value (Min Val, Max Val): An explanation on how we've organized this. The values in parentheses are estimated expected values to help demonstrate the thought behind the idea.

StartingFoundationSize: 7 (3,12): The amount of cards in each Foundation at the start of the game. All four Foundations should be equal. The top card of the Foundation should be face up.

EasyReactorLimit: 21 (11, 50): If the total value of the cards in a single Reactor equal or surpass this value, the reactor 'explodes' and the game ends. This variable will change often in playtest.

MediumReactorLimit: 18 (11, 50): If the total value of the cards in a single Reactor equal or surpass this value, the reactor 'explodes' and the game ends. This variable will change often in playtest.

HardReactorLimit: 15 (11, 50): If the total value of the cards in a single Reactor equal or surpass this value, the reactor 'explodes' and the game ends. This variable will change often in playtest.

FlipAmount: 3 (1,7): The amount of cards revealed and placed into the Waste Pile whenever the Flip button is clicked.

EasyActionCounter: 21 (11,50): Whenever the player takes an action, the action counter moves down, starting from the action counter amount. If it reaches zero, see above (move counter).

MediumActionCounter: 18 (11,50): Whenever the player takes an action, the action counter moves down, starting from the action counter amount. If it reaches zero, see above (move counter).

HardActionCounter: 15 (11,50): Whenever the player takes an action, the action counter moves down, starting from the action counter amount. If it reaches zero, see above (move counter).

Tutorial

Tutorial details TBD by game design.

Art

Screens will be made and arranged by Game Design.

Audio

Further Work

Cheats to Facilitate QA Testing

We will want to build some cheats into the game so that when we're playtesting the game, it's easier to test the specific parts we want. These cheats will be listed here, TBD by a Game Designer. Obviously all these cheats should be disabled and removed in the final version of the game.

Open Design Questions

- Is the game generally fun and headed in the right direction?
- How should scoring work?

Future Work

If time allows, we'll implement more features, such as:

1. Add social features to share info about the game on Facebook, Twitter, etc.
2. Add a prompt to rate the game
3. Add Telemetry and Analytics (gather data on player behavior)
4. Plan for (and implement) an app update
5. Add Power-ups
6. Add a progression system (achievements, badges, unlocks, etc.)

Potential Ideas

- TBD