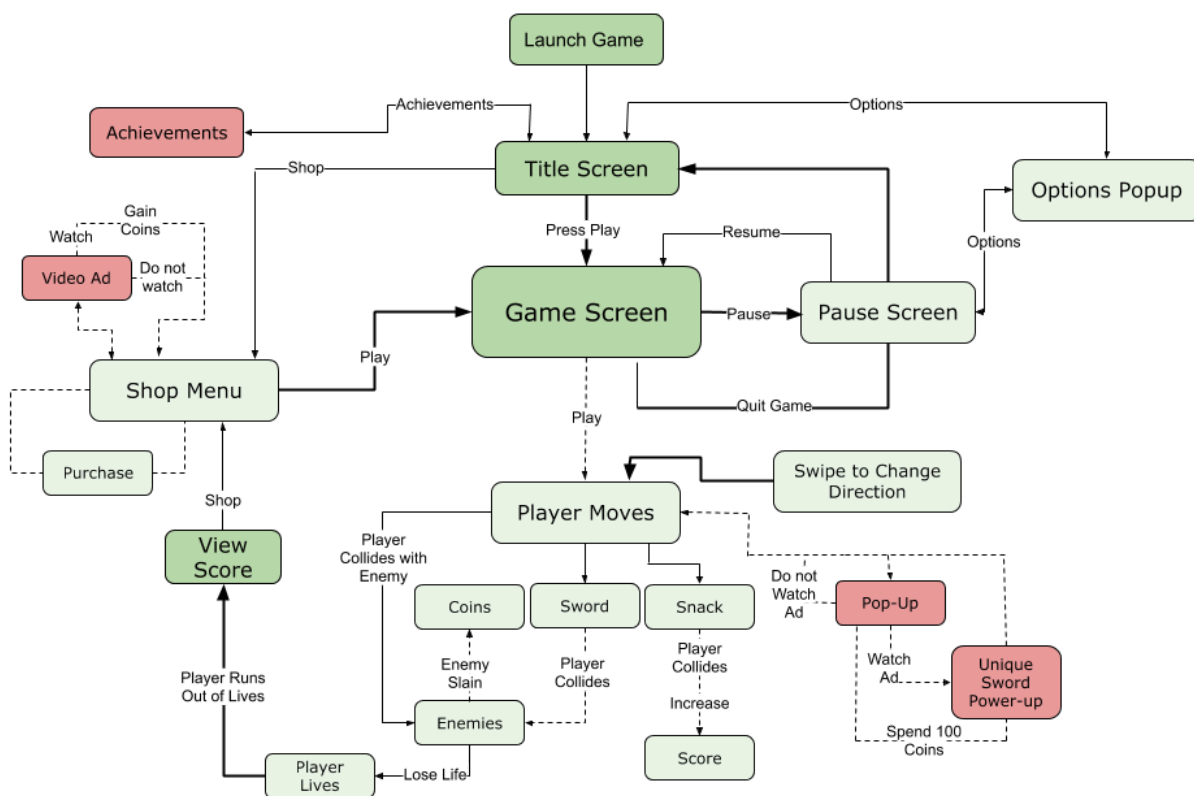


Save The Snacks – Technical Design Document

SYSTEM OVERVIEW

- **Systems, Features, and Requirements:** Save The Snacks uses swipe controls to change direction and enemies that can enter a vulnerable state, similar to the game Pacman. The player will attempt to survive as long as possible in the maze by avoiding enemies, and defeat them by collecting swords that can be tapped to make enemies vulnerable for a certain amount of time. There will be shop upgrades to increase sword duration and the player's total number of lives.
- **System Architecture:** Red nodes will not be implemented into the game.



DEVELOPER TOOLS

- **Game Engine:** Phaser 2.7.6 Framework for JS
- **Art & Animation Tools**
 - Tiled Map Editor (1.2.4)
 - TexturePacker (5.0.1)
- **Sound Tools:**
 - Audacity (2.3.2)

SOURCE CONTROL

All source code will be backed up through a repo on my personal GitHub account. I will use GitHub Desktop (2.0.4) in tandem with Visual Studio Code (1.35.1) to organize files and push to the repo. There will be a “development” branch that will be used to safely commit code without overriding the master branch, which will contain a stable build.

Link to the GitHub repository: <https://github.com/djblooky/save-the-snacks>

CODE & LANGUAGE OVERVIEW

- **Main Game Loop** - Per Phaser's structure, calls to additional functions will be made in the Preload, Create, and Update functions depending on when the action must be performed. Loading in image and audio files can take place in Preload, rendering images/audio as well as setting any physics, graphical, or input specification will happen in Create, and things that require constant checking like player movement/collision will happen on Update.
- **Game states:** Phaser 2 allows organization of code into different files that contain “game states.” I will be using the following states in their own JS files to contain code for different screens/loading points in the game:
 - Boot
 - Load
 - Menu
 - Level
 - Pause
 - Results
 - Shop
 - I will also use a separate JS file - game.js - to contain global variables and various data that I will need to use across multiple game states.
 - I will add each game state in “main.js” and call to start the program from here
- **Programming Languages & Standards** – JavaScript ES6 to work with Phaser 2.7.6
- **File Formats:**
 - .pngs for packed sprites/static images
 - .json for maps created in Tiled
 - .mp3 for audio files
- **Commenting** – Comments will serve to provide a clear understanding of different variables/functions' purposes and clarify the workings of particular areas of code.
- **Naming Conventions** – ESLint config standard style guide: <https://standardjs.com/>. This is my first game written in JS so I will refer to this often.

TECH GUIDELINES, GOALS, & RISKS

- **Technical goals:** Assets will be stored in the “assets” folder in the main game folder with index.html and game.js. Additional JS files will be stored in the “src” folder.

- **Risks:** Things that I acknowledge may pose obstacles going into the project
 - Tight timeline - This project may be constrained by the limitations of my internship which has an end date of August 9.
 - Lack of JS/Phaser experience - I will need to rigorously study the fundamentals of Phaser's Framework in order to be able to effectively and efficiently create my prototype/vertical slice.
 - Solo project - I am completing this project on my own, so certain aspects of it could take longer given multiple tasks cannot be completed in tandem

NOTEWORTHY GAMEPLAY SYSTEMS

- **Swiping**
 - Player can swipe in a direction to move that way so long as there are no walls blocking them
- **Collision**
 - Player can collide with snacks to "collect" them to add to their score
 - Player can collide with enemies to lose one life
 - Player can collide with swords to pick them up
 - Once a sword is activated, the player can collide with enemies to attack them.
- **Enemies Accumulate**
 - Two enemies will spawn on game start, but every x seconds (spawnRate) another enemy will enter the game. The spawnRate variable will decrease by y seconds every update, making it so that enemies spawn more frequently the longer the player lasts. The spawnRate could be capped at a minimum amount.
- **Leveling**
 - TBD - should there be multiple stages or should it be single stage/enemies come in waves?
- **Tapping**
 - Tapping can be used to select a sword from the player's hotbar if it has been collected.
- **Upgrades**
 - In the shop, there will be a tab to upgrade the player's sword duration
 - Another shop tab will allow the player to purchase additional lives
- **Timed Score Multiplier**
 - Snacks collected will increase from their base score value as time goes on
 - Enemies defeated will award more coins increasing from their base value as time goes on

UI ELEMENTS

Loading Screen

This menu will display 2D art and a loading bar that increases to display loading progress.

Title Screen Menu

- Play
- Settings
- Shop

Game Screen

- Pause button

Pause Screen

- Will include the build number
- Resume Game
- Settings
 - Music toggle
 - SFX toggle

Results Screen

- Shop Button

Shop Screen

- Purchase buttons
- Play button

CN REQUIREMENTS

- SDK Integration - This will not be necessary for this project
- Dynamic Content Management -
- Specs (Download and Install size) (Resolution) – need to consult producers on this.
My personal goal/assumption is that I will not need to deploy this to every possible device, but perhaps a small subset of devices.

MISCELLANEOUS

- Audio Systems
 - .wav / .OGG
 - Phaser built-in sound library
- Scripting System
 - JavaScript ES6
 - HTML5
- Analytics (SDK's) - n/a
- Ad Flows - n/a

TASK LIST

- Program core gameplay
 - Player movement
 - Enemy AI

- Life/score/coins systems
- Lose conditions
- Combat mechanics
- Working game settings and pause menu
- Complete all game assets
 - UI elements for all menu screens
 - Button UI (play, pause, shop, purchase, back, close, setting, sfx/music)
 - Lives icons
 - Sword icon
 - Character sprites
 - Enemies (2-3)
 - Playable characters (4)
 - Shop upgrade art
 - Sword upgrade art (3)
 - Character unlock art (3)
 - Collectables
 - Snacks (3-5)
 - Sword (1)
 - Stage map
 - Tiles (2)
 - Transition screen art (3)
 - Loading screen
 - Menu screen
 - Shop backdrop
 - Title treatment
- Retrieve audio assets
 - Music (title screen, in-game, end game)
 - SFX

CHANGE LOG

June 13, 2019 - Document template created by Haley Phillips

June 24, 2019 - Initial document completed

June 25, 2019 - Updated source control and UI elements

July 8, 2019 - Revised tasks list, risks, and gameplay systems

CONTACT

Cartoon Network Internal Development

Haley Phillips - haley.phillips@turner.com