Avalanche Adventures

Technical Design Document

David Laws

MGMS Degree Program

Full Sail University

September 14, 2014

# Table of Contents

Table of Contents 2

Introduction 3

Project Concept / Pitch 3

Target Platforms / OS Version(s) Supported 3

Technical Design Goal(s) 3

Development Engines / APIs / Programming Languages 3

Development Standards / Cold-Start Procedures 3

Project Functionality 4

User Modes 4

Main Mechanics / Sub Mechanics 4

AI Module(s) 4

User Interface / Inputs / Sensors 4

Graphics / Animation / Video 4

Audio 4

Networking 5

Databases / Server Side Technology 5

Advertising / In App Purchasing 5

Security 5

Localization Plan 5

Software Updates 5

Custom Topics 5

Additional Technical Information 5

Mockups / Flowcharts 6

Assets 7

Visual Assets 7

Audio Assets 7

Features 8

Crucial, Secondary, and Wish List Features (A, B, and C Features) 8

A - Crucial Features 8

B - Secondary Features 8

C - Wish List Features 8

Development Roadmap 9

Storyboarding and Game Design 9

Mobile Development 1 9

Mobile Development 2 9

Testing 10

References 11

# Introduction

### Project Concept / Pitch

Avalanche Adventures, a fast paced action game where you must use quick reflexes and special abilities to survive to storm of ice and save the festival.

### Target Platforms / OS Version(s) Supported

Avalanche Adventures will be developed for the Android platform, Android 4.1 or higher. It should also be playable both on phone and tablet settings, which means adding in graphics support for both devices.

### Technical Design Goal(s)

The first goal is to complete this as an independent project from start to finish. It is also introducing a RPG-like leveling system into the arcade style game. Previous games with a similar mechanic (avoiding falling objects) have not really attempted to expand much beyond dodging. The use of special abilities and character progression are the novel mechanics I would like to add to the formula. Developmentally speaking, in addition to the character progression the social aspects (leaderboard and social media) and animation will be a challenge.

### Development Engines / APIs / Programming Languages

Avalanche Adventures will be developed using the Unity platform, making use of the 2D features and sprite animation. The built in engine should contain sufficient features to complete the capstone requirements. Scripting will be done in C#, which is fully supported by Unity.

### Development Standards / Cold-Start Procedures

Repos for the project can be found at https://edo.fullsail.edu/underdog/git/david-laws/git and https://github.com/Zalfier/AvAdventures. We are using MonoDevelop - Unity version 4.0.1 as our IDE. The best way to get started would be to download Unity along with the demo or latest version of Avalanche Adventures. New characters and stages can be created by extending the character and stage base class respectively.

# Project Functionality

### User Modes

The game is entirely single player based. There are two game modes: Arcade and Adventure. Arcade is an endless survival mode where you compete for a high score. It represents the standard gameplay of surviving through dodging and breaking falling icicles. Adventure follows a story and contains numerous levels. Each level will attempt to tweak the standard gameplay model by adding in different objectives and mechanics.

### Main Mechanics / Sub Mechanics

The primary user mechanic is the special character abilities, which are on a cooldown timer and can be activated through tapping the screen. These abilities will have a predetermined effect and animation that originates from a single point, ie the player character. They will vary from character to character but revolve around destroying icicles and/or shielding the player. As part of the character leveling system the abilities will grow over time. The other user mechanic is movement. This will primarily be left to right movement along the bottom of the stage, but certain levels in adventure mode may seek to change this formula by adding up and down movement as a factor as well as scrolling levels. The actual falling icicles will also vary and will seek to challenge the way the player moves to dodge.

Finally, environmental effects will be features that temporarily alter the player’s movement ability, such as by slowing or increasing move speed in different ways. These will mainly be Adventure mode features as well. In addition I would like to include tools, one time use items which can be activated by tapping an on screen button, which will be able to dispel environmental effects, among other possible uses. They will be introduced in Adventure as needed. I would eventually like to include most or all of the unique Adventure Mode mechanics as options for altering the gameplay of Arcade Mode. These various effects can be added to the game by including movement multipliers, which can be adjusted to change movement speed.

### AI Module(s)

There are no plans for an extensive AI at this time. Some minor AI may be included in a boss fight during Adventure Mode, but this will be limited to simple movement and attempting to drop objects when the enemy passes over the player.

### User Interface / Inputs / Sensors

The user interface will be entirely touch based. Movement will be controlled by buttons located in the bottom corners of the screen. Advanced movements will add more buttons to the overlay as needed. If tools are included buttons will appear along the side(s). Settings will be included to adjust single sided buttons for handedness. Tapping anywhere that is not a button will be used to activate the abilities. Menus are all button activated. No additional sensors or peripherals are needed.

### Graphics / Animation / Video

Avalanche Adventures will feature sprite-based graphics. Rendering and animation are taken care of by Unity’s built in sprite features. Assets are custom, hand drawn images preloaded to the system as PNGs. Animations are done based on individual needs of the assets.

### Audio

What audio format or technology are you using? Not all versions of OSes on all devices support all formats. For instance, some devices with older Android versions do not support certain modern sound formats. Have you done any testing in this regard? Will you have background music and sfx? Are you doing anything interesting with the audio beyond just playing premade files? Do you have any voice acting in your project? If so, where are those assets coming from?

Audio will include sfx and background music, all premade. Different music will be used for menus and during gameplay. All audio is custom made and uploaded to Unity as WAV files. Unity’s built in Audio Files assets takes care of encoding and processing the files.

### Networking

This project does not use any networking technology. All game modes and functions are single player.

### Databases / Server Side Technology

The game has no need for a database (local or server) at this time. One may be needed later on if public leaderboards are created.

### Advertising / In App Purchasing

Some research has been done into advertising. Banner ads would likely be featured during menus as well as possible interstitial ads between games, possibly with time delays to prevent them for overwhelming players. I would like to eventually create a store for in-app purchasing, though no research has been done into the implementation of such. IAP would likely include the purchasing of in-game currency, as well as exp boosts for characters. In addition I would like to include ad removal as an incentive to making a small purchase in the store.

### Security

There will be no personal data collection and as such no research has been done into security.

### Localization Plan

At this time there are no plans for localization. If localization eventually occurs it will only need to translate menu labels. There is limited to no text during gameplay and no voice acting.

### Software Updates

Updates would be free to download, and would include any necessary changes to keep the game functioning on newer operating systems. In addition I would add extra content such as new adventure mode maps/levels/etc and new characters to unlock. Updates would be free, although most content would still require in-game currency to be unlocked.

### Custom Topics

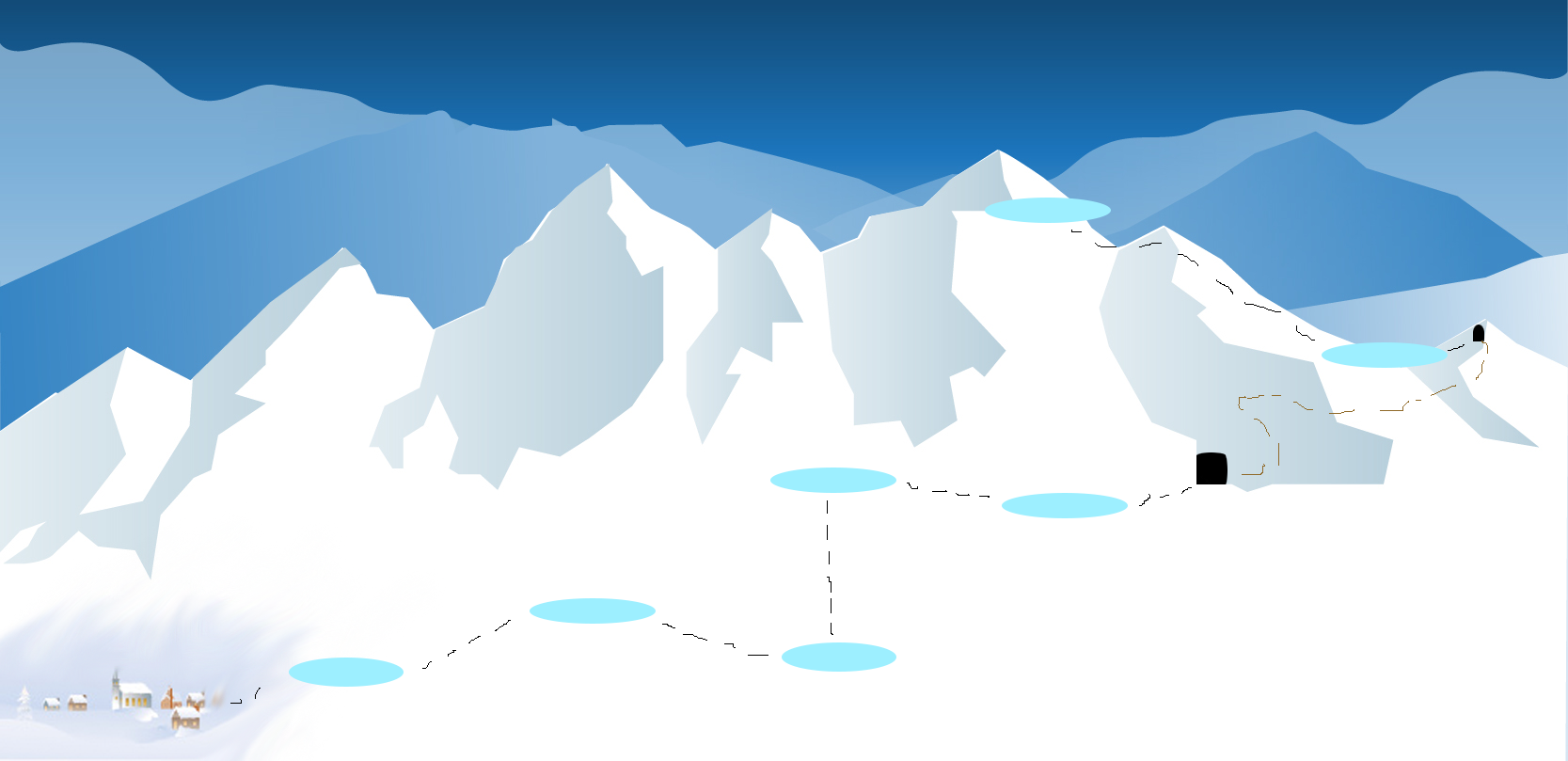
There are no custom features at this time.

### Additional Technical Information

No addition information at this time.

# Mockups / Flowcharts

The following represent mockups for game screens. Clockwise from the top left are: Main Menu, Adventure over-world map, In-game screen, and Arcade Mode menu.





Most of the assets included here are not final.

The arcade and adventure buttons on the main menu link to the two game modes respectively. Clicking on them will bring you to the next menu in line, either the arcade menu or the over-world map for adventure. Starting a level or arcade will bring you to a game screen similar to the one above, although it will include a more elaborate UI which will track score, money and ability recharge.

# Assets

### Visual Assets

|  |  |  |
| --- | --- | --- |
| What is it used for? | Filepath or location | Size |
| Character art 1 | https://github.com/lawsda/AvAdventures/tree/master/Concepts/CharacterConcept1 | 50-75KB |
| Character art 2 | ~/Concepts/CharacterConcept2 | 50-75KB |
| Character art 3 | ~/Concepts/CharacterConcept3 | 50-75KB |
| Icicles | ~/Concepts/IcicleV1 – Work in progress | 50-75KB |
| Menu Background | Work in progress | 300 KB |
| Arcade Background | ~/Concepts/Background – Work in progress |  |
| Adventure map | ~/Concepts/Adventure\_Prototype – Work in progress |  |

All visual assets are being custom made and still mostly in development.

### Audio Assets

|  |  |  |
| --- | --- | --- |
| What is it used for? | Filepath or location | Size |
| Menu background music |  |  |
| Game background music |  |  |
| Game background music |  |  |
| Character movement sfx |  |  |
| Ability sxf |  |  |
| Icicle break sfx |  |  |

All audio assets are being custom made and are still in progress.

# Features

### Crucial, Secondary, and Wish List Features (A, B, and C Features)

### A - Crucial Features

* Arcade Mode
  + At least 1 high score
* Menus
* At least 2 characters
  + Character selection
  + Fully animated and functional abilities
* Leveling
* Currency

### B - Secondary Features

* At least 4 characters
* Store to unlock extra characters and upgrade abilities
* Adventure mode intro
  + Shows map, levels, contains at least a tutorial level
* High-score board
* Music selection

### C - Wish List Features

* Expanding adventure mode
* Environmental effects
* Arcade mode customization
* Can make facebook/twitter posts from in game about high-scores

# Development Roadmap

### Storyboarding and Game Design

Complete prototype; can move character using buttons to dodge falling objects and can activate ability.

### Mobile Development 1

Week 1

* Clean up menus
  + Fonts, colors, etc
* Set up menus
  + Main, Arcade, Character select
* High score saves between plays

Week 2

* Arcade can be played with a single character option
  + Ability works
  + UI displays score, ability cooldown, currency
* Currency is collectable

Week 3

* Characters can gain experience and levels.
* Second character can be switched to.

Week 4

* Second character’s is playable
  + Can gain levels and experience
  + Can use ability
  + Changes UI to reflect new character
* Background art, sprite art, and music for the first portion of the game are complete

### Mobile Development 2

Week 1

* Character abilities can be upgraded using currency
  + Abilities gain power and new functionality through upgrades.
* Begin looking into social media connectivity
  + High-score posts to facebook and twitter

Week 2

* Begin work on additional characters
* Animations for the first 2 characters are complete

Week 3

* Additional characters have usable abilities
  + Can be switched to in character select
* Build tutorial

Week 4

* Additional characters are complete
  + At least 1 (3 total)
  + Can level and be upgraded, animated

### Testing

Make sure everything flows properly. Menus work, store unlocks items, items stay unlocked, high-scores save, etc.

Week 1

* Make sure UI scales correctly and is readable.

Week 2

* Improve the icicle spawning to try and increase difficulty

Week 3

* Restructure menus to be part of the same scene

Week 4

# References

Unity Technologies (2014) *Unity*. [Game Engine] Retrieved from unity3d.com

Xamarin (2014) *MonoDeveleop Unity.* [IDE] Retrieved from monodevelop.com