**Game Project**

Authors: Alex Henderson, Matt Jones

Title: TBD

**TOC**

1. Game Features
2. UI
3. System Design
4. Testing
5. Appendix
6. Index

**1. GAME FEATURES:**

* Genre: RTS
* Global Map:

Displays all players’ structures, military units (with different levels of visibility), neutral units, and neutral resource areas (e.g. timber, ore, gold, fish, etc.). The map is demarcated into discrete tiles (e.g. squares or hexagons in a cartesian system).

Military units may travel so many tiles based on the amount of provisioned food. Food is replenished at player/ally-owned storage structures, by successfully performing a foraging action on a valid resource tile, or by successfully looting/conquering enemy structures. If a military unit runs out of food, the soldiers will slowly begin to die of starvation or desert. Combat can be initiated from the global map when two opposing military units are on the same tile or when a military unit lands on a tile occupied by enemy structures.

Players boost their economy through constructing/ upgrading structures on the global map and winning battles.

* Economy:

All players begin with a base structure (i.e. a town/settlement) located somewhere on the global map and a starting amount of resources. This basic structure generates resources at a low rate based on the resource types of adjacent tiles.

Players can add structures to this base to boost resource production. Examples include: farms, which generate food regardless of the adjacent resources; mines, which increase ore production if the base is adjacent to one or more ore tiles (lumber yards, fisheries, and other resource-specific structures would work the same); markets, which produce gold in exchange for other resources. Players can upgrade the base structure to increase the range of its collection (example: level 1 base collects from all adjacent tiles, level 2 base collects from tiles two spaces away, and so on). Bases can be equipped with defenses (walls, towers, etc.) according to the base level. Defenses force extra measures (e.g. siege equipment) when an enemy attacks the base.

One important resource is people. Bases generate recruits slowly at first, but quicker as players upgrade the bases. Players can also build special structures to increase the population growth rate. Recruits can immediately be sent across the map as military units. They are upgraded in one of two ways: by gaining experience through combat, or by training at special military structures. Training takes some time and resources, but produces better results than combat experience. Once units have enough experience, they can specialize (i.e. become ranged, heavy troops, mounted, etc.). Some specializations may require special provisions (for example, mobile siege units require a siege works to construct and enough free recruits to operate). Players can also build structures (blacksmiths) which upgrade units’ defense and attack.

* Combat/Battle:

Engaged between two (possibly more) military units or one (possibly more) unit and structures. Players use only troops in current unit. The local map is generated based on the type of the global map tile on which combat was initiated. This local map is also demarcated into tiles.

???

* Multiplayer:

Tentatively slated for future versions and platforms.

Various additional features for multiplayer are included for increased gameplay:

- *Trade*: Trading of resources amongst players can be conducted (building name TBD).

- *Chat*: Players can communicate on a Global, Alliance, or Personal level.

**2. USER INTERACTION/INTERFACE:**

User UI devices:

*Mouse*: Main User interactive device; Point and Click on-screen action events.

*Keyboard*: Supplementary/ Advanced User interactive device — Hotkeys, numerical input, exc..

*Monitor*: 1 Monitor for game experience (Interactive graphical design to be implemented with multiplayer platforms).

**3. SYSTEM DESIGN:**

* Supported Architectures:

- x86 (Intel and AMD), ARM

* Platforms:

- Windows 7 - initial desktop prototype platform.

- Future platforms to be implemented with multiplayer features:

— OS X 10.9.5 Mavericks and higher

— MacOS 10.12.5 Sierra and future versions

- Apple iMac, MacBook Pro, MacBook Air.

— Windows 10

- All PC platforms running x86 architecture with Windows 10.

— iOS 10.3.2 and higher

- iPhone 5, 5s, 6plus, 7s.

- iPad (generations TBD).

- iPod (generations TBD).

— Android API 19/ KitKat and Higher

— Android OpenGL ES versions 2.0 and higher

*Phones*:

- Samsung Galaxy S, Alpha, Note series devices running Android OS.

- LG G, V, Nexus 4-5 series.

- Sony Xperia Z series.

- Motorola Moto X, G, Nexus 6, Droid Turbo series.

- Google Pixel, Pixel XL series.

- OnePlus One series.

- ZTE Axon series.

- HTC One M, U?? series.

- Asus Zenfone series.

- Huawei P series.

*Tablets*:

- Toshiba

* Model View Control Persistence Security (MVCPS) Component Design.
* Software application packaging format: executable with standard Windows installation manager.
* Default file directory: <User\_Space> C:/Program Files/

**4.** **TESTING:**

View:

* UI population responsiveness to be ≤ 1 second.
* Sanitize erroneous user-input fields (example: No strings for integers, no special chars of format specifiers, exc..).
* Graphical artwork Quality control (example: pixilation on window resizing, new/ overwriting artwork population, boundary collision testing, exc..).

Model:

* Game object Unit Testing

Control:

* Action Events
* Chat Moderation/ Filtering.

Persistence:

* User Data

Security:

* runtime assertions of Objects, User Data, exc..
* Chat.
* environment checking (no running in a debugger, exc..).

**APPENDIX**

**INDEX**

**Build Tools**

* Unity 4/5 Game Engine
* Microsoft Visual Studio 2015
* GitHub course and documentation control
* \*\*Unity source control for future builds possibly\*\*