

# Ninja Dojo Game

 Search this site[Home](#)[Admin](#)[Art](#)[Concept Art](#)[Reference Art](#)[Style Document](#)[Technical Docs](#)[Assembla](#)[Design](#)[Game Design Document](#)[Screen Mockups](#)[Weapons Test](#)[Milestones](#)[Sitemap](#)[Design](#) >

## Game Design Document

# Ninja Dojo

## Game Design Document

Version #3

All work Copyright © Rhys Yahata 2011.

All rights reserved.

## Copyright Information

All material contained in this document or pertaining to the design, development, or implementation of Ninja Dojo belongs solely to Rhys Yahata.

## Version History

Date	Author	Changes
5/15/11	Rhys Yahata	Initial design.
5/27/11	St. John Colon	Comments
7/4/11	Rhys Yahata	Updates for scope cut.

## Table of Contents

[Copyright Information](#)[Version History](#)

## Section 1- Game Overview

- [1.1 Game Concept](#)
- [1.2 Feature Set](#)
- [1.3 Genre](#)
- [1.4 Target Audience](#)
- [1.5 Game Flow Summary](#)
- [1.6 Look and Feel](#)
- [1.7 Project Scope](#)
  - [1.7.1 Number of locations](#)
  - [1.7.2 Number of levels](#)
  - [1.7.3 Number of NPC's](#)
  - [1.7.4 Number of weapons](#)

## Section 2 – Gameplay and Mechanics

- [2.1 Gameplay](#)
  - [2.1.1 Game Progression](#)
  - [2.1.2 Mission Structure](#)
  - [2.1.3 Puzzle Structure](#)
  - [2.1.4 Objectives](#)
  - [2.1.5 Play Flow](#)
- [2.2 Mechanics](#)
  - [2.2.1 Physics](#)
  - [2.2.2 Movement](#)
  - [2.2.3 Objects](#)
  - [2.2.4 Actions](#)
  - [2.2.5 Combat](#)
  - [2.2.6 Economy](#)
  - [2.2.7 Environment](#)
  - [2.2.8 Weapons](#)
- [2.3 Screen Flow](#)
  - [2.3.1 Screen Flow Chart](#)

- [2.3.2 Screen Descriptions](#)
- [2.4 Game Options](#)
- [2.5 Replaying and Saving](#)
- [2.6 Cheats and Easter Eggs](#)

### Section 3 – Story, Setting, and Character

- [3.1 Story and Narrative](#)
  - [3.1.1 Back Story](#)
  - [3.1.2 Plot Elements](#)
  - [3.1.3 Game Progression](#)
  - [3.1.4 License Considerations](#)
  - [3.1.5 Cut Scenes](#)
- [3.2 Game World](#)
  - [3.2.1 General Look and Feel](#)
  - [3.2.2 Location #1 – Forrest](#)
  - [3.2.3 Location #2 – Dojo](#)
- [3.3 Characters](#)
  - [3.3.1 Main Character](#)
  - [3.3.2 Sensei](#)

### Section 4 – Levels 12

- [4.1 Level #1](#)
  - [4.1.1 Synopsis](#)
  - [4.1.2 Introductory Material](#)
  - [4.1.3 Objectives](#)
  - [4.1.4 Physical Description](#)
  - [4.1.5 Map](#)
  - [4.1.6 Critical Path](#)
  - [4.1.7 Encounters](#)
  - [4.1.8 Level Walkthrough](#)
  - [4.1.9 Closing Material](#)

### Section 5 – Interface

- [5.1 Visual System](#)
  - [5.1.1 HUD](#)
  - [5.1.2 Menus](#)
  - [5.1.3 Rendering System](#)
  - [5.1.4 Camera](#)
  - [5.1.5 Lighting](#)
- [5.2 Control System](#)
- [5.3 Audio](#)
- [5.4 Music](#)
- [5.5 Sound Effects](#)
- [5.6 Help System](#)

#### Section 6 – Artificial Intelligence

- [6.1 Target AI](#)
- [6.2 Trap AI](#)

#### Section 7 – Technical

- [7.1 Target Hardware](#)
- [7.2 Development Hardware and Software](#)
- [7.3 Development Procedures and Standards](#)
- [7.4 Game Engine](#)
- [7.5 Scripting Language](#)

#### Section 8 – Game Art

- [8.1 Concept Art](#)
- [8.2 Style Guides](#)
- [8.3 Characters](#)
- [8.4 Environments](#)
- [8.5 Equipment](#)
- [8.6 Miscellaneous](#)

#### Section 9 – Management

- [9.1 Detailed Schedule](#)

- [9.2 Budget](#)
- [9.3 Risk Analysis](#)
- [9.4 Localization Plan](#)
- [9.5 Test Plan](#)

## Section 1- Game Overview

### 1.1 Game Concept

Ninja Dojo is a casual, 2-D side scroller with cute anime graphics. The player assumes the role of a young ninja who journeys into the forest to study Ninpo (The way of the Ninja). In the forest he trains at a dojo, where he attempts to learn different forms of ninjustsu. As the player trains at the dojo, the main character will gain new abilities and unlock new weapons to train with. Levels will be relatively short (one to five minutes) and will incorporate simple, intuitive controls, utilizing all the components of a smartphone. The game does not have an elaborate back-story and there are no plot elements and shifts. Ninja Dojo is meant for people who enjoy a quick game while waiting in line or riding the bus to work/school.

The player will have two options for general controls: semi-traditional joystick and buttons and an intuitive “one finger/button” mode. What makes the joystick semi-traditional is that it features a swipe pad. The swipe pad allows the players to imitate throwing the main character’s weapons by swiping his finger across the screen. The “one finger/button” mode will give the player a very unique and intuitive experience utilizing just his right index finger and left thumb. There will also be opportunities to use the microphone to imitate blow darts and accelerometer to control the main character.

### 1.2 Feature Set

#### Multiple Control Schemes

There will be two types of controls: multi touch joystick with a swipe pad and a “one button and finger” mode. The multi touch joystick will utilize an onscreen joystick that the player uses to control the basic movements of the character. There will be one large button that player can use to jump or perform other actions (like Legend of Zelda Ocarina of Time [the A button]), one button to switch weapons, and a swipe pad the user uses to throw weapons. The “cool” feature of the joystick mode is the swipe pad that players will use to control projectile weapons. To do this, a

player will swipe/flick his thumb over the swipe pad in the direction he wants to throw the weapon relative to the character (taking into account the velocity of the swipe). This gets rid of the cumbersome two-joystick layout, which is hard to use.

(Update 7/4: The One finger mode will not be implemented. However, it will be incorporated in a future release.) For the players who want a simpler interface, there will be a “one button and finger” mode. For this mode, the player will control the character by dragging his finger across the screen. The character will move in the direction that the user is dragging his finger. To jump the user will quickly move his finger upward. For throwing weapons, there will be a button that the player presses or holds with his thumb to arm the weapon. He then can swipe this finger in the direction that he wants to throw that particular weapon (taking into account the velocity of the swipe).

[St.John]: Be sure to include a screen in options where you can switch from "One Finger" navigation to "Classic" dPad navigation.

## Intuitive and Simple UI

To keep things simple, Ninja Dojo will try to minimize the usage of text. Any text (there may be exceptions to this) will be stored statically in an external XML file. This way Ninja Dojo can support multiple languages by just choosing which XML file to use.

[St.John]: Never use text on an in game asset if you can use an icon. Re-doing texture maps is a production killer. If you must use text for assets or for the UI, you want to use text strings pulled from an excel document. This makes language localization much easier - even if you \*think\* you are only publishing in one language... trust me, you will port to another language. When you do, you only have to send an Excel doc out for translation instead of source code. Talk to me if you need clarity on this. For example:

Wrong: Text = "Now my revenge is complete!"

Right: Text = "String:EN\_Quote!R17" (the value from cell R17 of the English language strings excel doc. What gets displayed is "Now my revenge is complete!")

## Main Menu

For the main menu, very basic text will be used. For example, simple English words and phrases like “start”, “continue”, “level select”, and “options” will be used.

## Pause Menu

For the pause menu, common, easy to relate to images will be used. For example, the return to play button will be an “X”, the restart level will be an arrow that forms a circle to point to it’s tail, and the return to main menu will be an image of a menu. This makes for a very simple and universal language menu system.

## HUD

The HUD will remain graphical for the most part. The player’s health and inventory will be graphically represented with small sprites. For HUD elements requiring the use of numbers to present various information like quantity or elapsed time, Arabic number will be used.

## Use of Other Device Hardware

Ninja Dojo will also try to incorporate other components of the phone during game play. (Update 7/4: All of the below features will NOT be implemented in this release. However, these features will be added in future releases and or updates)

### Accelerometer

During a mission that requires the player to walk across a tight rope, the accelerometer can be used instead of the normal controls to imitate balancing. This allows the user to very naturally and easily control the player’s movements.

### Microphone

When the player equips the blowgun, instead of pressing a button to fire it, the player will blow into the microphone to simulate using an actual blowgun.

## 1.3 Genre

Ninja Dojo is a 2D casual side-scroller. The parallax effect with non-dynamic lighting will be used for the camera view. The camera should look and function like any classic SNES side-scroller (e.g. Super Mario World, Contra III, etc.)

## 1.4 Target Audience

Ninja Dojo's target audience is casual gamers. These are people who enjoy a quick, fun game while waiting in line or riding the bus. Ninja Dojo should also appeal to the female audience because of cute graphics and game play with no gore. Since levels will be relatively short (one to five minutes at max), players can feel a sense of accomplishment and enjoyment within less than ten minutes of playing the game for the first time. Short levels also allows the player to pick up the game for a couple minutes, play it, and get a sense of reward and accomplishment by being able to complete a level without having to spend countless hours. Our targeted age range will be from 13 – 30. Ninja Dojo may appeal to younger or older players who enjoy games, but they are not the targeted audience.

## 1.5 Game Flow Summary

The game will be split up in to many short, small levels. The requirements for completing each level will be different. For example, since the game is about a young ninja in training, the first mission will be to learn how to control and move the character, whereas the next level objective will be to hit a certain number of targets with throwing knives. Players should be able to complete a level in one to five minutes.

There will be a score for each level and a ranking based on performance. A player's performance depends on how fast he completed the level, number of techniques/weapons were used, and accuracy. All these elements determine how "ninja like" the player was in the level. For example, if a player completed a level very quickly but wasted many shuriken or used many ninjutsu, then it would not be very "ninja like" and thus his score would drop. As the player completes levels he will unlock more training missions and gain access to new weapons and techniques. Before being able to use these techniques, the player must master them in training, which can consist of one to five levels, depending on the weapon or skill. (Update 7/4: Through research I found a new scoring system which I find more appropriate for Ninja Dojo. Instead of tracking movements and playing style, score will be purely based on accuracy [ratio of number of weapons used to number of targets] and elapsed time.)

## 1.6 Look and Feel

Ninja Dojo should have a very fluid feel. The player should feel though as if he/she has a good control over the character even though he/she is using a touch screen to control the main character. The overall game graphics should have a look that resembles Angry Birds, but still plays like a classic SNES side-scroller (especially those from the Super Mario series). Characters and objects will have a very cutesy look and game play should be relaxed and enjoyable.

[St.John]: For homework, check out "Urban Ninja". I think your game differentiates well in that it makes better use of "Ninja-like" features and player skill.

## 1.7 Project Scope

### 1.7.1 Number of locations

There will be two locations: the dojo/training grounds and the forest surrounding it.

[St.John]: Erica should create inspiration art of these main environments

### 1.7.2 Number of levels

Optimistically there will be 10 levels for the alpha build, but if time does not permit then there will be at least 5.

### 1.7.3 Number of NPC's

There will be 3 types of NPCs:

#### Sensei - 1

There will be one master sensei that teaches the main character.

#### Targets - 4

When practicing with weapons, instead of aiming at people the player will aim at targets instead. These targets will vary in size but will all be made out of wood and have a recognizable bull's-eye shape on them. There will be a small, medium, and large circular target as well as straw scarecrow shaped target.

#### Traps – 2

Although these are not characters, there will be traps in the game that can hurt the main character. In future version of Ninja Dojo there will be enemies but for now the traps will serve as enemies. The first type of trap is a bamboo spike.

These are sharpened pieces of bamboo that are placed on the ground or shoot out from a wall, ceiling, or the ground. The next type of trap is an automatic arrow launcher. This can either be fired on a timer or activated by a sensor. (Update 7/4: Traps will not be implemented for this release. However, they will be incorporated in a future release and or update.)

[St.John]: Erica should create concept art of these main NPC's as well as the Ninja. Q: Will there be more than one type of Ninja the player can choose? E.G. male/female? Classic archetypes? (Update 7/4: There will only be one main character with a fixed archetype and gender.)

## 1.7.4 Number of weapons

There will be 4-5 weapons in Ninja Dojo.

### Kunai – Throwing Knives

Kunai are a short-range projectile weapon with high damage that travels medium speed. The player will have to take gravity into account when using this weapon. If time permits, kunai will be designed such that the player can also choose to use them for slashes and stabs like a dagger. (Update 7/4: For this release the player will no be able to cut and slash enemies with the throwing knife. This feature may be released in an update or a new, separate weapon like a sword will be added.)

### Shuriken – Throwing Stars

Shuriken are a medium range projectile weapon with medium damage that travels at medium speed. The player will have to take gravity into account when using this weapon but not as much as kunai.

### Guni – Throwing Needles

Guni are a short to medium range projectile weapon with low damage that travels at high speed. Guni will hit the target almost instantly, therefore gravity will not effect it.

### Fukiya – Blow Dart

Fuikiya are a short to medium range projectile weapon with low damage that travels at high speed. Darts will be affected by gravity. (Update 7/4: This weapon will not be included in this release. However it will be included in a future one.)

### Kaginawa – Climbing hook and rope

This is more of an aide weapon that the player will use to get to climb to high platforms or swing across large gaps. (Update 7/4: This weapon will not be included in this release. However, it will be included in a future one.)

[St.John]: Erica should create concept art of these main weapons

## Section 2 – Gameplay and Mechanics

### 2.1 Gameplay

#### 2.1.1 Game Progression

The game will progress like Angry Birds. After a player completes a level/mission he/she will unlock the next one. Levels/missions will be grouped into sets and a player will unlock the next set one he/she has completed all the missions in previous sets.

#### 2.1.2 Mission Structure

Missions will be structured like Angry and the Super Mario Brother series. Levels will be short and can be completed with ease in about one to five minutes. There will be a clear objective in the beginning and the player will accumulate points based on how he/she completes the mission.

#### 2.1.3 Puzzle Structure

There will not be complex puzzles like the Zelda series that challenge the player, but rather there is a ranking system that assigns a letter rank based on the player's score. Score will depend on how "ninja like" the player completes the mission. The "ninja like" score factor is based on how fast and effectively the player completes the mission. The time aspect is similar to the Super Mario Brother series, except there is no time limit. So, the faster the player completes

the mission the higher the time portion of his/her score will be. The effective aspect will depend on two things: how many weapons he/she uses and how effective he/she is with the weapons; this is very similar to Angry Birds where the score depend on how many point combos the player gets plus bonus points for extra birds not used. Effectiveness will be calculated based on how the player decides to approach the mission. For example if the mission requires a player to bypass traps to get to a goal, a satisfactory approach would be to reach the goal but set of a few traps. A better approach would be to disable the traps and reach the goal. The best approach would be to reach the goal without setting off or disabling any traps. (Update 7/4: After doing research and design, the scoring structure will be changed for this release. A player's score will be based on two things: elapsed time and number of weapons used. [There will be a couple more factors like traps set off and lives remaining but because there are no traps for this release the score will not depend on it] Like Angry Birds, a 3 star system will be used. The only factor that will affect the star rank is the elapsed time. Two predetermined "target scores" will be set for each level; one score will determine the max elapsed time for three stars and another for two stars. Anything below those two target scores will get only one star. An additional "medal" will be attached to the player's stars if he/she is able to complete the level with only using the minimum number of weapons required. For example, if there are only three targets in the level, a perfect score would be only using three throwing stars to hit the targets. This gets rid of the complicated point system and clearly shows the player what is required to get a completely perfect score, something that I feel Angry Birds does not do a good job of.)

### 2.1.3.1 Basic Scoring Algorithm

The to calculate the optimal time, Dijkstra's Algorithm will be used. The player will be the starting node and all other nodes will be the targets. The edges of the graph will be calculated based on how long it would take to travel there. In calculating these times, it can be assumed that the player will be moving at max speed. For targets that are out of the player's reach (i.e. too high) calculate the travel time by taking into account any detours that it would require so that the player can reach the target. These detours will then need to be re-factored into the graph and Dijkstra's Algorithm should be recalculated.

### 2.1.3.2 Scoring Adjustments

The target time that is calculated by Dijkstra's Algorithm should be ideally used as the lower bound for the three stars scores, but that may not always be the case. For example, if a target is directly above a player but still within his throwing range, Dijkstra's Algorithm would factor in the time it would take to get there. For these kinds of cases, the level designer needs to step through his level and manually calculate it. Another work around would be to chose the

nodes as the spots the level designer would like the player to be in when attacking the target. That way it cuts down on the unnecessary travel time to the actual target.

[St.John]:every weapon and obstacle will need a value defined that contributes to the "ninja like" score (probably should call this value "Ninpo"). This will need to be balanced using a soak sheet. I will explain soak sheets later.

## 2.1.4 Objectives

Each level will have its own set of objectives. For example, the first level's objective is to travel through the forest to reach the dojo. The overall objective of the game is to study and learn Ninpo and master all the techniques. This is accomplished by completing the various training levels for a certain weapon or technique.

## 2.1.5 Play Flow

When a player plays Ninja Dojo for the first time, he/she will immediately plunge directly into a game play. There will be no lengthily cut scenes to watch or complex instructions to read. The general underlying flow of each mission will be the same. There will be brief graphical instructions that describe to the player what the mission objectives are. The player will then immediately dive right into to completing the mission. If he/she completes all the objectives without dying, then a clear screen with the player's score will appear. If the player dies before completing the mission a fail screen will popup and the player will have the option of returning to the mission select screen or restarting the mission from the beginning. Since levels are relatively short there is not need for save points during a mission. There will be a pause function, but there is now way to save the current state of gameplay and return after exiting the application.

[St.John]: Every level requires difficulty ramping. Read this Gamasutra article to insure Ninja Dojo achieves this requisite.

## 2.2 Mechanics

### 2.2.1 Physics

Physics for Ninja Dojo will try to be as realistic as possible. There will be gravity, friction, inertia, and harmonic motion (simple and damped). Physical factors that will not be accounted for are wind resistance and drag. For calculations, a relative weight system will need to be used. This will be decided upon later during implementation and tweaked to provide the desired results.

## 2.2.2 Movement

Movement will feel similar to Super Mario 64 but two-dimensional instead of three-dimensional. The player should be able to easily control the speed of the character by either dragging his/her finger faster across the screen or pushing the joystick farther in the desired direction. Since there is friction, there will be a skid/slide if the player abruptly changes direction of movement or does a rapid acceleration in a particular direction.

## 2.2.3 Objects

### 2.2.3.1 Picking Up Objects

For the scope of the project, the player will have limited ability to interact with his surroundings. He cannot pick up common objects like rocks or branches. His extent of interacting with the environment objects is

### 2.2.3.2 Moving Objects

For the scope of the project, the player will not be able to move any objects. In future versions the player will be able to push, pull, or carry common objects to help him reach places or confuse the enemy.

## 2.2.4 Actions

### 2.2.4.1 Using weapons

The player can control use weapons based on control scheme. See section 5.2 for more detail about the different control schemes.

## 2.2.5 Combat

There are no fight scenes or boss battles in this version of Ninja Dojo. Combat will be more one sided. For example some missions will require the player to hit a certain number of targets. Thus the combat is more target practice like. Another different kind of level could require the player to avoid getting hit by various traps, which is more of a survival

obstacle course. Of course, there will be hybrid levels that require the player to hit targets while avoid traps. (Update 7/4: No traps in this release)

## 2.2.6 Economy

Since there is no currency or purchases done in the game, there really is no need for an economy.

## 2.2.7 Environment

For the scope of the project, the player will have limited ability to interact with his surroundings. His extent of interacting with the environment is climbing walls, trees, and vines.

## 2.2.8 Weapons

### Kunai

Kunai are metal throwing knives. They are one of the most basic weapons in the main characters arsenal. They can be thrown short to medium distances and can be lobbed because of their weight. Kunai are thrown one at a time and fire can be thrown at the same rate as a normal person firing a revolver. They can be thrown at a variable speed but at max can be thrown at a medium speed.

Range: Short to medium (720 px)

Speed: Medium (240 px / sec)

ROF: Single shot with minimum cool down (.5 sec)

Weight: Moderate (80 px / sec<sup>2</sup>)

### Shuriken

Shuriken are metal throwing stars. They are one of the most stereotypical, basic weapons in any ninja's arsenal. They can be thrown medium distances with moderate to high accuracy and speed. They are relatively light and thus their flight path is not heavily affected by gravity. Depending on the time here is the ROF scheme: Shuriken can be thrown in three shot bursts; there is a moderate cool down between bursts and if the player decides to throw slowly they can be thrown single shot. Shuriken can be thrown at a variable speed but at max can be thrown at a medium to high speed.

Range: Medium (1024 px)

Speed: Moderate - fast (400 px / sec)

ROF: Single shot with moderate cool down (2 sec max delay)

Weight: Light (60 px / sec<sup>2</sup>)

## Guni

Kunai are thin metal throwing needles. They are not a typical ninja weapon but are one of the basic weapons that the main character can use. They can be thrown a short distance with high accuracy. Due to their thin and aerodynamic shape, guni are not affected at all by gravity. Depending on the time here is the ROF scheme: Guni can be thrown one at a time or charged for a five shot burst with moderate cool down time. They cannot be thrown at a variable speed and travel at high speeds.

Range: Short (480 px)

Speed: Fast (instant)

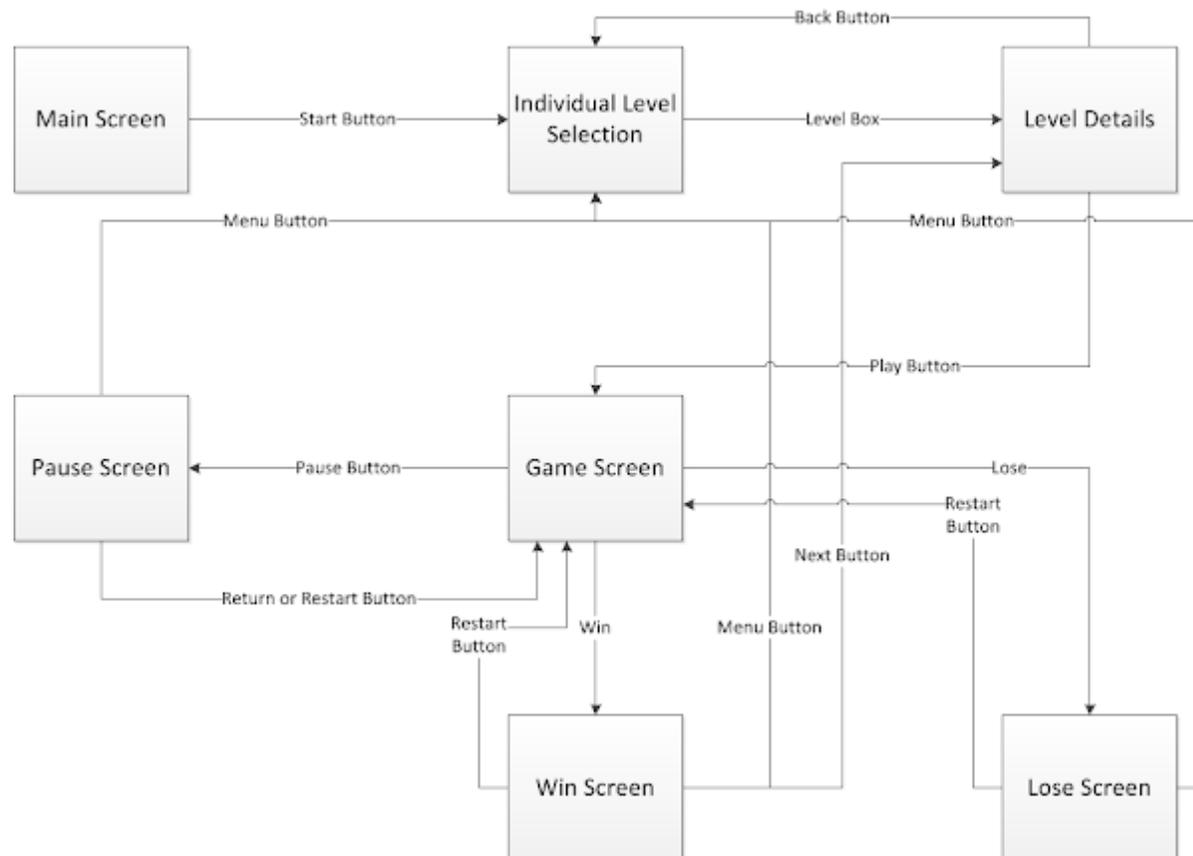
ROF: Single shot with minimum cool down or charged burst with moderate cool down (1.5 sec delay)

Weight: None

## 2.3 Screen Flow

### 2.3.1 Screen Flow Chart

See "Screen Mockups" for images of the screen. The screen progression will be relatively simple and should not require a flow chart to figure out what button will take you to what screen.



[St.John]: You will need to create a wire-frame of menu progression. An example wire-frame can be found here.

A great utility to create this is gliffy.com.

### 2.3.2 Screen Descriptions

**Main Screen** - This is the main screen that the player will see when he/she starts the game. The start button will take him to the level selection screen.

**Individual Level Selection** - This screen is very similar to ones found in games like Angry Birds and Cut The Rope.

Tiny squares will have the level number and the number of stars that a player has received on it. Locked levels will have a pad lock.

Level Stats/Break Down - This screen will show you the current high score (if the level was played) and what kind of elapsed time is need to get the different number of stars.

Gameplay HUD - See section 5.1.1

Pause Screen - The array of options that you will see when the game is paused. There is a resume gameplay, restart level, and return to menu.

Win Screen - Screen that you will see if player wins the level. There will be a choice to redo level, continue, or go back to menu. The score that was achieved will be shown.

Lose Screen - (Update 7/4: Is it possible to loose?)

[St.John]: See above.

## 2.4 Game Options

- (Update 7/4: For this release none of these options will be present. However in a future release all these options will be implemented)
- (Update 7/5: Clear data option might be added if it will fit into the screens.)
- Sound on/off – turn the background music and sound effects off.
- Change style of controls – change between the “one button and finger” mode and the joystick mode.
- Clear data – delete all scores and unlocked levels

## 2.5 Replaying and Saving

All levels/missions are repayable just like Angry Birds but saving the game while playing a level is not allowed. Since levels are relatively short, it shouldn't hurt the player too much to not be able to save state during gameplay.

## 2.6 Cheats and Easter Eggs

For the scope of this project there will be no cheats or Easter Eggs. However, for future versions there will be achievements similar to those found on Xbox 360 games.

## Section 3 – Story, Setting, and Character

## 3.1 Story and Narrative

Ninja Dojo is a casual game so there will be no immersive story and long cut scenes. This will allow players to easily dive straight into the game without having to play/watch a 30-minute cut scene to get to the first opportunity to save (Like most RPGs).

### 3.1.1 Back Story

The player assumes the role of a young ninja in training. He is young and headstrong; basically, your stereotypical hardworking student/hero. He travels into the forest in search of a legendary dojo to learn and study Ninpo. In the beginning of the game, he doesn't have many abilities. He can perform basic movements such as running, jumping, and climbing. However when he reaches the dojo and starts training, he gains access to many different weapons and learns new techniques.

### 3.1.2 Plot Elements

Since there is no antagonist of the story there really isn't a tension build and a resolution. Refer to the section on Game Progress to see how the game progresses.

### 3.1.3 Game Progression

Since there are no plot elements, the player gets the sense/feeling of progression through completing the various training missions. For the project scope, there will not be many levels as the main focus is getting a solid game engine. For future versions of Ninja Dojo, there will be groups of levels like Angry Birds and the player will progress in game similarly. Because levels are disjoint for the most part updates could easily add more levels to groups or even a new group of levels.

### 3.1.4 License Considerations

[St.John]: This includes use of other company's intellectual properties, such as "Cheerios", "Mickey Mouse", etc. Nobody owns the rights to "Ninja", so you are in the clear. Make use of Google to search for questionable things. You'd be surprised... things like "One Up" may be copyrighted.

### 3.1.5 Cut Scenes

For the scope of this project there will be no cinematic cut scenes.

## 3.2 Game World

### 3.2.1 General Look and Feel

The game world should have a clean, cheerful look to it. The graphics should resemble that of a well done 2D side scroller for Gameboy Advance. The images should be clean and crisp with low to moderate texture detail. The world should be semi-realistic, easily recognizable, and stereotypical so that players can easily relate to it.

### 3.2.2 Location #1 – Forest

#### 3.2.2.1 General Description

This is the forest that the dojo is hidden away in. The forest is not uninviting and creepy, but is untraveled and mysterious. It is very similar to the forest that surrounds Konoha in the Naruto anime series. It is very green and lush and ample sunlight fills the forest making it very visible during the day.

#### 3.2.2.2 Physical Characteristics

There is a lot of vegetation, but not so much to the point where it looks “out of control.” The forest looks as if someone is constantly grooming it, keeping it in a perfect, pristine shape. The grass is of medium to short length and it grows plentifully on the ground. Although the forest does not give off a creepy aura, there is still an over all mysterious vibe. This can be accomplished with the abundance of vegetation and trees. The closer the character is to the dojo, the more the surround looks like there is some one who lives in the forest. In other words, at the entrance of forest it seems that all the forest is a lush, vegetation dense habitat and not a training ground.

[St.John]:Your TDD should discuss how you plan to deal with environments. Single background image? Will there be parallax?

#### 3.2.2.3 Levels that use the area

Levels: 1,

### 3.2.2.4 Connections to other areas

In the heart of the forest lies the dojo.

## 3.2.3 Location #2 – Dojo

### 3.2.3.1 General Description

The dojo that the main character trains at is your stereotypical traditional Japanese rural dojo. A good example of this is the dojo that Kenichi trains at in the manga/anime series Kenichi: The Mightiest Disciple. The dojo is well kept and has a very majestic look.

### 3.2.3.2 Physical Characteristics

The dojo is surrounded by the lush forest and is penned in by a medium/low height wall, with traditional Japanese shingles on the top and a large metal and wooden gate used as the only entrance/exit. The dojo is a single story, traditional Japanese house. The dojo is primarily made out of wood. Parts that are not painted like the outside floors have a natural wood finish with a nice clean sheen. The room floors of the dojo are tatami mats and doors are shoji screens with black lacquered wood. The actual dojo building only takes up part of the actual fenced in grounds. There are training areas with targets and straw dummies. There is also a pond and few trees scattered through the dojo grounds

### 3.2.3.3 Levels that use the area

Levels: 2,

### 3.2.3.4 Connections to other areas

The forest surrounds dojo.

## 3.3 Characters

Due to the casual nature of Ninja Dojo, there aren't complex/deep background stories for the characters.

### 3.3.1 Main Character

#### 3.3.1.1 Back Story

The main character left his home and journeyed to the forest to learn and study Ninpo. It is his dream to become a great ninja and help those in need.

#### 3.3.1.2 Personality

The main character is your stereotypical hardworking pupil. He isn't a super gifted genius, but rather a hardworking young boy with a strong spirit and sense of justice. He is not timid and is very charismatic. He is very driven and does not easily give up. In this regard, he is rather stubborn, but in a good way. Although he isn't naturally talented, he has great potential and the sensei see this.

#### 3.3.1.3 Look

The main character is a young boy, about ten years of age. Looks a little bit like a trouble maker/rascal but not so much like Naruto. He is of average height and build with short black hair and big eyes.

#### 3.3.1.4 Special Abilities

N/A

#### 3.3.1.5 Relevance to the Game Story

He is the main character that the player plays as.

#### 3.3.1.6 Relationship to Other Characters

He is Sensei's only current pupil.

### 3.3.1.7 Statistics

N/A

## 3.3.2 Sensei

### 3.3.2.1 Back Story

Sensei is an old, retired ninja. In his prime, he was the best ninja. He used to teach but also retired from that. Because of the main characters spirit and desire to learn, Sensei decided to come out of retirement just to teach him.

### 3.3.2.2 Personality

Sensei is a kind, peaceful old man. He somewhat philosophical, like many old martial art sensei. He isn't strict or mean, but it is stern and is not afraid to be a little harsh on his pupils. He has a personality somewhat similar to the head sensei in anime series Kenichi: The Mightiest Disciple.

### 3.3.2.3 Look

Sensei's image is highly based of the great Aikido sensei, Morihei Ueshiba. He has a white hair (bald on the top, hair only growing on the back and sides), a long beard, and has a few wrinkles. He is usually smiling and happy and does not look threatening or frightening. He walks with a walking stick, but is still nimble (although he does not look so). He is not really frail, but is of a thinner build.

### 3.3.2.4 Special Abilities

N/A

### 3.3.2.5 Relevance to the Game Story

Is the person giving instruction before the start of a mission.

### 3.3.2.6 Relationship to Other Characters

Teaches and trains the main character.

### 3.3.2.7 Statistics

N/A

## Section 4 – Levels

### 4.1 Level #1

#### 4.1.1 Synopsis

Original Design for 1st level:

Eager to start training, the main character leaves home and journeys to forest that the dojo in. The main character must navigate through the forest and find the dojo.

(Update 8/2/2011): The first level will teach the player the basics of controlling the character. There are no plot elements mixed into to this level.

#### 4.1.2 Introductory Material

There will be graphical instructions on how to control the player. It will show how to control movement for both control schemes. The basic movements that will be explained in the beginning. (Update 8/2/2011): Depending on the time, there may not be any images. This is not ideal.

#### 4.1.3 Objectives

##### High Level

Original Design for 1st level:

The player needs to help guide the main character through the forest to find the Dojo that he wants to train at.

(Update 8/2/2011): The objective of this level is to get the player acquainted with the controls.

## Player

The player should understand basic movements such as walking, running, and jumping. (Update 8/2/2011): Attacking included.

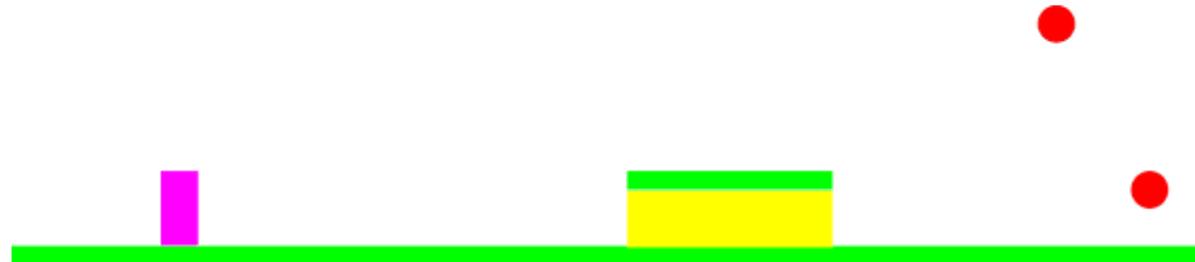
### 4.1.4 Physical Description

This level will pretty much be a straight shot from start to the finish. The player will have to jump at certain points in the level and climb up to the top of a tall platform. (Update 8/2/2011): No climbing for this release and there are two targets for the player to hit.

### 4.1.5 Map

Key:

- Purple: Player
- Green: Grass
- Yellow: Dirt
- Red: Target



## 4.1.6 Critical Path

There are two paths for this level. The first is the typical path that is not optimal because the player will likely try to become familiar with controlling the player and not move and attack optimally. The typical path is the player running forward to the hill and jumping on top. While on top the player can throw a knife at the first target (one higher in the air) and then when he jumps off or gets closer to the edge he can throw a knife at the second target.

The optimal path is as follows: The player moves forward (112 pixels) and throws a knife at 45 degrees to hit the second target (lower one). He then moves forward (243 pixels) and throws another knife at 45 degrees to hit the first target. Both knives need to be thrown at full speed.

Here is the math:

Distance from player start to second target (a) =  $960 - 128 = 832$

Distance need to move in order to be within range =  $a - 720 = 112$

Travel time =  $112 / 150 = .75$  seconds

Time for throwing knife =  $720 / (240 * \cos(45)) = 4.24$  seconds

Calculate how long it takes for the knife to hit first target given that it is in range.

$240 * \sin(45) * t - .5 * 80 * t^2 = 142$

$40 * t^2 - 169.71 * t + 142 = 0$

$t = 1.14658, 3.09617$

Knife travel distance =  $240 * \cos(45) * 3.09617 = 525$

Distance needed to travel to get in range of second target =  $880 - 525 - 128 = 227$

Travel time =  $227 / 150 = 1.513$

Time for throwing knife = 3.096

$.75 + 4.24 = 4.99$  seconds at most optimal

Math for typical:

Scoring

- 3 stars  $\geq 6$  seconds
- 2 stars  $\geq$

## 4.1.7 Encounters

No encounters.

## 4.1.8 Level Walkthrough

## 4.1.9 Closing Material

None

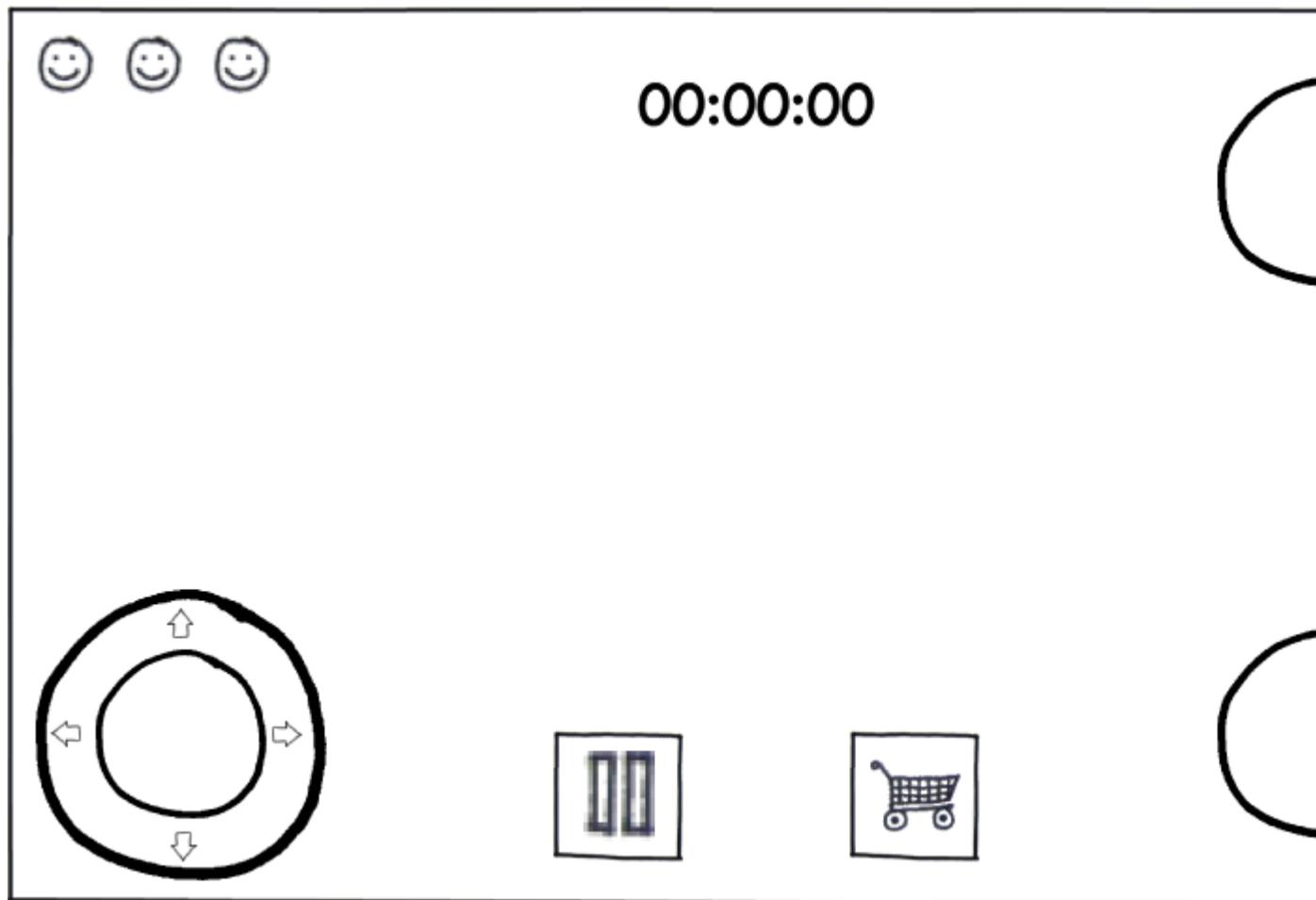
# Section 5 – Interface

## 5.1 Visual System

### 5.1.1 HUD

- Health – This will be displayed as heart sprites. When a player takes damage, the corresponding number of hearts will disappear. (Update 7/4: Since there are no traps for this release there will be no health.)
- Inventory – There will be a small sprite icon of the current weapon equipped and a number next to it to indicate the quantity. If the quantity is infinite then there will be no number. (Update 7/4: Since the inventory feature was cut for this release, this will not be implemented in this release.)
- Score – The gross points scored will be kept, constantly updated, and displayed in Arabic numbers. (Update 7/4: Since the scoring system may be changed this will no be displayed and this may get cut.)
- Time – The elapsed time will be kept in seconds with accuracy up to the nearest hundredth of a second, but only whole seconds will be displayed. A stopwatch icon will appear to the left of the elapsed time.
- Buttons
  - Pause – Circular onscreen button with the glass parallel lines as the icon.
  - Action button – Semi transparent, large, circular button. No text on it.
  - Attack button – Semi transparent, medium, circular button. Sprite of a sword on it to symbolize attacking.
  - Swipe area – Semi transparent, large circle. Has eight triangles placed on it to form a shape that resembles a compass.

- Inventory button – Semi transparent, medium, circular button. Sprite of a backpack on it to symbolize inventory. (Update 7/4: Since the inventory feature was cut for this release, this will not be implemented in this release.)



[St.John]: excellent synopsis of features. Hud doesn't need a wireframe, but should have a button diagram.

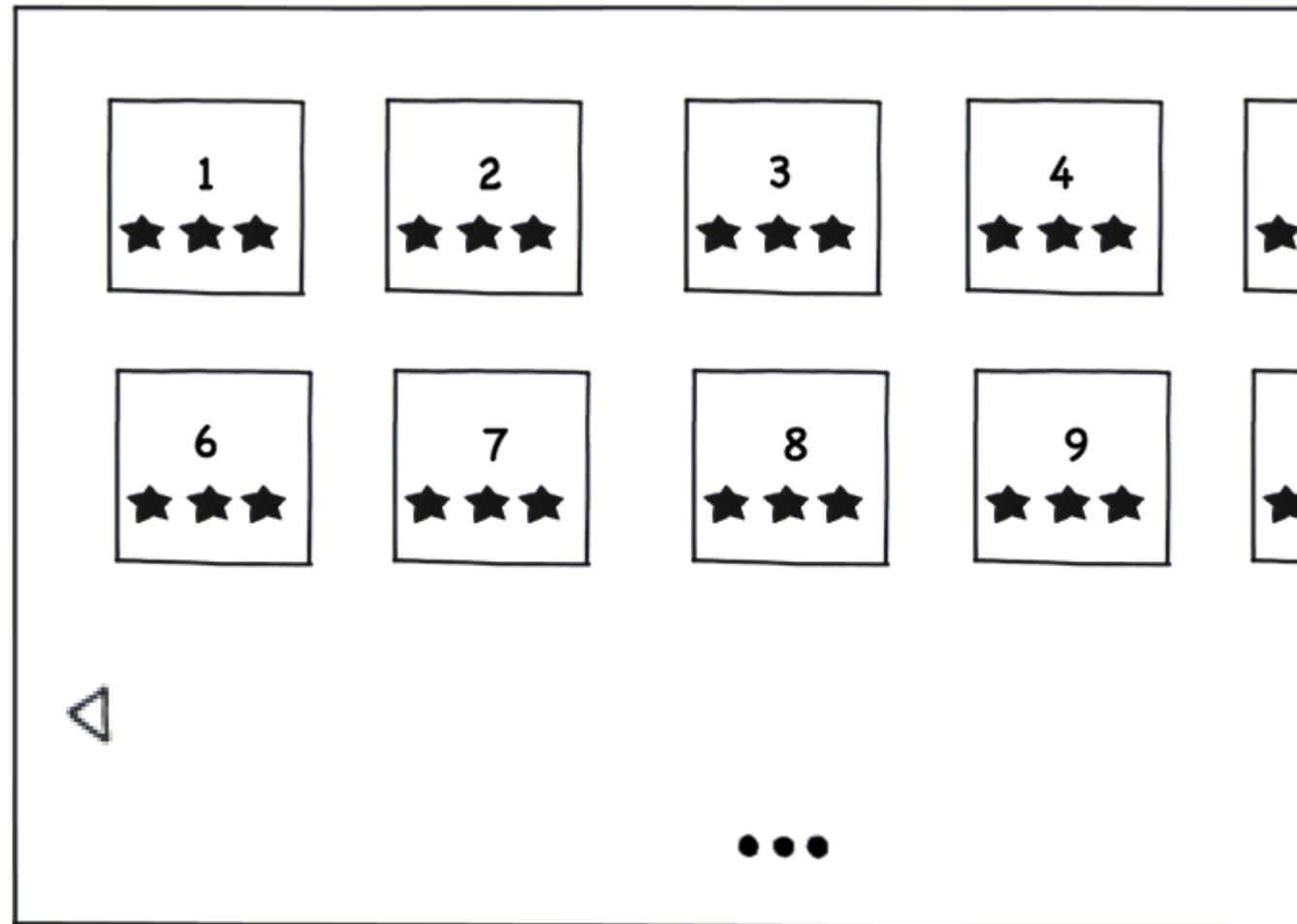
### 5.1.2 Menus

- Start – Static image with two buttons: play, options (Update 7/4: No options button)

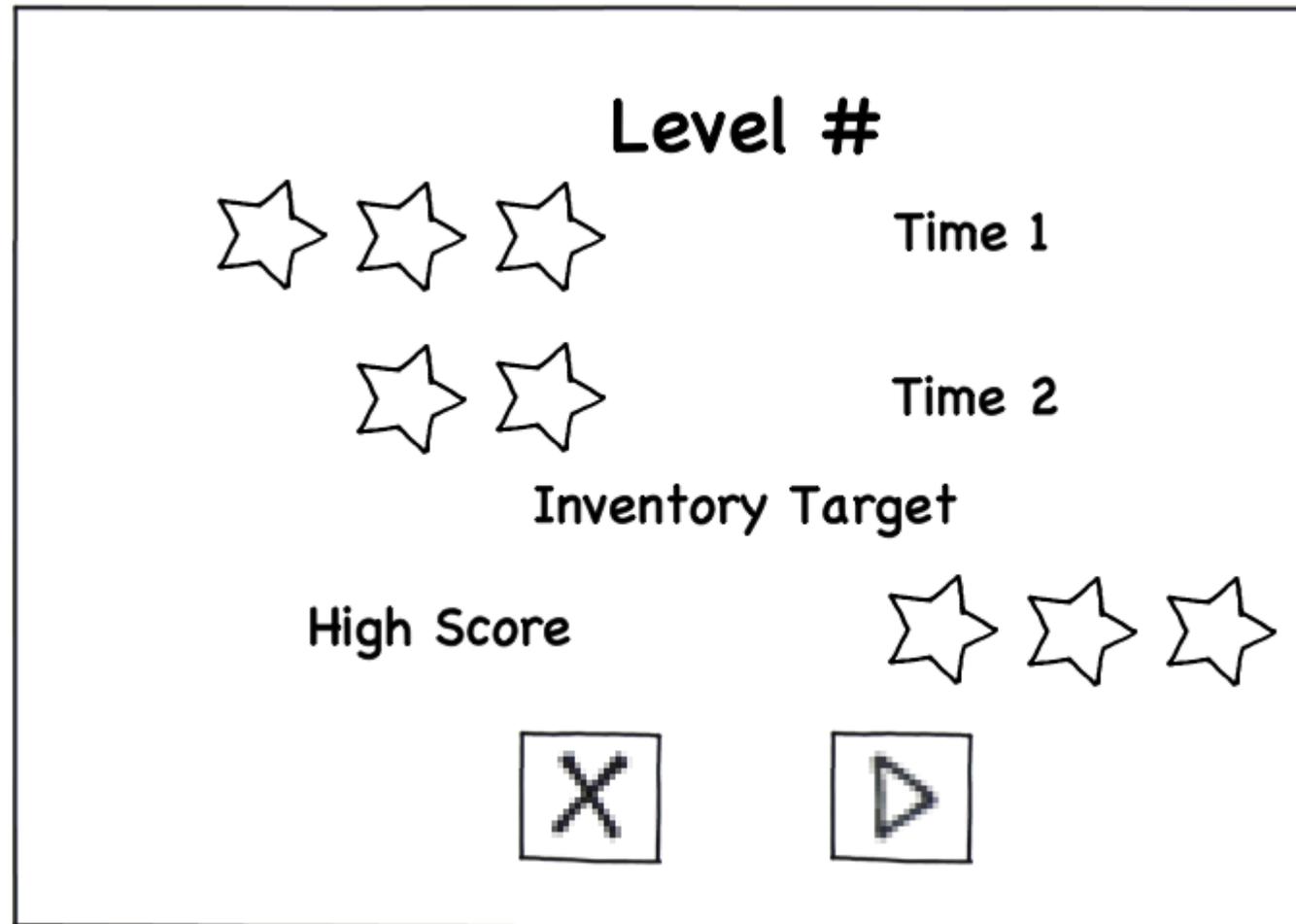
# Ninja Dojo

**Start**

- Options – Turn sounds on/off, Switch control schemes.
- Mission select – Select a group a missions. This menu will look and function very similarly to the Angry Birds group selection menu. (Update 7/4: Since many levels may not be created for this release, this screen will not appear until a later release.)
- Individual Mission/Level select – Select a mission/level to play. There will be a grid of squares with a number for the level number and a letter for the rank. Locked levels will have a padlock on them.



- Individual Level Breakdown - This screen will display the current score of the level (if completed) and show what scores are necessary to get two or three stars and what is considered to be 100% accuracy.



- Pause – Back to main menu, restart level/mission, options, and return to game. Each menu option will be a button with an icon.
  - Back to main menu – A square with three dots and lines arranged such that the lines are parallel and running horizontal. The dots will be lined up in one column and the lines in another. There will be one line and dot per row. Similar to the Angry Birds button.
  - Restart – An arrow that forms a circle to point to its tail.
  - Options – A gear/cog. (Update: Since there are no options in this release this will not be implemented)
  - Return to game – A circular button with an X on it.



- Inventory – A grid containing the sprites for various weapons/techniques that player has equipped. A check mark will be placed next to the sprite of the weapon/technique that is currently equipped.
- Win – This screen will be displayed if the player completes the level. It will display the number of stars that the player has earned and whether or not the player completed the level with perfect accuracy.
  - Restart Button – Will load the same level
  - Menu Button – Takes the player back to the individual level selection screen.

- Next Button – Takes the player to the level details screen for the next level.



- Lose – This screen will be displayed if the player fails to complete the level.
  - Restart Button – Reload the current level
  - Menu Button – Takes the player back to the individual level selection screen.



[St.John]: excellent synopsis of features. Be sure to map them out on the wireframe/ flow chart.

### 5.1.3 Rendering System

Ninja Dojo will use an OpenGL ES 2.0 rendering system. This rendering system will run on a separate thread from the game play and will refresh every 16-32 milliseconds (30-60 fps). Only images that are in view area of the camera will be rendered and displayed. All images will be sprites so there is no need to do any dynamic image processing like shaders, particle FX, or lighting FX. Images will be processes and sliced up into vector buffered objects.

### 5.1.4 Camera

The camera will be a side view, dynamic movement, non-zooming camera. The camera will always remain focused on the player, keeping him in the center of the screen. There will be no zooming in or out. When the player reaches the end of the level the player will still remain in the center of the screen. If extra “filler” needs to be added to prevent the player from going out of bounds, some sort of wall/barrier will be placed. The image will extend (or a background image will be placed) so that the player doesn't see empty space.

### 5.1.5 Lighting

For the scope of this project there will be no dynamic lighting. Shadows and highlights will be part of the sprites.

## 5.2 Control System

### Joystick

There will be a multi touch onscreen joystick that the player uses to control the basic movements of the character. There will be one large button that player can use to jump or perform other actions (like Legend of Zelda Ocarina of Time [the A button]), one button to switch weapons, and a swipe pad the user uses to throw weapons. The “cool” feature of the joystick is the swipe pad that players will use to control projectile weapons. To do this, a player will swipe/flick his thumb over the swipe pad in the direction he wants to throw the weapon relative to the character (taking into account the velocity of the swipe). This gets rid of the cumbersome two-joystick layout, which is hard to use.

### One button and finger

(Update 7/4: This control scheme will not be included in this release. However, this will be one of the first things implemented in a future release.) For this mode, the player will control the character by dragging his finger across the screen. The character will move in the direction that the user is dragging his finger. The faster the player moves his finger across the screen the faster the character will move in that direction. To jump the user will quickly move his finger upward. For throwing weapons, there will be a button that the player presses or holds with his thumb to arm the weapon. He then can swipe this finger in the direction that he wants to throw that particular weapon (taking into account the velocity of the swipe). There will also be an action and inventory button described up above in the joystick control scheme.

## 5.3 Audio

Ninja Dojo will not rely on audio for game. It is not necessary for a good gaming experience. (Update 7/4: We still want audio but it will not be in this release.)

[St.John]: You can use "out of the box" royalty free audio for the game. Good sites to check out are:

- <http://www.freesound.org/index.php>
- <http://www.bytesizesound.com/index.html>
- Google any one of these parameters: Sound + Music + Royalty + Free

## 5.4 Music

Music will vary based on the surrounding but will be soft and mellow. The instruments used will be traditional Japanese instruments like the shakuhachi flute, koto, and shamisen. It will have minor pentatonic sound. The background music should have a stereotypical oriental sound. A good example is the C# minor pentatonic scale.

## 5.5 Sound Effects

Sound effects will be minimal and consist of weapon, hit/miss, and button press sounds. For the scope of this project sounds can be minimal.

## 5.6 Help System

There will be no in game help system. The instructions should provide adequate help. If they do not, then they need to be redesigned. A help screen requires a player to sift through text to find out how to perform the action they want. This does not fit the simplistic nature of Ninja Dojo.

# Section 6 – Artificial Intelligence

## 6.1 Target AI

For the scope of the project, all targets will have static movement. That means that all movements are hard coded into the target and it will not respond to player actions or movements. Examples of different types of movement are:

standing still, back and forth (left to right), and up and down. When a target is hit, it will flash repeatedly for a couple of seconds and then fade out.

## 6.2 Trap AI

Traps will respond to player movement and actions. When a player either gets close enough to a trap or sets of a sensor (trip wire or pressure plate) the trap will fire. In the case of bamboo spikes, they will shoot out from wherever they are placed. For the automatic arrow launcher, it will either shoot at a static location or the player's location when he/she activated the trap.

## Section 7 – Technical

### 7.1 Target Hardware

The desired minimum requirements for Ninja Dojo are:

- Android 1.6 – If 1.6 is not feasible, then the minimum version will be 2.0.
- Multi-touch screen
- OpenGL ES 2.0 support

### 7.2 Development Hardware and Software

- Android SDK for the Eclipse IDE
- For testing, engineers can use either Android devices that meet the minimum requirements or an emulator.

### 7.3 Development Procedures and Standards

[St.John]: Please duplicate these procedures in the Engineering Section at the root of your Google site.

#### SVN

- All code will be stored in a repository.
- The trunk will contain the latest stable built of Ninja Dojo

- Engineers should work off branches and then integrate code into the trunk
- Engineers need to inform the rest of the team when add something or removing something from the trunk.
- All commits need to have a brief description of what was modified.
- Engineers should avoid mass commits to the trunk.
- Engineers cannot touch each other's branches without approval of the owner of the branch.

## Coding Standards

- Class Naming Convention
  - Camel case with the first letter always capitalized
  - Abbreviations should avoided unless common
- Variable Naming Convention
  - Camel case with the first letter not capitalized
  - Abbreviations should avoided unless common
  - If the variable is used as reference to something outside the glass have the variable start with the letter 'm'
- Enum Naming Convention
  - Snake case
  - All caps
- Constants Naming Convention
  - Snake case
  - All caps

## 7.4 Game Engine

The game engine used for Ninja Dojo will be developed in house specifically for Ninja Dojo.

## 7.5 Scripting Language

XML will be used to parse the Android Manifest, permissions list, layouts, variables, and constants.

## Section 8 – Game Art

## 8.1 Concept Art

## 8.2 Style Guides

The general look and feel the game should look a like a very well polished Gameboy Advance 2D side-scroller. The style will be very anime like with clean crisp edges and lively colors. There won't be a lot of texture detail, but just enough so that the player can recognize certain textures and patters like wood grain. A good example of this is the Naruto Saikyo Ninja Daikesshu game series for Nintendo DS.

## 8.3 Characters

Characters will have a similar art style to the overworld character sprites Camelot uses in their Gameboy Advance RPGs. (e.g. Golden Sun and Mario Tennis). They will be a lot cleaner and use fewer colors. They will still feature a larger than normal head and big eyes. Characters should look a little childish, cute look while still remaining clean and crisp. The main character will resemble a 2-D version of the main character from I-Ninja, but look cuter and less serious.

## 8.4 Environments

Backgrounds and background elements will also follow the same childish, cute, clean, crisp look. There will be various background and foreground layers to recreate a parallax effect for the side-scrolling engine. The Naruto Saikyo Ninja Daikesshu game series for Nintendo DS is a good example of the desired look and feel for the background.

## 8.5 Equipment

The main characters equipment should be design such that it blends well with the character sprites. When the main character is holding a weapon, it should noticeable, but not so much that it distracts from the overall image. A good example of this is the weapon sprites used in the Naruto Saikyo Ninja Daikesshu game series for Nintendo DS.

## 8.6 Miscellaneous

## Section 9 – Management

## 9.1 Detailed Schedule

This schedule is subject to change.

Task	Due Date
Game engine design <ul style="list-style-type: none"> <li>▪ Controls</li> <li>▪ Graphics</li> <li>▪ Physics</li> </ul>	6/5
Characters – Concept Art	6/5
Background – Concept Art	6/5
Controls Engine	7/3
Physics Engine	6/19
Graphics Engine	7/10
Player <ul style="list-style-type: none"> <li>▪ Animation</li> <li>▪ Movement</li> <li>▪ Weapons</li> </ul>	7/17
Target/Trap <ul style="list-style-type: none"> <li>▪ Animation</li> <li>▪ Movement</li> </ul>	7/17
Sprites <ul style="list-style-type: none"> <li>▪ Background</li> <li>▪ Background Elements</li> <li>▪ Player</li> <li>▪ Weapons</li> <li>▪ Targets</li> </ul>	6/19

<ul style="list-style-type: none"> <li>▪ Traps</li> <li>▪ HUD</li> <li>▪ Menu</li> </ul>	
Level Design	6/26
Level Implementation	7/10
UI Elements <ul style="list-style-type: none"> <li>▪ Menu</li> <li>▪ HUD</li> </ul>	7/24

[St.John]:excellent schedule overview. At our next meeting we will review your detailed schedule in the Dev tracker.

## 9.2 Budget

- Assembla repository - \$20/per month
- Visual and audio assets - \$100 total
- Android Market registration - \$25
- Engineers may choose to invest in a development device (e.g. Android smartphone or tablet), which will not be apart of the budget.

## 9.3 Risk Analysis

There is little to no risk involved in this project. The repository is relatively cheap and there are very little expenditures. In the worst-case scenario, at the end of the summer we will end up with an incomplete game. If this happens, team members can continues to stay on board during the semester and put in 5-10 hours per week during their free time.

## 9.4 Localization Plan

[St.John]: You are presenting an asset risk mitigation plan below. Localization refers to how you deploy a product in territories other than the United States. Specifically, translation and customs consideration. I spoke to this issue above when I mentioned text strings. Let's talk more about this at our next meeting.

## Communication

All official communication will take place in the form of emails or telephone/video conferencing. Instant messaging can be used but team members need to log it down in an email thread. During a telephone/video conference call, one person will be assigned to take notes, which will be archived and available to the team.

## Code

All code will be stored in the repository. Code should not be shared through email. Code can be backed up on an external device or on a local drive as long as Rhys is notified and the person storing the code takes full responsibility if the code is stolen or copied without authorization.

## Documents

All documents will be stored on Google Docs and shared with the whole team.

## Visual Assets

Visual assets such as sprites and concept art will be stored on the Ninja Dojo Google Site, uploaded and shared as a Google Doc, and stored in the repository in appropriate folder. Originals can remain with the author, but must be sent to Rhys when the author is no longer a part of the team.

## Audio Assets

Audio assets such as background music and sound effects will be stored on the Ninja Dojo Google Site and in the repository in appropriate folder. Originals can remain with the author, but must be sent to Rhys when the author is no longer a part of the team.

## Design Docs, Guides, References, and other information

Any design docs, guides, reference manuals, or other form of information related to the development of Ninja Dojo will be stored on the Ninja Dojo Google Site. All team members will have access to these files.

## 9.5 Test Plan

All code that is written needs to be unit tested. Engineers should hold code review and keep design documents up to date. For complex pieces of code, multiple engineers should unit test it and compare results. For the actual game, engineers will install and run the game on Android devices that meet the minimum requirements. An Android emulator should be avoided because it does not provide an accurate emulation of user experience. All bugs will be logged into a Google doc and tagged with: a timestamp, description of the bug, description on how to recreate it, threat level, and completion timestamp and version.

[St.John]:Do not log bugs using a Google doc!!! Use the ticket System in Assembla... it is vastly superior to tracking in a doc. Let's talk more about this at our next meeting.

Great job on the GDD!!! :)

### Comments

You do not have permission to add comments.

---

[Sign in](#) | [Recent Site Activity](#) | [Report Abuse](#) | [Print Page](#) | Powered By [Google Sites](#)