|  |  |
| --- | --- |
|  | **2008** |
|  | PM  Suzanne Boylan  David Foster  Dev  Jerome Byrne  Darragh Coy |

|  |
| --- |
| [Ninja – Revison 6] |
|  |

Contents

[1- Game concept 4](#_Toc201029399)

[1.1 - Game outline 4](#_Toc201029400)

[1.2 - Setting 4](#_Toc201029401)

[1.3 - Story 4](#_Toc201029402)

[2 - Game mechanics 5](#_Toc201029403)

[2.1 - Character abilities 5](#_Toc201029404)

[2.2 - Camera 6](#_Toc201029405)

[2.3 - World objects 6](#_Toc201029406)

[2.4 - Character weapons 7](#_Toc201029407)

[2.5 - Enemies 7](#_Toc201029408)

[2.6 - Power-ups 8](#_Toc201029409)

[2.7 – Difficulty level 8](#_Toc201029410)

[2.8 - Goals & rewards 9](#_Toc201029411)

[2.9 - Game duration & level structure 9](#_Toc201029412)

[3 - Target market & audience 10](#_Toc201029413)

[3.1 - Target audience 10](#_Toc201029414)

[3.2 - Age rating 10](#_Toc201029415)

[3.3 - Market research 10](#_Toc201029416)

[4 - Artwork & style 11](#_Toc201029417)

[4.1 - Art direction & colour schema 11](#_Toc201029418)

[4.2 - Art pipeline & techniques 11](#_Toc201029419)

[4.3 - Audio direction 12](#_Toc201029420)

[4.4 - Audio techniques 12](#_Toc201029421)

[5 - User interface & controls 13](#_Toc201029422)

[5.1 - HUD & in-game user interface 13](#_Toc201029423)

[5.2 - Menu structure 13](#_Toc201029424)

[5.3 - Control layout 13](#_Toc201029425)

[6 - Tools and technologies 14](#_Toc201029426)

[6.1 - XNA 14](#_Toc201029427)

[6.2- Adobe Photoshop 14](#_Toc201029428)

[6.3 - TV Support 14](#_Toc201029429)

[6.4 - XBOX Live Support 14](#_Toc201029430)

[6.5- Game engine 14](#_Toc201029431)

[7 - Risk analysis 15](#_Toc201029432)

[7.1 - Artwork 15](#_Toc201029433)

[7.2 - Time 15](#_Toc201029434)

[7.3 - Audio 15](#_Toc201029435)

[7.4 - Technical risks 15](#_Toc201029436)

[7.5- Level design 15](#_Toc201029437)

[7.6 - Localisation 16](#_Toc201029438)

[8 - Schedule 17](#_Toc201029439)

[9 - Project terminology 18](#_Toc201029440)

# 1- Game concept

## 1.1 - Game outline

With this project, we propose to create a 2D platform / action game in which the player takes control of a skilled ninja from feudal Japan. This Ninja uses throwing knives as his primary weapon. Throwing knives can be thrown in any direction and are used to eliminate swarms of enemies which are trying to kill our character. The player can also make use of the ninjas agile abilities such as wall running, wall jumping, rolling etc... Levels will be arena based where the player is confined to screen space (no scrolling). The player must defeat all enemies to move onto the next level where the difficulty will increase. The player makes use of walls and platforms to bounce off in the game and pickups are available such as health and bombs etc.

[Required action: Decide if scrolling or confined levels would be preferable. WILL DO AFTER PROTOTYPE]

## 1.2 - Setting

The game will be set in a variety of locations in medieval Japan, ranging from small rural villages to towns to the fortified castles of powerful feudal warlords. The game as such will reflect the style of this era and location.

## 1.3 - Story

You play a ninja whose shogun master has been ruthlessly murdered by a rival shogun. Your master had made a pact of friendship with the shogun, only to be betrayed. Now you must uphold your honour and avenge your master’s death.



Defeating the treacherous and evil shogun will form the main motivation for the player in the game.

Your quest begins in your home village which has been overrun by the shogun’s henchmen. You must first battle to clear the village of these men and gradually fight your way to the treacherous shogun’s castle- where the final confrontation will take place.

# 2 - Game mechanics

## 2.1 - Character abilities

Character description

The player’s character in the game will be a ninja in the style of medieval Japan. Not much information will be given on the background of the ninja other than the fact that he is fighting to avenge his master’s death. The character will wear traditional ninja attire and his face will be covered so as to conceal his identity.

[](http://bp1.blogger.com/_YPFDDjQ_y_Y/SBiR1baOsCI/AAAAAAAABNE/vGe0kYi6bE0/s1600-h/ninja_manga.jpg)

The ninja’s face will be covered but he will not have these knives on his hands!

Health & lives

The character will have a health bar which will be emptied as more damage is taken. This health bar will be generously large enough so that the player will not be frustrated by dying frequently; but not so large as to make the game too easy.

Once the player dies the level will restart and the player will have to try again. There will be no concept of ‘lives’ as with similar types of games.

Weapons

The character will be able to attack using weapons. Available weapons are detailed in the next sub section.

Running

The player’s character will always run when the directional stick is moved, but will gain in momentum as the player moves more and more. This momentum gain will be quick so the player will not need to spend too long building up speed to do a successful wall run.

Jumping

The ninja will be able to jump off the ground. The jump distance will be quite high given the agility of the ninja. The player will also be able to direct the ninja’s movements and fall once in mid-air.

Wall running

The player’s character will have the ability to run up walls for a short amount of time. The amount that the player run up the walls will depend how much speed has been built up from running on the ground.

Wall jumping

While the player is wall-running it will also be possible to do a wall-jump. With this move the player will quickly jump off the wall in a direction perpendicular to the surface of the wall. The player will be able to chain wall jumps together, that is jump from one wall to another but with each consecutive jump will lose power and distance.

## 2.2 - Camera

The camera in the game will always be centred on the player and will follow the player wherever s/he goes. There will be a slight lag on the camera so it moves smoothly with the player’s movement; that is when the player moves further away the camera will move quickly towards the player, and when the player is close to the camera the camera will move more slowly.

## 2.3 - World objects

Platforms

Platforms are main objects the player will stand on in the game. The game will follow platform conventions where the player can jump upwards from platform to platform and stay standing on the platform above. The player will **not** be able to cling onto the edges of platforms as in some games; we want to keep the amount of possible actions for the ninja small so the game-play remains fluid and streamlined.

Walls

Walls are nothing special, apart from the fact that the player will be able to bounce off them (see wall jumping) in order to reach new areas.

Destructible objects

There will be a variety of obstacles such as crates or weak walls that the player will be able to smash with the sword. These won’t serve much purpose except only to enhance the feeling of interactivity with the game world and make the player feel powerful.

[Required action: Confirm LIST OF DESTRUCTABLE OBJECTS]

## 2.4 - Character weapons

Sword

The player’s character will have a sword at his disposal for close range combat. The sword will be a lot more powerful than ranged attacks with the shurikens but it will also have the disadvantage that the player will be in greater danger when closer to enemies.

Shurikens

The ninja will have ninja stars or shurikens at his disposal. Shurikens are deadly throwing knives in the shape of a star that can be used to attack enemies at range. Shurikens will not be as powerful as the ninja’s main weapon (the sword) but will have the advantage that the player will be able to attack at range. The player will have an infinite amount of these shurikens at his/her disposal.



This render shows how the shuriken is structured

## 2.5 - Enemies

All the enemies in the game will be ninjas just like our character. We will reuse the sprites and animation from our character but change the colour / style of the enemy ninjas to differentiate them from the player. There will be two main types of enemy ninja

Apprentice samurai

These enemies will walk around passively and attack the player once the player gets too close. Apprentice samurai are the weakest form of enemies in the game and are easy fodder for the player shurikens.

Master samurai

Similar to the apprentice samurai, except these samurai are more proactive in attacking and pursuing the player. They move around much quicker and take more damage.

Ninjas

Ninjas have the same characteristics as the player. They can jump off ledges and fire shurikens at the player. Ninjas are generally the most difficult type of opponent in the game to hit; however they are not as physically tough as the master samurai and do not take as many hits to kill.

## 2.6 - Power-ups

The player will find a variety of different power-ups in each level that can be used to gain advantage over the enemy.

Ninja boost

This pickup activates immediately once the player walks over the power-up. The ninja-boost pickup grants the player extended sprinting and jumping capabilities, as well as the ability to attack quicker. This pickup lasts for a certain amount of time before the player returns back to normal.

Ninja shield

This rare pickup temporarily grants the player’s ninja invulnerability to damage. The effect is short lasting so the player must make good use of this ability while it is active. The power-up activates immediately once it has been collected.

Ninja blast

This pickup is collected by the player but does not immediately activate once collected. The player has control over when the ninja blast power-up will be used. The ninja-blast power-up releases a powerful blast of mystical energy around the player which disintegrates enemies on contact.

Health vial

Health vials will restore a portion of the player’s health once picked up.

## 2.7 – Difficulty level

There will be 3 available difficulty levels in the game: easy, medium and hard. On the hard difficult level enemies will move faster, be tougher and there will more of them. The opposite will be true on the easy difficulty level. Medium difficulty level will be moderately challenging but achievable by most players.

[Required action: decide if different difficulty levels will mean different amounts or placement of enemies. may require additional level design work.]

## 2.8 - Goals & rewards

The ultimate goal of the game is to defeat the enemy shogun and complete the ninja’s quest. To do this the player must battle through an increasingly challenging series of levels. Each level will be complete only once all foes on that level have been vanquished.

A scoring system will also be used as a metric to measure the player’s performance and provide an incentive for replay. This list has yet to be finalised but these elements may affect the final score of the player:

* The time taken to complete each level
* How much damage the player receives during each level
* The amount of bonus pickups found

[Required action: Confirm final SCORING SYSTEM]

High scores will be kept on a *per-level* basis rather than for the entire game. The game will keep track of the player’s best score and update it when the player has beaten his/her current best.

## 2.9 - Game duration & level structure

The game will begin with a brief introduction screen explaining the story of the game and the justification for the player’s actions. This screen will be followed by the in-game sections (levels) themselves.

Levels will be short; lasting between 5 and 10 minutes. To prolong each level and make the combat more interesting, the player will have to fight several waves of enemies. Each wave will be progressively more difficult and challenging for the player. At the start of each wave the player’s health will be replenished and there will be a short pause in the action to provide brief respite for the player.

Once a level has been completed a brief message will be shown congratulating the player and the game will automatically switch to the next level or the end of game screen, depending if the last level has just been completed. The next level will also be ‘unlocked’ once the current level is finished- meaning the player can skip to the level if desired at the main menu screen when starting a new game.

We do not have an exact number of levels as of yet but 3 main levels have been confirmed. The player will first start in the ninja’s home village where s/he must fight off the invading shogun’s warriors. The next level will take place in the woods as the player travels to the castle of the evil shogun. The final level will take place in the castle of the shogun itself, and this will be the scene of the final confrontation. More levels may or may not be added due to time constraints.

[Required action: Confirm final level listing]

[Required action: Confirm Number of enemy waves]

The game will end with a splash screen congratulating the player and explaining the outcome of the player’s quest.

# 3 - Target market & audience

## 3.1 - Target audience

Based on the genre and theme of the game, we think our game would appeal mainly to the following types of players:

* 15-35 year old males who enjoy fast paced action / reflex games
* 30+ veteran gamers who have played classic arcade games like *Shinobi*, Street fighter, and Golden Axe on platforms such as the SNES, Genesis and the NES.
* Any other users who enjoy downloading action / arcade games off the XBOX Live Arcade download service.

## 3.2 - Age rating

Considering the violent theme of the game; we would assume that an age rating of 15+ would be most appropriate for this game. The game will not be realistically violent and would be more cartoon like or similar to a comic-book in appearance; hence the reason why we think 15+ rather than an 18+ certification would be suitable for the game.

## 3.3 - Market research

XBLA Survey

We conducted a small survey into gamer preferences for XBOX Live Arcade and similar download services. Although the response was only moderate and the amount of data returned was small we have found that in general that action games sit very well on XBOX Live Arcade in particular.

XBLA Investigation

We did some speculative research into the titles currently present on XBOX Live Arcade. There definitely seems to be quite a lot of arcade / action type games available on the service; more so than most casual games services. This might indicate a preference for these types of games on this particular service.

XBLA Sales figures

We managed to track down some unofficial sales figures for XBOX Live Arcade. Although these are not official and the methods used to create these figures may be called into question, they do give at least some indication of what’s selling on XBLA:

<http://www.vgchartz.com/news/news.php?id=626>

Aside from card games which form the top two biggest selling games on XBLA, shooters and action games seem to get a very high standing on the service.

Casual games magazines

We have been reading casual games magazines to gain an appreciation for the casual games businesses. These magazines contain articles relating to business issues such as payment models to development issues such as scope and project management.

# 4 - Artwork & style

## 4.1 - Art direction & colour schema

The artwork will be more comic-book style in nature rather than ultra realistic. Colours will not be overly brash like some comic-book style games, but will not be muted either. We intend to go for a more western interpretation of medieval Japan rather than the manga style of drawing which is very popular in modern Japanese cartoons.



The ninja character should follow a style similar to this drawing



An illustration for what the environs might feel like.

## 4.2 - Art pipeline & techniques

We will be using Adobe Photoshop as a tool to touch-up and modify the artwork in the game. We will begin initially by either creating reference artwork or by temporarily borrowing artwork from elsewhere to give an idea of what direction we would like the artwork to go in. We hope to produce some of the easier artwork ourselves; however more difficult artwork such as the character sprites and animations may have to be outsourced elsewhere.

[Required action: identify artwork pieces required for game and either create within the project team or outsource elsewhere]

## 4.3 - Audio direction

The audio for the game will consist of sound effects for the characters, weapons and power-ups along with a backing music track. There will be no speech for the game other than simple grunts etc. for the characters; this will help keep the game locale agnostic so it can be enjoyed equally in any region. Some ambient sound effects may also be used in the background to add to the atmosphere. Music should have an oriental theme to it but would preferably be more ambient in nature and sit well in the background, so as not to intrude too much on the game itself.

[Required action: Decide if ambient sound effects are appropriate]

[Required action: Finalise a list of required sound effects and create or outsource]

## 4.4 - Audio techniques

The audio techniques used for this project will be fairly standard. Depending on the sound effect required we will either use normal voice acting, record sound effects using foley techniques, or create new sounds using synthesisers. We may also use a combination of the three techniques, depending on the sound required. Another potential option also will be to license sound effects from elsewhere.

DSP effects and other wave editing tricks will also be used in to mould the sound into its final form. All sounds should be compressed and normalised and EQ adjusted so that the optimum clarity can be achieved on the sound output device.

Audacity will be used as the main wave editor for the project and a variety of freeware DSP plug-ins will be used to process the sound effects. We have a variety of synthesisers at our disposal, but Native Instruments FM7 would probably be the best synthesiser for generating sound effects due to its powerful FM synthesis engine.

We are unsure as of yet how the music will be created or even if we will do it ourselves. We have the Fruity Loops music sequencer at our disposal and we could create the music if required. It would take time to get proper results however but it would also probably take just as long to find and license a track if we were to out-source also.

# 5 - User interface & controls

## 5.1 - HUD & in-game user interface

The heads up display will be very minimal.

* The score will placed centre top of the screen.
* The number of ‘ninja blasts’ the player has will be displayed at the bottom left of the screen.
* The player’s health will be displayed as a bar at the bottom centre of the screen.

## 5.2 - Menu structure

Main Menu

When the player boots the game they will be greeted with a screen containing the title of the game and three options:

* **Start game.**
* **Options.**
* **Quit.**

If the player selects ‘Start game’ the following menu screen will appear.

* Start game
  + **Choose level** – the player scrolls through and selects a level which has been unlocked (levels yet to be unlocked will be greyed out).
  + **Choose difficulty** – the player scrolls though easy, medium, and hard and selects their desired difficulty.
  + **Confirm** – this will start the game.
  + **Back** – this will bring the player back to the main menu.

If the player selects ‘Options’ on the main menu the following options will appear:

* Options
  + **Music volume –** the player can select how high the background music is.
  + **Sound effects –** the player can select how high the sound effects in the game are.
  + **Brightness –** the player can alter the brightness.
  + **Confirm –** Saves the players options and goes back to the main menu.
  + **Cancel –** Does NOT save the players options and goes back to the main menu.

If the player selects ‘Quit’ on the main menu then the game will exit and the player will return to the dashboard.

In-game (pause) menu

When the player has started a game they will have the option to bring up an in-game menu.

The menu structure here is similar to that of the main menu. When the player pause’s the game a screen with 3 options will appear.

* **Back to game.**
* **Options.**
* **Restart level.**
* **Back to main menu.**

Selecting ‘Back to game’ will close the pause menu and return to the game (the same can be done by pressing the pause menu again).

Selecting ‘Options’ will display options screen (same as main menu options). The difference here is that after selecting ‘confirm’ or ‘cancel’ the player will be returned to the main pause menu screen.

Selecting ‘Restart Level’ will restart the current Level (starting on stage 1 of that level). The player’s health will be replenished and any power-ups and scores will be reset to default.

Selecting ‘Back to main menu’ will quit the current game and the player will be brought back to the main menu screen.

## 5.3 - Control layout

* Move the player – push on the left analogue stick in the desired direction.
* Fire throwing knife / shuriken – Push on the right analogue stick in the desired direction.
* Jump – right trigger.
* Use ninja blast – left trigger.
* Sword strike – ‘A’ button.
* Pause menu – start button.

This button configuration may be refined over the course of the development cycle if required.

# 6 - Tools and technologies

## 6.1 - XNA

We plan on using the XNA framework to develop our game. Although we have access to the XDK here at Microsoft, neither of our two programmers have had prior experience with the XDK and the development effort / learning curve required would be much higher than developing with XNA. Given that time is at an absolute premium, RAD tools like XNA must be given preference to over traditional tools like the XDK. We intend on using XNA version 2.0 because although 3.0 is currently available, it is only a preview release at the time of writing and not a final product.

[Required action: GET AN XBOX CREATORS CLUB ACCOUNT]

## 6.2- Adobe Photoshop

We plan on using Adobe Photoshop to edit and produce all the art-assets in the game. Photoshop is almost the de-facto standard for editing images and we already have some familiarity with its usage.

[Required action: GET LICENSES FOR THIS PRODUCT]

## 6.3 - TV Support

We aim to support 4 different types of televisions / VDUs:

* Standard definition (480p) with 4:3 aspect ratio
* Standard definition (480p) with 16:9 aspect ratio
* High definition (720p) with 16:9 aspect ratio

[Required action: either acquire or locate this hardware so we can test all modes]

## 6.4 - XBOX Live Support

We had originally hoped that we could implement support for XBOX achievements into the game. However, we soon found out that the XNA framework does not support this feature because of potential for the points system to be devalued / undermined by unlicensed third parties developing for the XBOX using XNA.

We do not plan to have downloadable content or online play so the game will not support any of the XBOX Live enabled features.

## 6.5 - Game engine

We will use a custom coded game engine for this project. A game engine for such a project would be overkill and we would spend more time learning how to use it rather than actually producing the game. Also, our choice of XNA as our development platform reduces the choice we have for such a thing. There are many potential game engines available in C/C++, there are very few as of yet for XNA.

## 6.6 – Versioning system

We intend to use the SVN (Subversion) versioning system in order to track changes to the project and keep a history of our work. We have considered using Source Depot; but we are only familiar with the client side of Source Depot and not the server side; hence since we are more familiar with SVN we think it would be better to stick to this tool rather than to spend valuable time learning another tool.

We will set up a repository on one of our development machines and track changes from there. For ease of use we will also use the Tortoise SVN GUI front-end to subversion rather than interact with the versioning system from the command line.

[Required action: setup a SVN repository and create user accounts for repository]

## 6.7 – IDE

We will use Visual Studio 2005 as the IDE for this project. XNA is designed specifically to work with Visual Studio and the feature set in Visual Studio is perhaps one of the best out there.

# 7 - Risk analysis

## 7.1 - Artwork

Artwork is a big worry for this project. We have two programmers working on the game but no dedicated artists. Although some effort could be made by the programmers to produce art assets, the time it would take to produce such assets would undoubtedly be higher and the quality would not be as good as artwork from someone experienced in the field.

We hope to recruit some talent from Microsoft internally to help out but finding artists will be difficult and even more difficult will be finding artists with time to offer. We will need to address this situation shortly once it becomes clearer what assets we will be requiring for the game.

[Required action: decide what will be done with regard to this issue]

## 7.2 - Time

Time is a major concern with this project. We are looking at a development time which will be 8 weeks long at maximum- which is an extremely short time-frame indeed. We will also have to dedicate some of our time to other duties at Microsoft so we will have to manage the project very carefully if we are to overcome the time hurdle.

## 7.3 - Audio

We are moderately concerned about audio assets although not quite as concerned as we are with artwork. If need be we could always purchase some royalty free sounds and use them in our project. We do not as yet plan to have any speech and it is looking likely that will be also the case for the foreseeable future. We still are deciding what will be done about music also; although that would be less of a concern then the actual sound effects themselves.

## 7.4 - Technical risks

We do not foresee any huge risks with regard to the technical side of the project. The development of the engine itself should be relatively straightforward and there should not be too many technical hurdles to overcome.

## 7.5- Level design

Level design is a worry for us because we do not have any tools to create worlds with yet. Although it would be desirable to create our own level editor, it would not be wise given the short time span we have at our disposable. The best path would probably be to implement level files in the flexible .XML format and use that to create our levels. We could also follow a path used in previous projects we worked on and use a command-line based level editor; we already have a command parser for this purpose.

[Required action: decide what will be done with regard to this issue]

## 7.6 - Localisation

Localisation is a slight worry because XNA does not support advanced font rendering engines that could support a multi-byte characters like those found in Chinese. The font rendering in XNA is based on bitmap fonts and can only support a maximum of 256 possible characters. This limitation would be fine if we only intended to support western European / one byte character sets; however it would pose a severe problem for other more complex character sets.

We could take the route of embedding all our text into bitmap files which would then be displayed in game. Photoshop PSD files could then be localised to produce the correct text for each language.

[Required action: decide what will be done with regard to this issue]

# 8 - Schedule

To be confirmed.

[Required action: decide AND AGREE upon a schedule for the project]

# 9 - Project terminology

|  |  |
| --- | --- |
|  |  |
| EQ | Abbreviation for ‘equalizer’. An equalizer is an audio tool which allows selected frequencies in a sound to be boosted or reduced.  EQ can be used to boost base frequencies to give a sound more kick or decrease the high end frequencies to take the sharp edge off a sound. |
| 480p, 720p etc | Measures the number of lines in a display, or the sharpness / image resolution. High definition displays support better resolutions. |
| DSP | Digital sound processor. A DSP processes a stream of audio and applies some modulation to that stream, such as a reverb or chorus effect. |
| FM Synthesis | Frequency modulation synthesis. A form of synthesis what uses the amplitude of one wave to affect the pitch or frequency of another wave. A popular form of synthesis to generate sound effects with. |
| HUD | Heads up display. Part of the user interface in a game that is visible to the player throughout normal game-play. |
| PSD | Photoshop document. A layered and highly editable image file format used by the Adobe Photoshop image editor. |
| RAD | Rapid application development |
| XDK | XBOX Development kit. A proprietary development kit for the XBOX 360 console that includes C++ compilers, code libraries and documentation required to develop for the XBOX 360 platform. |
| SVN | Shorthand for the Subversion versioning system. A versioning system is a server program that tracks changes and the history of files in a project. |
| IDE | Integrated development environment. An IDE provides code editing, syntax highlighting, code completion, and build creation along with other development tools all in the one unified environment. Visual studio is one such IDE. |
| XNA | An easy to use framework for developing games on Windows and XBOX 360. |