**VR Medieval Mysteries**

**Production Plan Document**

Version 6

1 Sep 2016

No longer current – split into GDD and TDD

Written by Kathy Smart

All work copyright ©2016

Table of Contents

[Purpose of document 3](#_Toc455674935)

[Version control 3](#_Toc455674936)

[1 Rules 3](#_Toc455674937)

[2 Performance limits 4](#_Toc455674938)

[3 Optimisation 6](#_Toc455674939)

[4 Programming 6](#_Toc455674940)

[4.1 Mechanics 6](#_Toc455674941)

[4.2 Inputs and outputs 7](#_Toc455674942)

[4.2.1 VR special needs 7](#_Toc455674943)

[4.3 Menus 7](#_Toc455674944)

[4.3.2 In non-play level 8](#_Toc455674945)

[4.3.3 In gameplay 8](#_Toc455674946)

[5 Technical guidelines 9](#_Toc455674947)

[5.1 Naming conventions 9](#_Toc455674948)

[5.1.1 Audio clips 9](#_Toc455674949)

[5.1.2 Models 9](#_Toc455674950)

[5.2 Exporting 9](#_Toc455674951)

[5.3 Versioning, captioning, sourcing 9](#_Toc455674952)

[5.4 Tools to use 9](#_Toc455674953)

[6 Project scope 10](#_Toc455674954)

[6.1 Editor/Engine 10](#_Toc455674955)

[6.2 Model maker 10](#_Toc455674956)

[6.2.1 Maya and Photoshop 10](#_Toc455674957)

[6.2.2 Estimated person days 10](#_Toc455674958)

[7 Timeline 11](#_Toc455674959)

[8 Priority list 11](#_Toc455674960)

# Purpose of document

Produce a production plan that:

* assesses the known risks of the project
* identifies limitations of the developer
* includes a schedule with a timeline for development, dates for client feedback and signoff meetings
* lists assets and resources required for the prototype
* determines the tools to be used, and why they were chosen
* reflects the changes based on client feedback

To be read in conjunction with GDD.

# Version control

|  |  |  |  |
| --- | --- | --- | --- |
| Version 1 | PPD for programmers to review | 27 June 2016 | Kathy |
| Version 2 | Alpha milestone; cut skybox making, moved UI wireframe to GDD | 7 July 2016 | Kathy |
| Version 4 | Included mechanic Travel back to HQ. Deleted estimate for number of days to do the work as this was never estimated. Explained relevant benefits of Unity 5.4 build. Added Tools details. Trophies/gifts now shrink according to number of wrong guesses, and gain red dot if detective power used. | 29 July 2016 | Kathy |
| Version 5 | No more red dots. If detective power is not used, player gets gold star on plaque. | 3 August 2016 | Kathy |
| Version 6 | Refined rules – player can return to HQ in middle of mission, no knocking on shutters, interaction with objects, specify size of reduction for incorrect guesses, there are four endings, game replayability. Added UI sounds. Moved UI. Added clue mechanics. Changed movement. New menu option – turn off naming dialogue. Deleted requirement for audio clips to have small space at start. Added ProTools icon. | 1 September | Kathy |

# Rules

* This is a walking simulator game using visual and sound cues. Every visual clue has a matching sound clue
* There is explanatory dialogue in the Greeting level but after that, player initiates dialogue
* Player chooses mission in HQ and is teleported to that place, he can walk all over the map but cannot access a different mission without returning to HQ
* Player can return to HQ at any time
* Only filler missions can be chosen out of linear order
* No time limit for completing mission
* Player can walk anywhere within walls
* Player cannot enter buildings
* Player can interact with clue objects, generally they go to specified place or rotate in place
* Player can move specified objects – trophies, stones in wall, bonfire logs, flaming torch
* Player can use detective power which is a verbal hint, but then won’t get gold star for mission
* Player can repeatedly listen to same hint from Detective Power
* Player can choose to arrest or warn and game will vary accordingly (NPC reactions, reward)
* Immediate feedback for incorrect guess is handcuffs not working, derisive NPC dialogue and need to try again
* Delayed feedback for incorrect guess is trophy/gift reduced in size by 20% for every incorrect guess
* Major missions progress the time of day which is indicated by bell ringing and skybox change
* Completed missions result in trophy (for arrest) or gift (for letting off with warning) which player finds displayed in HQ
* Player can rearrange trophies and gifts whenever in HQ after second mission
* Mysteries all contain part of larger mystery, solved at the very end
* End game results in one of four victorious displays depending on player decisions
* Game can be replayed for different mystery investigations – each mystery will have same complainant, same suspects, but random perpetrator
* Game can be replayed to obtain perfect line of trophies or gifts
* Game can be replayed to obtain entire story by warning all the incorrect suspects before warning perpetrator

# Performance limits

* VR set up - see VR set up on Design board in Trello
* Build and launch - see Build and launch on Design board in Trello
* camera
	+ VR - any camera with no render texture is automatically rendered in stereo to the GearVR, and positional and head tracking is automatically applied to the app camera, overriding the camera’s transform
	+ put Main Camera inside CameraContainer
	+ any camera control must set camera position and orientation to the camera parent
	+ ensure add VR EyeRaycaster to Main Camera
	+ no camera zoom
	+ avoid using Euler angles whenever possible; quaternions are preferable. Try looking straight up and straight down to test your camera; it should always be stable and consistent with your head orientation
	+ can see virtual body but only where head would crane naturally
	+ can see wherever head goes so colliders must stop PC going through walls
	+ OVR Camera Controller moves at Start to calculated eye level, which may be wrong
	+ Gear VR 1440 split 1280x1440, DK2 1080 split 960x1080
	+ add VR Input script to use GearVR buttons
* movement
	+ all stops, starts and moves controlled by user
	+ short small infrequent moves
	+ walking pace only - constant speed - 1.4 metres per second
	+ no head bobbing in first person
	+ move in direction looking
	+ no diagonal oscillations and off-axis vertical rotations, no strafing, back-stepping or spinning
	+ never decouple user and camera movements
	+ no slopes, stairs, lifts
	+ teleport with visual cues and original orientation
	+ no moving horizon line or landmarks
	+ always able to look around, even in menus, game pause and cut scenes
	+ no clipping
	+ make raycasts ignore all trigger colliders by Edit > Project Settings > Physics [Manager] > Uncheck "Queries Hit Triggers"
* time
	+ Player Settings - Time - TimeManager - Fixed Timestep 0.166666, Maximum Allowed Timestep 0.166666, leave TimeScale as 1
* editor
	+ game View preview does not apply lens distortion but Oculus gameplay does
	+ static batching is not static in the editor
* lighting
	+ see Lighting on Design board in Trello
* all gameplay to be guided by sound as well as visual cues
* Gear VR optimisation limits due to reduced GPU and CPU capacity of phone and need to have wide range of vision at 90 fps - see Art for GearVR on Art board in Trello

# Optimisation

* less than 50 draw calls per frame
* fixed clock level API, try for 0,0
* don't reduce clock rate
	+ If the GPU times stay under 12 ms or so, you can probably reduce your GPU clock level. If the GPU times are low, but the frame rate isn’t 60 FPS, you are CPU limited
	+ Use the Oculus SDK’s predictive tracking, making sure you feed in an accurate time parameter into the function call. The predictive tracking value varies based on application latency and must be tuned per application.
* don’t use chromatic aberration correction on TimeWarp
* MSAA - 2 or 4
* reduce the eye target resolution
* occlusion culling. http://docs.unity3d.com/Manual/OcclusionCulling.html
* render geometry front to back (Unity does Forward Render automatically, don't change it to Deferred)
* Unity apps should always use the multithreaded renderer option, since two cores running at 1 GHz do work more efficiently than one core running at 2 GHz
* dynamic batching - get Unity to batch all of the visible geometry using the same texture up into a single VBO and send it to the GPU all at once
* transcendental mathematical functions (pow, exp, log, cos, sin, tan, etc) are quite expensive
* while precision (float vs half vs fixed) of floating point variables is largely ignored on desktop GPUs, it is quite important to get good performance on mobile GPUs.
* always distribute release build [optimised] rather than debug build
* Android scripting http://developer.android.com/guide/index.html

# Programming

## Mechanics

* Follow how-to-use-Gear-VR tutorial
* PC movement and collision
* Crime scenes on map show availability by sound and colour
* Teleport to crime scene
* NPCs wake when PC approaches
* PC faces NPC to begin and continue conversation
* Move around crime scene
* Identify clues, examine clues
* Speak to complainant/suspects
* Use detective power
* Arrest or warn
	+ Handcuffs and warning book go from idle to awake to interactive then act
	+ Perpetrator goes to handcuffed pose
	+ Handcuffs disappear from PC and appear on perpetrator
	+ Gift presented by thanker in handcuffed pose
* Skybox changes and bells indicate time passing
* Trophy or gift appears in HQ after completed mission, has smaller size if first guess(es) incorrect
* Trophies/gifts can be rearranged
* Plaque in HQ changes after each main mission
* Plaque in HQ has places for a gold humming star for every mission which are enabled if detective power not used in that mission. If entire game played without detective powers, humming gold stars twinkle and sing
* Unique clue mechanics for every mission
	+ 1. Rotate poker to view pining cat hairs
	+ 2. Lift ripping sound cloth beside suspect’s torn clothing
	+ 3. Player dialogue identifies order of objects on ground in market (top, second top…)
	+ 4. Lift decorated pin beside mark on suspect’s clothing
	+ 5. Five different trail sets with 48 different symbols each with unique SFX and identifying dialogue. Idle, awake, interactive, identified as part of trail.
	+ 6. Player can lift and drop bonfire logs, move torch and light fire which moves and grows
	+ A. Footprints trail each footprint idle to awake if in same trail, turns off if next one not same trail
	+ B. Suspects change place in line by moving to left when spoken to
	+ C. Lift writing implement beside graffit to compare sounds
	+ D. Player lifts stones which stick in place if move to right place. When wall built, player can get pumpkins to lift beside wall to compare size of worms/pumpkin shapes

## Inputs and outputs

* Oculus main menu using large button short and long press
	+ pause game -stretch call
	+ back to main menu - stretch call
* Oculus main menu using tap button and swipe pad
	+ start game - stretch call
* point face at ground to move
* point face at zenith to use detective power
* point face at interactible objects to start them rotating
* point face at NPCs to make them interactible to
	+ have dialogue
	+ warn them
	+ arrest them
* phone volume control must be done manually before game or by pausing game and taking off headset
* sound
	+ games should be played using headphones because the output source follows the user’s head movements when they wear headphones, but not when they use speakers
	+ add Oculus Audio SDK spatialization component to Unity's native audio object objects
	+ Native Spatializer for Unity: Set void SetParameter(ref AudioSource source) function within ONSPAudioSoure.cs to public. When instantiating a prefab with this component, please be sure to call this function (and pass a reference of the AudioSource component) before calling AudioSource.Play().
	+ spatially correct audio source for every sound
		- no speech over central audio channel or LandR channels. Spatialize audio to NPC position. Place audio source/listener in centre of player's head
	+ no pinpoint sounds - sounds must attenuate over distance
	+ no stereo ambient loops (traffic, birds etc) as will not match head tracking
	+ multiple mono loops for moving object like wind or fly or bullet
	+ very few main loops
	+ wideband sounds with high frequencies
	+ small audio cues can distract
	+ match audio room size with visual room size
	+ outdoors, turn down reflections except the floor
	+ small rooms - Oculus - high reflection values in small rooms may cause distortion due to volume overload
	+ no reverb unless stationery
	+ Edit > Project Settings > Audio Manager > Default Speaker Mode - Mono
* UI
	+ localize player with sound FX and dialogue saying what things are
	+ identify objects with sound FX and dialogue
	+ ability to turn off identifying dialogue
	+ feedback indicating extended gaze
		- footstep sounds for moving
		- idle and awake NPC sounds and red face for dialogue interaction
		- idle and awake detective power sounds
		- idle and awake handcuffs sounds
		- idle and awake warning book sounds
		- idle and awake and interactive clue sounds
		- idle and awake object sounds
	+ Pause
		- players can touch back button to pause game
		- players can long-press back button to leave game and can return to exact point and position they left
	+ Universal menu - Gear VR apps must implement Universal Menu access through integration with the Oculus Mobile SDK when long-presses are detected.

### VR special needs

Remind player to take off headphones or earphones before taking off headset.

## Menus

#### VR housekeeping

* how to use back button, long and short presses
* how to miss rest of VR housekeeping
* how to stop
* how to share
* how to use tap button (reward this)
* how to swipe (reward this)
* player is seated
* player has headphones or headset for better audio experience
* player is ready to play, game will take 5 – 15 minutes
* start game

#### Main menu

* resume
* go to options
	+ display
		- graphics [increase contrast for low-vision players]
		- resolution
	+ sound
		- master volume
		- sound FX
		- music
	+ guidance
		- controls
			* how to stop
			* how to share
			* how to tap
			* how to swipe
			* how to pause
			* how to speak
		- state of game - player's progress so far
		- hints - what still needs to be done
		- hints – turn on/off dialogue naming everything
* go to credits
* new game
* go to Oculus home
* exit

#### Start menu

* start game
* go to options
* go to Oculus home

#### Pause menu

* continue game
* go to options
* go to main menu

#### End menu

* play again
* go to main menu

### In non-play level

* splash screen
	+ start game
* credits
	+ return to main menu

### In gameplay

* greeting
	+ game start
* tutorial
* main mission
* filler mission
* end mission

# Technical guidelines

## Naming conventions

### Audio clips

Audio clips should

* start with letter not number
* have no hyphens [ - ]
* have no more than 3 underscores [ \_ ]
* end with length in number of seconds to help with programming. Generally round up to next second but where timing is more important, use fractions of second
	+ - Example **Char\_NameOfClip\_x**, where *x* is length of clip in seconds rounded up e.g. 1.5f

### Models

PC\_003

## Exporting

* Maya to Unreal - smoothing is automatic now but remember to multiply by 100
* Unreal to Android phone - use Android SDK and Audio SDK

## Versioning, captioning, sourcing

* Sourcetree as front-end for GitHub.
* gitpath = C:\Users\User\AppData\Local\Atlassian\SourceTree\git\_local\bin\git.exe

# Tools

## Equipment

* Android Samsung Note 5 phone and cable
* Gear VR headset

## Editor/Engine



Unity was chosen as it features

* native support for Photoshop and Maya to create game assets
* community made assets
* strong documentation and forum support
* support for Oculus Gear VR
* existing compatibility for multiple platforms, allowing portability without a massive rewrite for each platform
* sound files can be ogg, nearly 10 times smaller than Unreal Engine's 16 bit WAV files
* texture files can be legacy Vertex Lit, much smaller than in Unity
* uses C sharp code with which programmers are familiar

Specifically the 5.4 Unity build has

* better multithreaded rendering
* better VR performance

### Disadvantages of Unity

* bloated buggy VR code
* VR movement using clumsy reticle

Unreal Engine has large model and sound files which preclude its use so we will have to use Unity and create our own VR scripts, particularly movement.

## Version control

### GitHub via SourceTree

* Perforce is not available to students outside school.
* Other source control programs do not have the power of GitHub.
* GitHub has an excellent gitIgnore function useful for preventing the saving of bulky non-original code such as Standard Assets.
* GitHub is supported at AIE with tutorials and teacher help.



## Model maker

### Maya and Photoshop



* excellent for creation and texturing of simple shapes

## Sound

* powerful, easy to use, free digital multi-track audio editor and recorder

# Timeline

By Alpha stage

* All game mechanics and gameplay systems
* All UI and feedback systems
* AI and behaviour
* Cameras and controls
* Launching from, and returning to, front end (main menus)

Features not in by alpha should be cut.

# Priority list

PC movement

Collisions

...