

Literature Review

Under Lock and Key

Adam Hubble • P17175774

Literature Review

Introduction

In relation to my study focus of three-dimensional puzzler games for mobile platforms, I am conducting a series of literature reviews to better inform my decisions for developing my proposed game; from which, I aim to gain an understanding for what is considered similar and different within existing games across the genre. Similarities between games are often referred to as “genre conventions”, whereby, games bound by the same genre typically focus upon “mechanics and game design patterns that deliver a particular play experience” [1]. Incorporating genre conventions that can be identified across multiple games, would purpose for my game to have an establishment within the puzzle game genre and therefore provide a sense of “familiarity” [1] for players. Meanwhile, implementing techniques that are not considered generic or common, would enable my game to be considered unique and innovative; this can be achieved by identifying game design similarities and differentiating from them, examples being the “theme and game objective”. These game design components are considered “separate from genre” according to Tulia-Maria [1]. Providing a sense of differentiation could also increase the level of game immersion for players, in which, players may find “events in the story” of the game less ‘predictable’ [1]. In acknowledgement of these research objectives, the literature reviews will make attempt to provide focus upon the puzzle game genre, as my reporting and development focus. To note, there is a lack of academic-based literature available, to accommodate for this research focus entirely.

Throughout the reviews of each literature piece, I will specifically explore the various aspects of mobile puzzler games in respect of aesthetics, mechanics, narrative, user interface (UI) and level design; for which, I believe to be the fundamental game design concepts. Existing games will be explored to discuss these concepts mostly, for which, my methodology is to “find the common components in the games that are used to exemplify the genre” [2]. Moreover, I also intend to explore films to help navigate the narrative, level design and aesthetic components of my game; whereby, the proposed theme of the game aims to resemble a sinister and mystery type setting throughout. My intentions for the theme, are to encourage suspenseful and slowed gameplay to inspire players to interact with their environments cognitively; for such, techniques used within the thriller and mystery genres of film, will be explored to reinforce the theme I am intending to create. As the theme of my game focuses predominantly upon player emotion, I will further investigate the effects that such games pose, on player emotions.

Reviewing literature

As my initial source of interest, I had wanted to explore the design aspects of video games bound by the puzzle genre. As referred to previously, understanding the design aspects the ‘genre exemplifies’ [2], enables the identification of “genre conventions” [1]; for my game to be recognised as a production of the genre, it requires to adopt some of the techniques recurring in existing puzzle games. For which, the article “The study of Principles of Puzzle Game Design” [3] explores the principles of puzzle game design in focus of “graphics”, “sounds”, “interaction and feedback” as well as “storylines and gameplay”. Regarding “graphics”, the article explores the application of images within puzzle games to take into consideration of the “size of the images” [3], in relation to player restraints on the ability to see images clearly and to see other world spaces; the scaling of images is thereby considered to be a significant design concept. The mobile puzzle game series ‘The Room’ [4], attempts to visualise environments scaled to the real-world and makes use of images that are centric to puzzle resolution; the series achieves image clarity from the use of the movement mechanics of a single camera, that can be controlled to increase the focus level on image detailing within the game world.

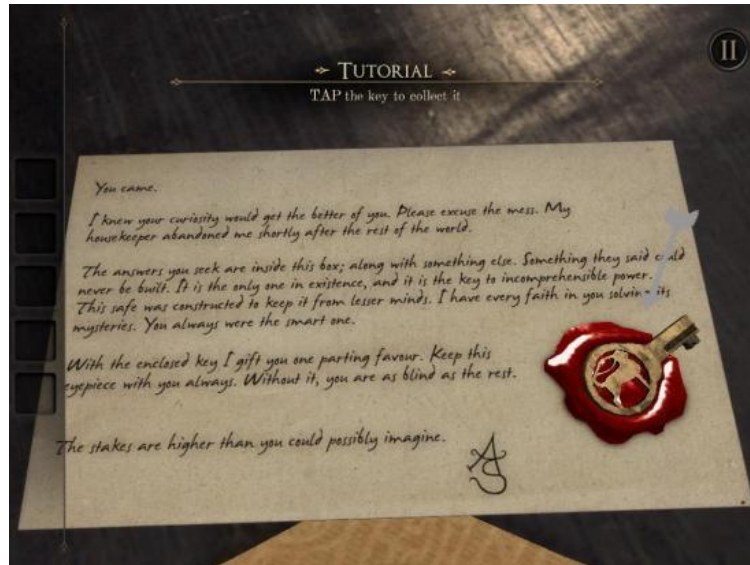


Figure 1: The Room, camera focus upon a letter for image readability [5]

Moreover, in relation to audio elements of puzzle game design, the authors describe the application of “sounds or background music” to “match the games” [3] and expands to mention the requirement of audio to be suited to all players. Within the context of my proposed game, audio will be suited to allow progression in gameplay for players in any culture or environment; multicultural agility has to be addressed, therefore linguistic audio will be excluded. This technique is also applied within The Room series [4] also, whereby the prominence of sound is exemplified in the forms of backgrounding music, as well as interactive and transitional sound effects [6]. Furthermore, in focus of player interaction and feedback, the article describes the “most important principle” to be the friendliness of the games interface [3]; this takes into consideration the clarity of the games objectives, the order of which the objectives are presented in ‘logically’, the appearance of the interface, the ease of understanding game mechanics and the possibility of players being able to determine the pace of gameplay. In essence, the authors refer to the simplification of game interaction, for a more comprehensive player experience. In relation to my proposed game, as a mobile-based game, interaction intends to be communicated through the means of buttons, as well as screen tap orientated input on game world objects; buttons will collectively form the user interface for the game. From the implementation of said game interaction, dictates a range of easy to understand game mechanics for players. Meanwhile, the order in which puzzles can be solved, intends to be sequential and portray linearity; this is to maintain player cognition and avoid deterring the player from any fabricated frustration. However, as previously mentioned, the game theme intends to control the players pace of gameplay, as opposed to players controlling the “pace of the game” [3]; in which, players will be encouraged to play slowly. Within ‘The House of Da Vinci’ series, game interactions are considered to be simple from the requirement of finger swipe or tap gestures and are enhanced by the use of recurring mechanics also [7]. An exemplifying game interaction that satisfies this claim, is the rotary mechanism of objects; which is typically controlled by circular finger swipe motions [8]. Moreover, the appearance of the game’s user interface can also be considered simple; this is due to the lack of buttons that enables it to be easily navigable.



Figure 2: *The House of Da Vinci*, showcasing the games graphical user interface [9]

Lastly, in focus of the narrative element of puzzle game design, the article presents “storylines and gameplay” to “offer information for the players”, such information is presented as the “main purpose” of puzzle games [3], as the authors describe. In which, the concept of ‘storylines’ is discussed further as the ‘creator of game situations’; which allows for “goals and challenges” to be implemented and for players to attempt and “achieve” them [3]. Meanwhile, the authors refer to “gameplay” as the process in which players ‘interact’ with the puzzle components within games and become ‘entertained’ from doing so; the article concludes the purpose of puzzle games, for the production of “challenges and interesting interactions” [3]. In further relation to both *The Room* series [4] and *The House of Da Vinci* [7] series of games, the “storylines” introduce “goals and challenges” through the occurrences of cutscenes [10]; these are typically queued, when players overcome the preceding “challenges” [3] in a single area of the game world, or are simply introduced to a new environment or story event. Cutscenes purpose for the transition between “game situations”, which allows for newer “goals and challenges” [3] to be presented to players, linearly. Meanwhile, the concept of ‘information offering’ is typically provided through the means of letters [5] throughout the game world, as well as prompting players with clues.



Figure 3: *The Room*, player prompted with a clue for the use of a found object [11]

Moreover, the “gameplay” element [3] across the games is presented through the interaction with static and obtainable objects; individually, each object represents to be a component of a puzzle, but

when the objects are interacted with in a specific sequence, players can achieve the “goal” [3], being puzzle resolution.

Concluding my findings of the article, it is inevitable that the “graphics”, “sounds”, “interaction and feedback” as well as “storylines and gameplay” [3], are significant puzzle game design aspects that need to be considered when producing my own game. It is further apparent that the intricacies of images and audio, can determine whether a player can or cannot progress through a game. As well, the order of game events poses a significance for player cognition; narrative and interaction within my game, therefore requires some form of linearity in the sequence of presenting game events. However, my findings of puzzle game design aspects are conceptual, in which, the authors discuss puzzle game design superficially; the lack of exemplifying techniques that each of the concepts employ, is a limitation of this paper, hence the need for exemplifying said concepts in existing puzzle games.

For my second source of interest, I wanted to investigate the affect that game elements have upon player emotions; as previously mentioned, suspense is the focal emotion I am wanting to engage throughout the theme of the game. For which, the article “Optimizing Player and Viewer Amusement in Suspense Video Games” explores suspense, as an emotion that can be achieved in games through signalling “the location of a threat” to players; this concerns the amount of “information provided to the audience” [12], to achieve such. In relation to suspense, the authors describe “suspenseful narratives” to typically adopt “first or third-person” camera perspectives, so that players have “the same visual and audible information as the main character” [12]; this technique can be seen within the horror game ‘Slender: The Eight Pages’ [13].



Figure 4: Slender: The Eight Pages, first-person camera perspective [14]

Moreover, the article continues to explore suspense simulation in video games, in similarity of film productions. In which, the article discusses player experience to involve aspects of “exploration”, “traps”, “persecution” and “claustrophobic environments” [12]. All of which aspects of player experience, reside within Slender[13] and both The Room [4] and The House of Da Vinci [7] game series’.



Figure 7: *The House of Da Vinci*, environment [15]



Figure 5: *The Room: Old Sins*, environment [16]



Figure 6: *Slender: The Eight Pages*, environment [17]

In further scope, a film which exemplifies all of these aspects is 'Jigsaw' [18], which presents the narrative of people imprisoned within "claustrophobic environments", to escape, they are required to 'explore' and solve a series of puzzles in the forms of "traps" [12]. Furthermore, the authors describe "suspenseful situations" to be communicated to players, via the application of "visual images, text, music, speech and environmental effects"; of which, sounds are said to be better suited to signify a "situation" before a player is able to "see it" [12]. The article continues to comment that 'changing sounds' or 'visually modifying the environment', 'are strategies typically used in suspenseful video games' [12]. In which case, many of the exemplified games [4] [7] [13] and films [18] [19] adopt low-key lit environments and dark colour palettes to heighten "frightening cues that increase the emotional response, without the need of changing the viewport" [12]; this is a recurring concept I want to implement within my production. Moreover, relating to sounds, sounds typically change in dynamic throughout films [18] [19] and games [4] [7] [13], the authors claim that "finding the right moment to provide the information" is essential for developing players "expected emotional response" [12]. Furthermore, information providance is considered conversely in respect of players becoming "stressed" when "a lack of information" is given; this infers increasing player challenge but increase in "level of suspense" also [12].

In summary of my findings, the authors have informed the ways in which suspense is orchestrated within video games and the techniques that can be used to fabricate an "emotional response" [12] from players. Additionally, the article and exemplifying sources have explored game design choices in regard to level design, narrative development and aesthetics; in which, discussions regarding "suspenseful narratives" [12], has mostly informed me with considerations for my own game. Although the articles focus is predominantly upon the horror game genre, the authors do not concern the interactive mechanics of horror games; but alternatively, the aspects of game "theme and game objective" [1], this thereby fulfils its usefulness for this investigation and poses no restraints on relevance to my own production.

References

- [1] CASVEAN, T.P. (2015) An Introduction to Videogame Genre Theory. Understanding Videogame Genre Framework. *Athens Journal of Mass Media and Communications*. [Online] Vol. 2. (Issue 1). Available from: <https://www.semanticscholar.org/paper/An-Introduction-to-Videogame-Genre-Theory.-Genre-C%24%83%C8%99vean/4dd13db214c63cd19c9d0e97df098170dd75dbc0> [Accessed 17/12/19].
- [2] BJORK, S. and HOLOPAINEN, J. and LUNDGREN, S. (2003) Game Design Patterns. In: *Digital Games Research Conference, The Netherlands, November 2003*. Rockland: Charles River Media, p. 4.

- [3] ZHOU, Z. and WU, L. (2012) The Study of Principles of Puzzle Game Design. In: *2012 International Symposium on Information Technologies in Medicine and Education, Japan, August 2012*. Hokkaido: IEEE, pp. 1-2.
- [4] Fireproof Games (2012) *The Room*. [Online] Mobile. Guildford: Fireproof Games.
- [5] GameZebo (2013) *The Room Walkthrough*. [Online] Available from: <https://www.gamezebo.com/2013/09/03/room-walkthrough-cheats-strategy-guide/> [Accessed 17/12/19].
- [6] Techzamazing (2015) *The Room Three (iOS/Android) Gameplay Walkthrough - Part 1*. [Online film] Available from: <https://www.youtube.com/watch?v=XFI93xkH75M> [Accessed 17/12/19].
- [7] Blue Brain Games (2017) *The House of Da Vinci*. [Online] Mobile. Bratislava: Blue Brain Games.
- [8] Techzamazing (2017) *The House Of Da Vinci - Walkthrough Gameplay (iOS / Android / STEAM)- PART 1*. [Online film] Available from: <https://www.youtube.com/watch?v=KAYk8F3wo5E&t=538s> [Accessed 17/12/19].
- [9] Adventure Gamers (2018) *The House of Da Vinci*. [Online] Available from: <https://adventuregamers.com/articles/view/34525> [Accessed 17/12/19].
- [10] Techzamazing (2019) *The House of Da Vinci 2 Gameplay Walkthrough (Android, iOS, Steam) - Part 1*. [Online film] Available from: <https://www.youtube.com/watch?v=YOG1pJ5okXI> [Accessed 18/12/19].
- [11] PRASAD (2012) 'The Room' for iPad game review. [Weblog] *BLOG.GSMARENA*. 7th October. Available from: <http://blog.gsmarena.com/the-room-for-ipad-game-review/> [Accessed 18/12/19].
- [12] DELATORRE, P. and LEON, C. and HIDALGO, A.S. and TAPSCOTT, A. (2019) Optimizing Player and Viewer Amusement in Suspense Video Games. *IEEE Access*. [Online] Vol. 7. Available from: <https://ieeexplore.ieee.org/document/8742555> [Accessed 18/12/19].
- [13] Parsec Productions (2012) *Slender: The Eight Pages*. [Online] Available from: <http://www.parsecproductions.net/slender/> [Accessed 18/12/19].
- [14] WYCISLIK-WILSON, M. (2017) Slender: The Eight Pages review. [Weblog] *techradar*. 27th July. Available from: <https://www.techradar.com/uk/reviews/pc-mac/software/slender-the-eight-pages-review-1326593/review> [Accessed 18/12/19].
- [15] AppUnwrapper (2017) 'The House of Da Vinci' Review: This Seems Familiar. [Weblog] *AppUnwrapper*. 25th June. Available from: <https://www.appunwrapper.com/2017/06/25/the-house-of-da-vinci-review/> [Accessed 18/12/19].
- [16] REEVES, B. (2018) The Room: Old Sins. [Weblog] *gameinformer*. 23rd January. Available from: https://www.gameinformer.com/games/the_room_old_sins/b/ios/archive/2018/01/23/the-room-old-sins-game-informer-review.aspx [Accessed 18/12/19].

[17] GAME FRONT (2012) Game Front Primer: Everything You Need for Slender: The Eight Pages. [Online] Available from: <https://www.gamefront.com/games/gamingtoday/article/game-front-primer-everything-you-need-for-slender-the-eight-pages> [Accessed 18/12/19].

[18] *Jigsaw*. (2017) [Film] Directed by MICHAEL SPIERIG and PETER SPIERIG. USA: Twisted Pictures.

[19] *Insidious: Chapter 2*. (2013) [Film] Directed by JAMES WAN. USA: Blumhouse Productions

Functional Requirements

Under Lock and Key

Adam Hubble • P17175774

Software functional requirements

Functional requirements of software applications represent the requirements of software component functionality, behaviourally. Outlining functional requirements of software is necessary for ensuring that users of the software can accomplish tasks, in use of it. In context of video games and most of all my production focus, a puzzle game, there can only be one user type, a player. The requirements of the software therefore purpose to enable a player to complete all of the set objectives throughout the game; this enables players to reach their goal. This document aims to identify the functional requirements of my production and provide insight into the way in which players can use the software.

Game overview

‘Under Lock and Key’ proposes to be a three-dimensional puzzle game, that can be viewed from the first-person and third-person perspectives. The basis of the game pursues the narrative of the playing character, who becomes captured and then imprisoned within a house-type environment; inside each room accessible to the player, features a series of interconnected puzzle components. Players are required to explore and interact with their environments, to solve all of the puzzles within every accessible room. For every room that is completed by the player, nears them to escaping the prison. Each room represents a level to note.

Basic functional requirements of the game

The basic functional requirements of the software application revolves around a player’s ability to interact with menus and initial scenes, which enables a game to be in a state of play or configuration. The following content lists these requirements.

Functional requirements

- The application will allow a user to be a player
- The application will allow a player to start a game session
- The application will allow a player to pause the state of a game session
- The application will allow a player to toggle the active state of subtitling
- The application will allow a player to exit a game session
- The application will allow a player to transition between the game’s scenes

Functional requirements of game scenes

The functional requirements of the game’s scenes, purposes to outline the ways in which scenes are designed to enable a player to interact with objects within their environments and complete objectives. For which further allows players to approach the end goal of the game: escape. The following content lists these requirements.

Functional requirements

- The scenes will allow the presence of interactable game objects
- The scenes will allow the presence of non-interactable game objects
- The scenes will allow the presence of plains to surface the game objects
- The scenes will allow the presence of a player-controlled game object
- The scenes will allow the presence of numerous player-controlled camera objects
- The scenes will allow the presence of game physics, provided by the existence of game object collider and rigid body components

- The scenes will each represent a level, excluding the menu and initial game scenes
- The scene will allow the presence of non-linguistic audio, in the forms of instrumental music and sound effects
- The scenes will allow the presence of light objects, to ensure the visibility of each scene
- The scenes will allow the presence of user interface objects, to enable the movement and interaction of player-controlled and interactable game objects
- The scenes will allow the presence of text overlaying, to linguistically represent sound effects in the form of subtitles

Functional requirements for the player of the game

The functional requirements of the game in relation to the player, determines how a player is able to control and introduce an interaction with objects within their environments, with the aims of attempting to complete the games objectives. The following content lists these requirements.

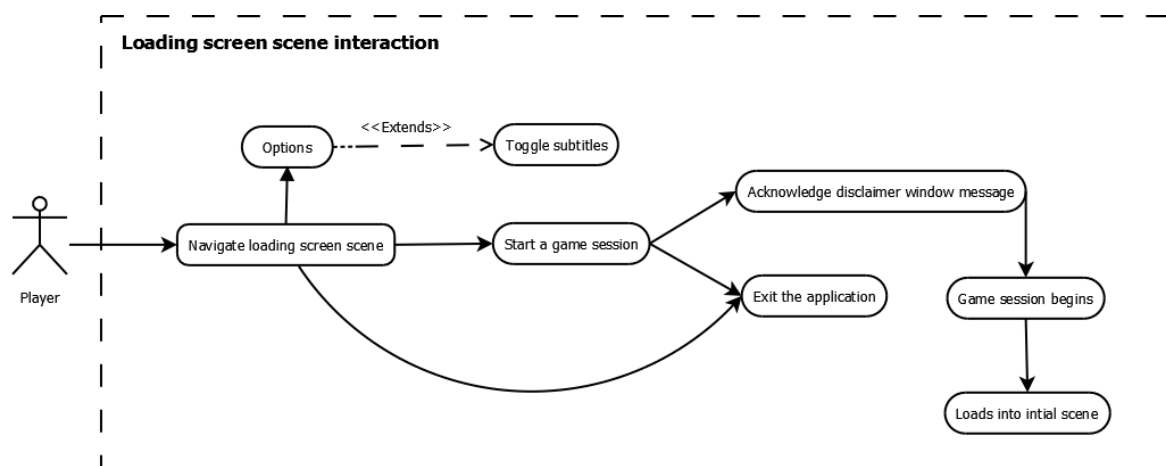
Functional requirements

- The player-controlled game object can traverse in the facing direction of the active camera object, when the corresponding user interface buttons are interacted with
- The player-controlled game objects facing direction can be adjusted, when the corresponding user interface buttons are interacted with
- The player-controlled game object can interact with interactable objects, when within the required proximity of these objects, through touch, hold and swipe interactions
- The player-controlled game object should remain upright, in the event of colliding with other game objects
- The player-controlled game object should not be able to leave the intended area of play, unless all of the events within the current level are recognised as being complete (puzzle logic)

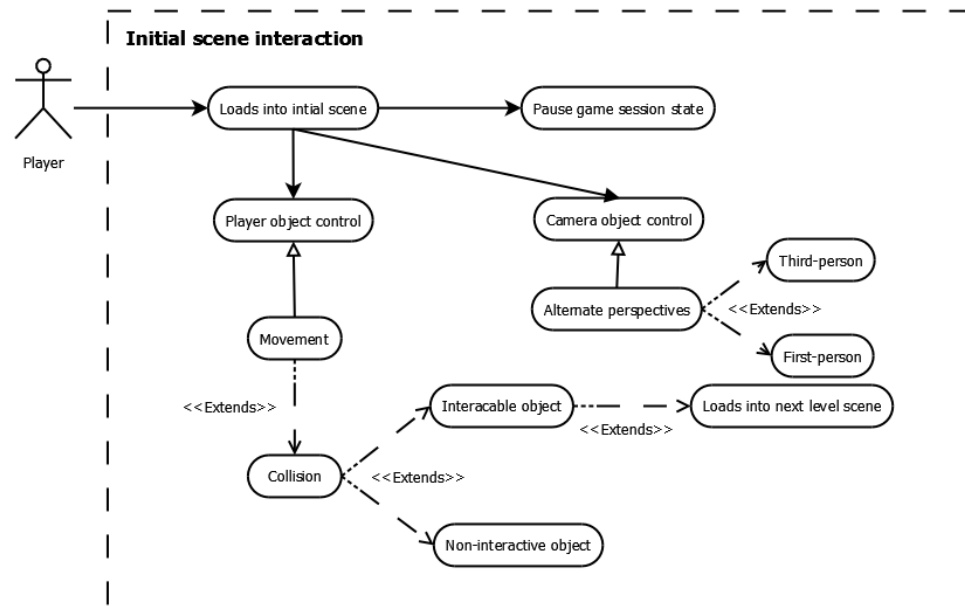
Software use cases

Software use cases exist for describing the relation between the way software applications are interacted with, by external users. The focus of software use cases is to illustrate the ways in which users can achieve particular goals and the requirements of the software to enable such. In context of my production, the following section of content will outline player relationships with my game proposal, in attempt of showcasing the software's functional requirements, as specified above.

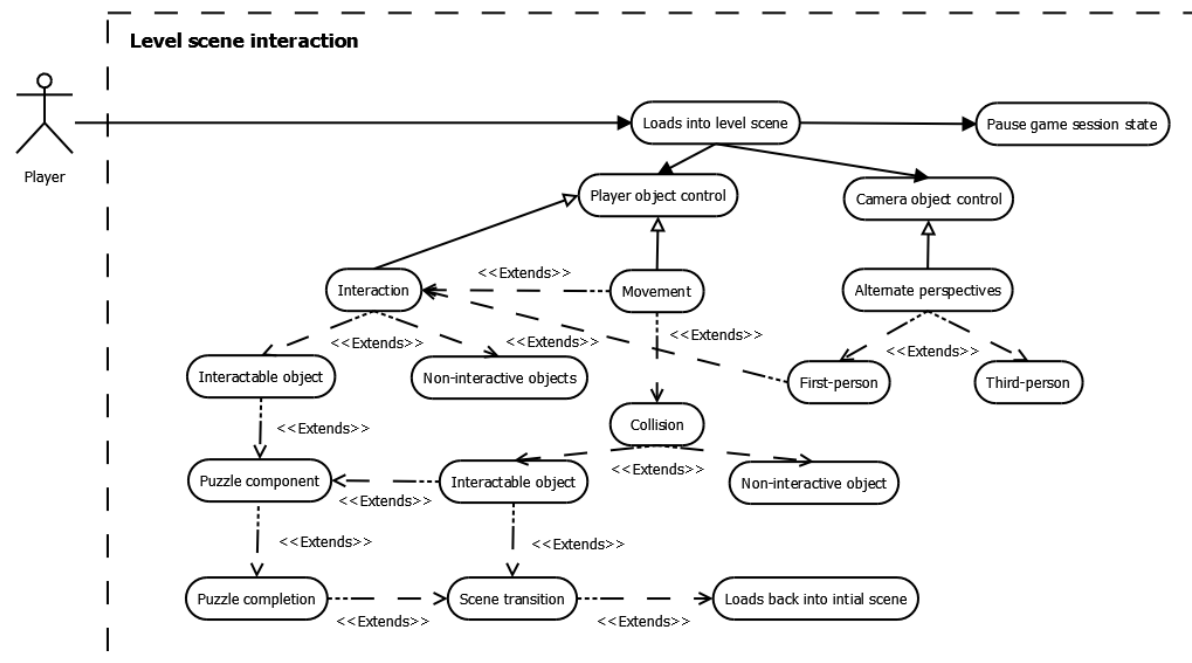
User case: loading screen scene interaction



User case: initial scene interaction



User case: level scene interaction



Indicative Test Plan

Under Lock and Key

Adam Hubble • P17175774

Methodology of testing

The purpose for testing, suits the requirement of acknowledging proper functionality within software applications. In the scope of my project proposal, a mobile based game, a series of test cases will be conducted to ensure that the software meets its technical and functional expectations. Thereby, the purpose of this document is to describe and illustrate the testing strategy of the software outlined; the document will provide indicative test cases upon the initial software setup, to provide insight into how test scenarios are orchestrated and therefore measured.

Test objectives

In relation to testing objectives, the mobile game will be tested in validation of gameplay quality, as well as application reliability, usability and performance. For which, testing will conform to unit, Blackbox and performance profiling, testing approaches; these methods are suited for testing both a software's internal and external design. As previously mentioned, the software testing will be tailored to the games technical and functional requirements.

In relation to quality testing of the software, the game will be tested in accordance to its expectations of being bug-free, to enable the best gameplay experience for players possible. The nature of these tests will predominantly be focused upon the mechanism components of the game. Meanwhile, relating to application reliability, usability and performance, the game will be tested in relevance of performance profiling; this serves to provide players with a stable and therefore enjoyable gameplay experience. This provides emphasis on the management of graphical representation and vastness of level design for the game.

Test strategy

Regarding the development of my software piece, I have chosen to adopt the Agile methodology of software development, SCRUM, to enable constant and gradual inspection and adaptation to the software, in the form of sprints. This allows for me to remain comprehensive about the application I'm developing and aims to increase the quality of my software deliverable.

In correspondence to testing, I am similarly going to adopt an Agile software testing methodology; the purpose of this methodology is to align the testing processes with the development processes of the software. In which, I propose that the game is tested compositionally, after each component is implemented within the game; this enables the code base of the game to be more robust and bug-free, as the games development becomes increasingly expansive. Moreover, in respect of future development, as defective functionality can be detected and removed gradually, the code base of the software will be enabled to be simplified and easier to interpret. I believe that this strategy of testing will be better suited for locating functionality defects and performance consumption issues, progressively; in which, the most recent implementation would indicate to be the most probable cause for error.

In relation to the testing methodology, I have selected exploratory testing as my testing basis. For which, exploratory can be used for testing the expected behaviours and functionality of software components, throughout its development. Test cases are not typically configured in advance of developing software functionality and are instead designed and executed simultaneously. This allows for the code base of software to be refactored, to enable aspects of the software to behave as expected, which can be determined from the pass rate of the test cases produced. It can also be considered beneficial for discovering component errors, that other testing techniques may ignore.

Blackbox testing

Purpose of Blackbox testing

Blackbox testing is a software testing technique which focuses on application behaviour and performance; the code base of a given application is not known and so the functionality of a given application is determined to be working correctly, incorrectly or not working from a visual basis. This is measured from differences observed in application output, from calling various inputs in the application, differences in application output are subject to the working functionality of the inputs to note.

In relation to the mobile game as a piece of software, Blackbox testing enables the identification of lacking and non-functioning game mechanics at a surface level; as there is no need to review the games code base to acknowledge the implemented functionality's working status. Being able to construct test cases quickly aids the pace of development, in which errors can be quickly acknowledged also; therefore, I have nominated this testing method for its suitability.

Blackbox testing cases

Case	Summary	Process	Actual result(s)	Expected result(s)	Passed?
1	Player object travels upwards and descends gradually when jump button is pressed.	Jump button embedded within the UI is pressed, when the player object is 'grounded'.	Player object travels upwards once jump button is pressed; the player object gradually descends due to gravity parameter set.	Player object travels upwards once jump button is pressed; the player object gradually descends due to gravity parameter set.	
2	Player object rotates around its own axis positively in the 'Y' axis when turn right button is pressed or held.	Turn right button embedded within the UI is pressed.	Player object rotates right around its own axis in the 'Y' axis at a constant rate once turn right button is pressed or held.	Player object rotates right around its own axis in the 'Y' axis at a constant rate once turn right button is pressed or held.	
3	Player object rotates around its own axis negatively in the 'Y' axis when turn left button is pressed or held.	Turn left button embedded within the UI is pressed.	Player object rotates left around its own axis in the 'Y' axis at a constant rate once turn left button is pressed or held.	Player object rotates left around its own axis in the 'Y' axis at a constant rate once turn left button is pressed or held.	
4	Player object travels forwards to its current facing direction when move forward button is pressed or held.	Move forward button embedded within the UI is pressed.	Player object travels forwards to its current facing direction at a constant rate when the forward button is pressed or held.	Player object travels forwards to its current facing direction at a constant rate when the forward button is pressed or held.	
5	Player object travels backwards from its current facing direction when move backward button is pressed or held.	Move backward button embedded within the UI is pressed.	Player object travels backwards from its current facing direction at a constant rate when the backward button is pressed or held.	Player object travels backwards from its current facing direction at a constant rate when the backward button is pressed or held.	
6	Camera perspective changes when camera cycle button is pressed.	Camera cycle button embedded within the UI is pressed.	Camera perspective changes, via disabling the current camera at the time the camera cycle button is pressed and enabling the disabled camera at the time the camera cycle button is pressed.	Camera perspective changes, via disabling the current camera at the time the camera cycle button is pressed and enabling the disabled camera at the time the camera cycle button is pressed.	
7	Player object deflects off objects, does not glitch	Player object travels towards rigid object with	Player object continually is pushed away from the object it collides with, the player	Player object continually is pushed away from the object it collides with, the player	

	through other rigid objects with colliders.	collider and collides continually.	object remains upright and on a surface; player object does not glitch through other rigid objects with colliders.	object remains upright and on a surface; player object does not glitch through other rigid objects with colliders.	
8	Disclaimer window appears with the according disclaimer text and disappears on on-click events.	Start button is pressed at the initial scene when the game is loaded. The disclaimer window button is then pressed if the disclaimer window appears.	The start button is pressed at the initial scene when the game is loaded, the disclaimer GUI window appears in the centre of the screen with the according text, when the disclaimer window (button) is pressed, the disclaimer window disappears and the proceeding scene is loaded.	The start button is pressed at the initial scene when the game is loaded, the disclaimer GUI window appears in the centre of the screen with the according text, when the disclaimer window (button) is pressed, the disclaimer window disappears and the proceeding scene is loaded.	

Unit testing

Purpose of unit testing

Opposing Blackbox testing, unit testing is a software testing technique which focuses on specific units or components within a given application; the code base of the given application is known, which is necessary to determine whether each unit within the applications functionality works as intended, in isolation. This is measured from a statistical basis, to ensure the accuracy of the tested units.

In relation to the mobile game as a piece of software, unit testing enables the identification of unit correctness or accuracy within application functionality. Therefore, unit testing is useful for determining numerical output correctness, specifically for the use of measuring forces applied to and acting on objects in the game, as well as pinpointing arithmetic and conditioning errors, which make up functionality within the application. Unit testing has been nominated as a testing method for its providance of precision for this matter.

Unit testing cases

Case	Summary	Process	Actual result(s)	Expected result(s)	Passed?
1	Determine whether the player object can only jump once, from when the corresponding button is pressed, and object is grounded	Interact with the jump button, whilst the player object is in the animation of jumping	'isGrounded' returns 'false' when the player object is mid-air and returns 'true', when the player object is in contact with the grounding plane	'isGrounded' returns 'false' when the player object is mid-air and returns 'true', when the player object is in contact with the grounding plane	
2	Determine whether the subtitle sequence is only replayable after the current sequence coroutine finishes	Interact with the subtitle sequence button, whilst the subtitle sequence is active	'morseCodeAudio' returns 'false' when the coroutine is active and returns 'true', when the coroutine has finished	'morseCodeAudio' returns 'false' when the coroutine is active and returns 'true', when the coroutine has finished	
3	Determine whether objects are only interactable with, when the first-person camera is enabled	Attempt to interact with interactable objects, while the third-person camera perspective is active	'mousePressed' returns 'false' when the third-person camera is active and returns 'true', when the first-person camera is active	'mousePressed' returns 'false' when the third-person camera is active and returns 'true', when the first-person camera is active	
4	Determine whether the disclaimer GUI window closes upon clicking on it, as acknowledgment	Interact with the start and disclaimer GUI window buttons, in the menu screen scene	'displayDisclaimer' returns 'true' when the disclaimer GUI window is initially made and returns 'false', when the disclaimer GUI window button is pressed upon	'displayDisclaimer' returns 'true' when the disclaimer GUI window is made and returns 'false', when the disclaimer GUI window button is pressed upon	

5	Determine whether one camera remains active and the other remains inactive, when the camera is toggled	Interact with the camera toggle button, embedded within the UI	'firstCamera' returns 'true' when application starts and 'false' when cameras are toggled again. 'secondCamera' returns 'false' when application starts, and 'true' when cameras are toggled again. 'toggleCamera' returns 'true' and 'false' per alternating button press	'firstCamera' returns 'true' when application starts and 'false' when cameras are toggled again. 'secondCamera' returns 'false' when application starts, and 'true' when cameras are toggled again. 'toggleCamera' returns 'true' and 'false' per alternating button press	
---	--	--	--	--	--

Performance profiling

Purpose of performance profiling

Performance profiling is a software testing technique that is particular to system performance, when a given system is executing software. Performance profiling has no intervention with an applications code base and like Blackbox testing, performance profiling can be observed visually, also. For which, performance profiling can be used to determine how systems perform in relation to responsiveness and stability, when executing software; a systems resource consumption and reliability can be measured.

In relation to the mobile game as a piece of software, performance profiling will allow the reliability of the software to be measured in systems, in the unit of frames per-second (FPS). This dictates the smoothness of the games representation and is dependent on the hardware of the executing system, the games visual representation, the number of events and objects that appear in a scene and the interactions that occur between these objects. In which, higher displaying units of FPS indicate a better system performance. I have nominated performance profiling, as a testing method to determine the playability of the game, over the duration of its development.

Performance profile test cases

Case	Summary	Process	Actual result(s)
1	First scene, start-up performance profiling	Load into the first scene of the game, capture performance using Unity profiler	Peak usage: CPU (97.5%), GPU (91.4%), RAM (1.55GB) Minimum usage: CPU (96.2%), GPU (13.4%), RAM (1.49GB) Average FPS: 145.58
2	First scene, player exploring environment performance profiling	Load into the first scene of the game and explore environment, capture performance using Unity profiler	Peak usage: CPU (98.4%), GPU (95.9%), RAM (1.82GB) Minimum usage: CPU (96.3%), GPU (17.5%), RAM (1.71GB) Average FPS: 141.70
3	First scene, switching camera perspectives performance profiling	Load into the first scene of the game and switch camera perspectives continuously, capture performance using Unity profiler	Peak usage: CPU (97.3%), GPU (85.8%), RAM (1.91GB) Minimum usage: CPU (97.2%), GPU (21.6%), RAM (1.79GB) Average FPS: 137.23
4	Loading screen scene, start-up performance profiling	Load game from menu screen, capture performance using Unity profiler	Peak usage: CPU (97.8%), GPU (29.4%), RAM (1.18GB) Minimum usage: CPU (97.0%), GPU (10.8%), RAM (1.17GB) Average FPS: 145.61

System Design

Under Lock and Key

Adam Hubble • P17175774

System overview

'Under Lock and Key' is a mobile-based game that aims to provide players with a three-dimensional outlook on solving puzzles, as a game bound by the puzzle genre. Players of the game are expected to be able to view the game from within first-person and third-person perspectives, to help solve the games puzzles; the basis of the game will require players to be able to interact with objects throughout various house-type environments, in the form of rooms. A puzzle should be represented as a room, in which, the objects that lie within each room should be components of the puzzle. To complete all of the game's puzzles, players are required to transition between each of the rooms. For which, when each of the room's objectives can be identified as complete, the player should be able to proceed and therefore complete the game.

System Design Document (SDD)

SDD overview

This SDD purposes to describe the reporting systems requirements, system architecture, format of input and user interface (UI) design, for which the format of input will address. All of which information, is necessary for navigating the development and implementation of the system, programmatically. In the context of the project, the intended audience of this document aims to be for the project manager and developer(s). Moreover, in relation to the systems requirements, there should be correspondence with the systems functional requirements.

System design

'Under Lock and Key' aims to enable players to be involved within an immersive gaming experience that adopts conventions of mystery and suspense to enhance the narrative plot of the game; for which, is based upon a prisoner with only one objective, to escape. The systems design purposes to enable the narrative of the game, where a player can interact with and resolve puzzles progressively. A single system is sufficient for this purpose.

Design objectives

- Camera object control
- Player object control
- Player interaction
- Puzzle sequencing
- Scene transitioning
- Navigable configuration menus
- Subtitling
- Disclaimer prompt
- UI addressing control and interaction
- Conventional to puzzle genre
- Stable system performance

In relation to the systems intended audience, it is necessary for users of the system to be advised about the nature of the games content, as themes of suspense are intended to be illustrated aesthetically and portrayed as sinister. The systems design makes considerations for player discretion, through implementing disclaimer-related prompts; this aims to inform all age of player, but especially youthful players, whom are considered more prone to being scared.

Design assumptions

The systems availability will exist across android mobile platforms only, for which, the system will be developed and deployed for android version 4.1 (Lollipop) and above devices; this supports the majoritive population of android user devices. Throughout the development of the game, changes and alterations to the systems functionality and arrangement are probable to occur, mostly as the result of testing and debugging. In which, development and testing processes of the system are likely to be conducted simultaneously, attempting to achieve proper functionality and robustness of the systems code base, progressively. System development will attempt to adhere to the Agile methodology SCRUM; testing, exploratory. Moreover, users of the system will be players of the game, who will be able to complete the games objectives through interacting with a mobile devices interface. Users will not require an active internet connection to use the system, therefore there will be no network considerations.

Design constraints

The systems design will present increased focus upon the implementation of game functionality, rather than performance considerations; however, significant performance issues will be accounted for throughout the games development. Thereby, there will be a trade-off between the quantity of functionality and quality of performance. In further correspondence to testing, there is a lack of devices available to analyse the systems performance effectively; an array of devices is required and so the systems performance cannot be assured across numerous devices. Due to the potential longevity of the game, the design will only consider multiple levels of the game, as the result of time restraints and lack of development and testing personnel. Also, in scope of time restraints, as the systems development is time dependant, the games aesthetic and some narrative intentions may not be delivered; this could also be anticipated by a lack of third-party resources available, to aid the games development.

System architecture

Relating to the systems architecture, I have opted for an object-orientated programming (OOP) approach for the use of classes, for which, each of said classes proposes to separate and group the systems attributes and behaviours in the form of data and functions, by association. From the application of classes, the system aims to be organised structurally and as a commonly used data structure within previous developments, the application of classes assumes a hastened rate of development. Additionally, the system proposes to adopt the principle of inheritance between classes, for the means of interacting with and overriding base class members; this exemplifies the concept of polymorphism and is useful for classes that have similar functionality. Furthermore, the system will also make use of encapsulation from the existence of class member access modifiers, these govern the accessibility of class members to external classes; access modifiers will allow classes of the system to provide protection from accidental member modification, from other classes residing in the system.

In the demonstration of the proposed systems architecture, I have constructed a Unified Model Language (UML) diagram for the purpose of representing the systems structural design. In which, the diagram aims to acknowledge the relationship between the systems classes and to identify each of the classes data and functional members. From the existence of the UML class diagram, the development of the system can be navigated to achieve the functional requirements of the game and can therefore fasten the development process. However, as previously mentioned the arrangement of the systems structure and its accompanying functionality are subject to change; therefore, the diagram presented below should be interpreted as the preliminary system design.

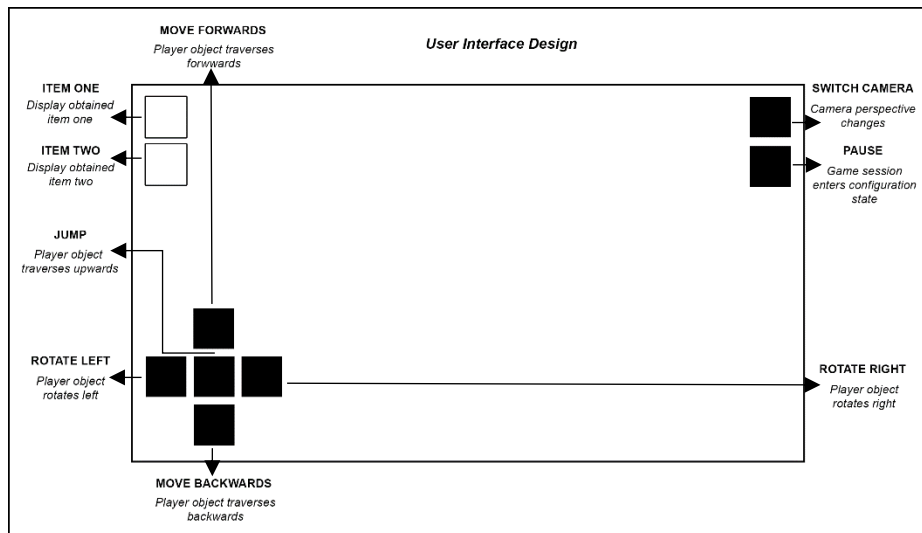


Figure 1: In-game UI design, when the game session is in a playable state

Moreover, a player should be made able to pause the game sessions state and adjust the games configuration when within said state, as detailed within the game's functional requirements:

- The application will allow a player to pause the state of a game session

Pausing the state of a game session can also be addressed via the application of a button, when pressed. Furthermore, the button can also be positioned near to the alternating camera perspective button, which attempts to enable the interface to be compact and easily navigable. Unlike player-controlled movement, alternating camera perspectives and entering player configuration menus are assumed to be interacted with less, which insinuates that this design choice is suitable as players will be encouraged to press these buttons consciously; thus, avoiding accidental interaction. As expected from entering the paused state of the game session, the system should facilitate the following:

- The application will allow a player to toggle the active state of subtitling
- The application will allow a player to exit a game session

In the following image, the intended functionality of the pause menu interface can be conceptually visualised and addressed.

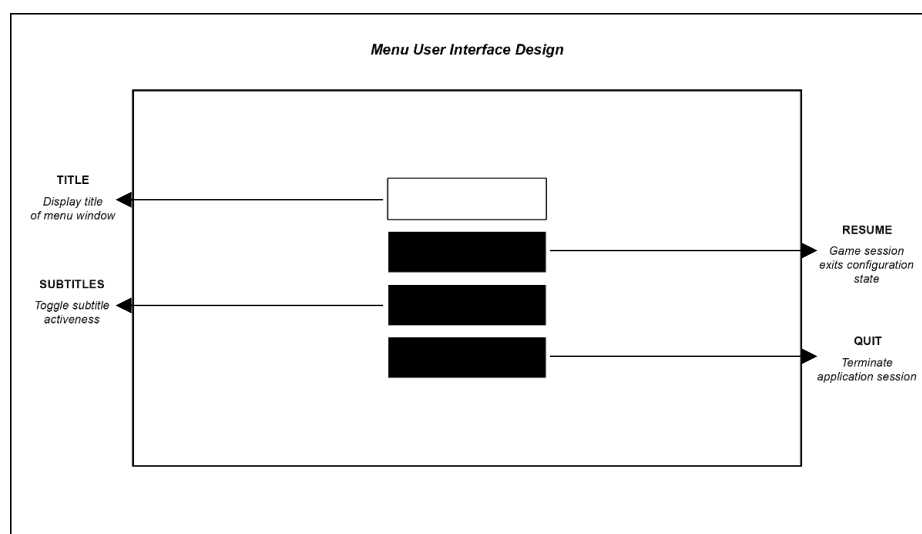


Figure 2: Menu UI design, when the game session enters the configuration state

In addition to fulfilling the functional requirements of the proposed system, the in-game UI design also aims to enable players to identify items they have obtained, from their interactions in each puzzle room scene. In accordance to the game's objectives, items in the form of game objects are proposed to be obtained by a player and used in conjunction with other objects when interacted with; from a player being able to acknowledge that it has possession of items, aims to navigate a player to its next objective in the sequence. In which, the way an item appears visually attempts to familiarise players with a related game object in the surrounding environment, therefore, the next objective or puzzle component becomes more obvious to the player. Summarily, said technique will enable gradual gameplay progression and will further try to prevent player frustration from arising.

Player acknowledgement regarding item possession can be addressed via the application of images corresponding to the game objects a player obtains. Such images should be placed and positioned away from all of buttons previously specified but should reside within the edging perimeter of a user's device. This design choice aims to prevent accidental interaction with buttons and to maintain the compactness and comprehensive state of the interface, also.

Meanwhile, in relation to the interface of the loading screen scene, a player should be able to start a game session as specified within the functional requirements:

- The application will allow a player to start a game session

As seen within all of the previously illustrated interfaces, the ability to start a game session will also be addressed via the implementation of a button, when pressed. However, unlike the in-game UI design, there is no requirement for specific button positioning as an active game session would not exist. In which, the arrangement chosen for the loading screen UI imposes to be unique in comparison to the other UI designs, whereby it is considered aesthetically driven. In correlation to the pause menu UI design, the loading screen UI design also addresses the following functionality:

- The application will allow a player to toggle the active state of subtitling
- The application will allow a player to exit a game session

In consideration of the interface's functionality, the diagram below attempts to exemplify the visual and functional aspects of the interface.

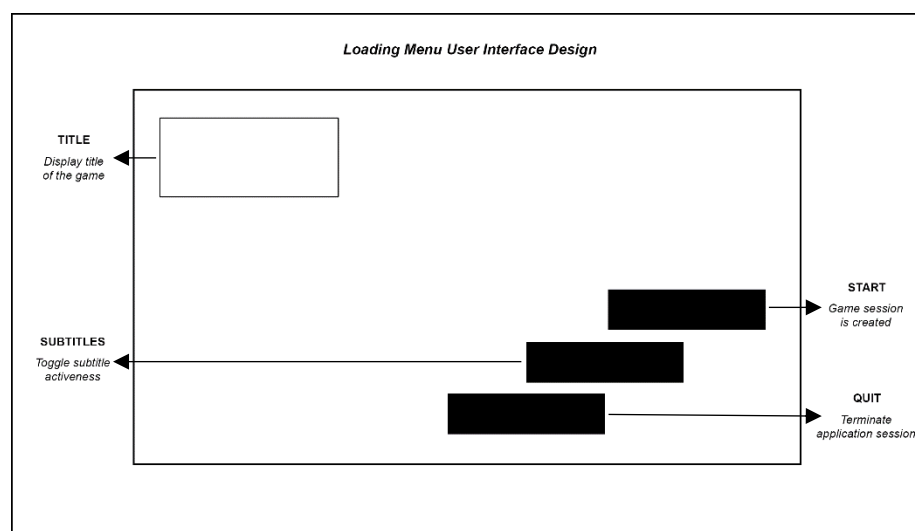


Figure 3: Loading screen UI design, when the application initially loads into memory

Implementation

Under Lock and Key

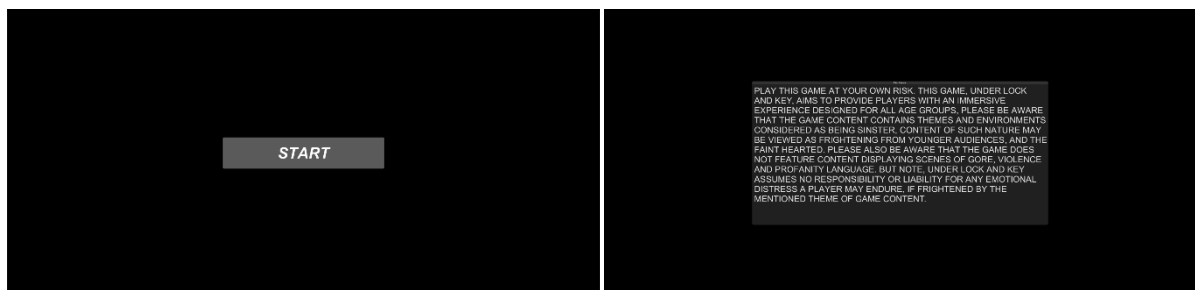
Adam Hubble • P17175774

Software implementation

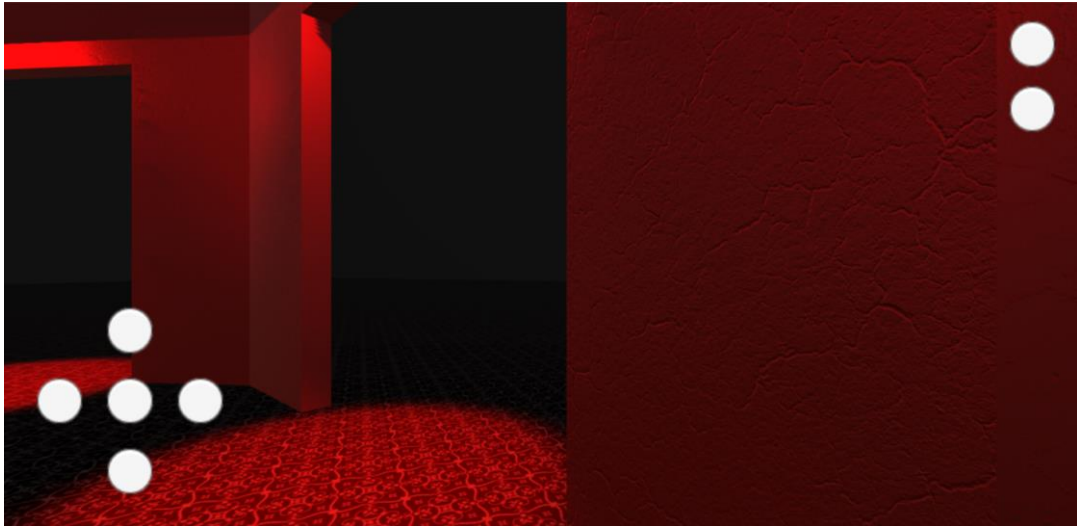
Regarding the current state of development of my puzzle game, there are currently two scenes with implementation: the loading screen scene and the initial base scene. The loading screen scene enables player to start the game, whereas the initial base scene will be used to enable players to alternate between each level scene.



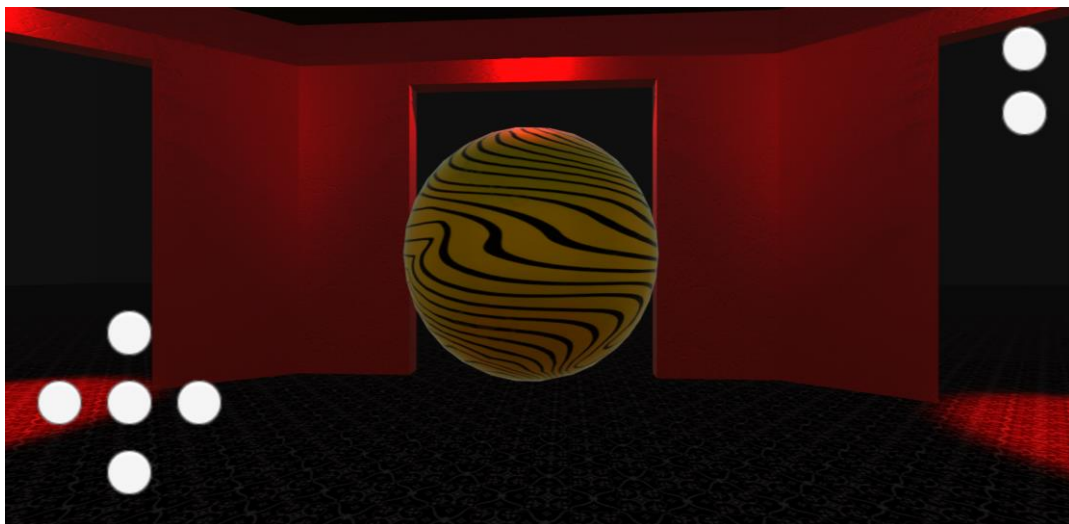
Relating to the loading screen scene, players are able to start the game from button interaction which initiates a disclaimer GUI window to appear; this has been implemented to advise players about the following graphic aspect of the games content. The initial base scene can be loaded when the disclaimer GUI window button is pressed, in acknowledgement of the message.



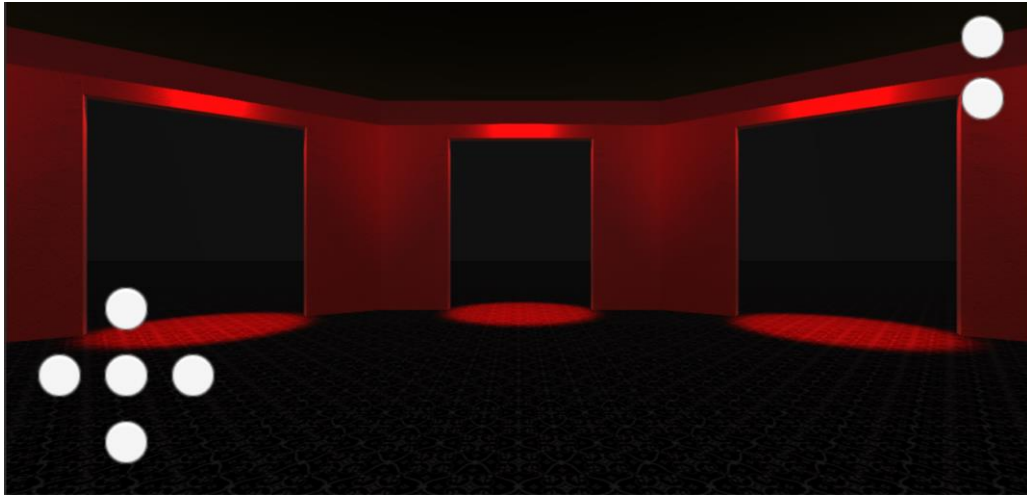
The basic visual structure of the initial base scene is present, where a series of game objects, lights, textures and UV maps have been applied to provide this.



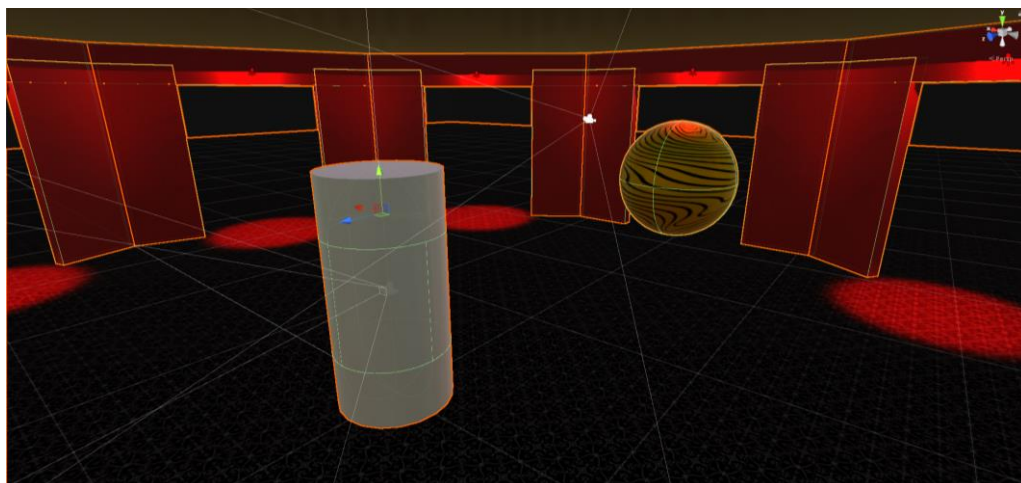
Also, to introduce the concept of interaction, a sphere game object has been placed with the corresponding script applied, to demonstrate a working functionality of rotary mechanics. For all interaction, I have implemented code that enables computer and mobile platform input, this is to test the game within preliminary builds.



Moreover, a basis for the graphical user interface (GUI) is present, which enables the player to move around in the environment, jump around the environment, switch camera perspectives and queue a subtitle sequence; the subtitle sequence has been implemented as a testing mechanic, for use within the future development of the game.



In relation to the game objects within the initial base scene, the player object has been implemented as a cylinder with the associating collider to best suit collisions across all of the scenes, existing and future. Environmental based game objects have also been assigned colliders, without rigid body components, to prevent unintended object movement.



Current script files:

