Design and development a first-person puzzle game based on time-travel game mechanics

Adrian Sim¹, Chen Tet Khuan², Md Baharul Islam³
School of Computing, Asia Pacific University of Technology & Innovation, 57000 Bukit Jalil, Kuala Lumpur, Malaysia
E-mail: totallagness@gmail.com, tet.khuan@apu.edu.my, baharulisam.md@apu.edu.my

Abstract—Due to the availability of advanced technology in graphics software and hardware, human-computer interaction is receiving considerable research attention in past few decades, particularly game-based interaction. To our best knowledge, very limited first-person puzzle games released in video game market in recent years. In this paper, we present a first-person puzzle game using the concept of time-travel game mechanic. Firstly, we have designed a set of 3D models and game level using Autodesk 3D Studio Max. We design four puzzles including brick wall, slide, car track, and exit the house/room. Then, a cross-platform game engine, namely Unity utilized to develop the First-Person Puzzle (FPP) game based on changing the timeline past, present, and future. We perform a set of experiments to validate our game that ensures all features are working in gameplay. The user acceptance test proves the performance of our system is better than the state-of-the-art first-person puzzle games.

Index Terms— first-person game, game mechanics, human-computer interaction, puzzle game, time travel, video game

1. INTRODUCTION

In the current mass influx of game releases from game companies surging through the market, there seems to be a lack of first-person puzzle games showing up on the market. Because of that, we are now looking into why triple A game companies have not been making any first-person puzzle games in the recent years? There were only a few that was created by indie game developers instead like The Room series available on Android and iOS, which is a puzzle-solving escape the room game. Antechamber that is a mind-bending psychological exploration game with really hard puzzles that is available on the PC; SUPERHOT by the SUPERHOT Team which is a first-person shooter game that uses time as another dimension where the world moves only when the player moves which is available on consoles and PC; and more recently Resident Evil 7 which is released by a triple A company but it only uses a few parts of game for the puzzle solving experience as the game mainly focuses on the action and fighting zombies part. Antichamber is a game made by developer Alexander Bruce and the game was published by Demruth on 1st of February, 2013. The game is a “mind-bending psychological exploration game where nothing can be taken for granted.” (Steam, 2013) as stated on the game’s webpage on Steam. It essentially is a puzzle game but with puzzles that are really hard to grasp at first but once you understand how the game works, you will have a rough idea on how to solve the next puzzle that is presented to you. For example, one of the first puzzles you see in the game is the blue and red stairs puzzle as seen in the figure below.

Figure 1: Easy Level Puzzle (Steam, 2013)
The game starts off with one of the easier puzzles to introduce the concept of the game towards the player. In the first puzzle, there’s a red and blue stairs, when you take any of the stairs, they will loop and come back to the same exact spot with the same pair of stairs. The player can either take the same route they used earlier or try the other but in the end it will always end up in the same place. Even in the first puzzle, a lot of players are not able to solve it directly by themselves. That is when the game helps the player out, there is an image that serves as a hint to the player on how to solve the puzzle and it will change into a riddle if the players interact with the image to give players an extra slight hint, if the player takes the other path instead, the game will still bring the player back to the same place but this time with another hint opposite of the previous hint. After using both pathways, the figure must figure out what is the actual solution here, if they attempt to keep trying the same pathways, it will still bring them back and no extra hints are given. The actual solution is to walk towards where you came from which is the back as the hints discreetly suggested the player to turn around instead.

In this game, we have a hints system that discreetly provides information to the players to give them an idea on where to go next if they are stuck in a place for too long. The game also has a time limit so the less time the player has left to complete the game, the faster the hints prompt in.

2. Related works

As mentioned in the earlier sections, the current market is lacking releases of first-person puzzle games on the mass market by triple A companies. Eric Zimmerman, a game designer and theorist, takes to task four terms that unacceptably run amok in the new media field: narrative, interactivity, play, and games (Harrigan, 2004). Suggesting that narrative should come first. Interactivity is almost the opposite of narrative; narrative flows under the direction of the author, while interactivity depends on the player for motive power (Adams, 1999). There is a direct, immediate conflict between the demands of a story and the demands of a game. Divergence from a story's path is likely to make for a less satisfying story; restricting a player's freedom of action is likely to make for a less satisfying game (Costikyan, 2003). Outside academic theory, people are usually excellent at making distinctions between narrative, drama and games. If I throw a ball at you, we do not expect you to drop it and wait until it starts telling stories (Eskelinen, 2001). In addition, another interesting finding, by adding a storyline, violent acts in the video game are justified as opposed to how the old violent games were just trying to be violent. Adding narrative changes identification, presence, emotion and motivations (Schneider, 2004). Besides that, emotions form a vital component of our experiences with interactive entertainment, motivating the cognitive decisions made during gameplay (Nacke, 2009). Hence, interactivity and narrative are opposing things and that it is hard for a developer to just lean towards one side, they will only be drawing out the players that lean towards the side that they chose. Instead, a sweet spot, balance point, should be discovered by the developer instead. Picking puzzle games as my choice of genre was mainly because they also allow the players to think while they play, which makes their brain work. Puzzle games are so similar to those of education (Linehan, et al., 2014). When the player completes the puzzle, they get satisfaction from completing them because they had to spend brainpower and time to figure out how to complete the puzzle, similar to presenting a challenge to a person. Good games are challenging, encourage players to spend large amounts of time on a specific task, and allow players to learn through exploration and failure (Gee, 2003). We also need to consider pacing in the game to make it accessible for most players. A lot of players do not want to be thrown into the fray immediately, they want a build up to the story, start off with a simple and quick tutorial to understand how the game works, introduce game mechanics bit by bit and put them into use in the game, giving them time to understand them. A lot of popular games like Portal or Braid (which are the really popular puzzle games in the past 10 years) uses a very structure way of teaching players how to use skills available in the game (Linehan, et al., 2014). Other than just the narrative and the game mechanics part, my research also allowed me to create a first-person puzzle game to put it into practical use for tests. For the slow-paced and easygoing game that I am creating for the research, I did some reading on aim assist as that is present in many first-person games, as they tend to help a lot especially when the player is using a game controller as their input. Aiming is a critical component of FPS games, and aiming speed and accuracy is often a main differentiator between experts and novices. To improve novice targeting ability, aim assistance could be used. Aim assistance improves the accuracy and speed of target acquisition by manipulating factors such as the size of the target in motor space (Rodrigo Vicencio-Moriera, 2014). Aim assist essentially pulls the player’s reticle crosshair towards the target so that they do not have to aim it at the exact spot, making it much easier for the player to complete their tasks in game. In some games, the game pulls the bullet that is fired by the player towards the enemy instead without affecting the player’s reticle crosshair, but in the game that I am going to create there will be no shooting so we
will not be using this concept. Aim assist is generally frowned upon by players in multiplayer games but are much more desirable in single player games.

While researching on the topics too, there have been discussions about how immersion is affected by the different camera perspectives, i.e. first-person vs. third-person. After a research from Korea, the results of their study confirmed that first person POV is more immersive than its third person counterpart (Denisova & Cairns, 2015). The main effect of the experimental manipulation was due to effect on real world dissociation, challenge and cognitive involvement (Denisova & Cairns, 2015). First-person games also improve player’s cognitive flexibility, a test was performed to test for differences in reaction time between video game players and non-video game players, the results showed that video game players have better cognitive skills and episodic memory structures (Colzato, et al., 2010). Not only that, they also produce improvements in sensory, perceptual and spatial cognitive functions that are different from the expertise acquire in the game (Spence & Feng, 2010). With all these info, the plan for the game would be to go with the first-person perspective. But with the first-person perspective choice, we would also need to ensure that the game will be running at an adequate amount of frames per seconds so that immersion is not broken. It is recommended to make the game to run at least 60fps even if the resolution of the game might possibly be required to render at a lower quality. Frame rates as low as 3 fps and even 7 fps are almost un-playable as users cannot adequately target opponents (Claypool, et al., 2006). Whilst, frame resolution has little effect on user performance. Users are able to effectively target opponents even at low resolutions (Claypool, et al., 2006).

3. Game design and discussion

3.1. Tutorial Puzzle

When the game starts, the player loads into the present timeline master bedroom. The player will be locked inside the room with no way out, the door will be locked and when the player walks up to the door and try to open it, a message will pop out and show that the door is locked. The game will start off allowing the player to explore the room while having them still locked inside it. It will be used to let the players understand how the timeline switching works. A piece of paper will be placed on the table showing how the timeline switch works and a piece of paper explaining the controls.

3.2. Puzzle 1: Brick Wall Puzzle

When the player first exits the master bedroom, they are still in the future timeline. If the player continues to roam around the place, they will find a brick wall in the kitchen. The player can walk up to the wall and on closer inspection, a brick is jutting slightly further out than the other pieces and it looks like it can come off. A message will show up when the player walks into its trigger box and it lets the player know they need to find a chisel and a hammer to extract the piece of brick. The hammer can be found in the future timeline toilet. The chisel can be found in the future timeline living room. Both objects are placed on the ground and can be found with ease if the player explores around, the message to ask the player to “Press F” also shows up and notifies the player that the item can be picked up. Both items are metal based and this stops the player from changing timelines if they try to do it. The same logic that was used for the crowbar is applied here but I have since created a framework for all the trigger boxes in the game after finishing the tutorial puzzle. This is to optimize the game by not having too many scripts in the project folder. The framework is called TriggerBox.cs and it is made in a way it will check if the player is in the trigger, if the player is in it, it will display the message
that it needs to show for each individual trigger box, and when the player presses the F key on their keyboard, it will start performing the logic that is written for appropriately assigned trigger box.

3.3. Puzzle 2: Slide Puzzle

After that brick wall puzzle, the player can then enter the garage to find a sliding puzzle toy that is interact-able. The player can then “Press F” to enter the sliding puzzle. If the player has not allowed the brick wall to spawn in the present timeline, basically not placing the brick in the past, the sliding puzzle object will request the player to find somewhere to hide the key first before allowing them to enter the puzzle. When the player interacts with the sliding puzzle, they are shown a 3x3 grid with cubes that have numbers on top of them.

![Figure 3: TriggerBox Framework’s Appearance](image)

The player can move these blocks with the arrow keys, it will move the empty space around and replacing whichever the block the direction the player has chosen, e.g. pressing W(up) in the above figure will move the hole upwards and replace the position of where the number “3” block is, the number “3” block will move downwards instead. The player’s objective is to organize the blocks into ascending order from top to bottom, left to right, number 1 till 8. The above figure is how the game component looks like that is made from the script. There is a RunPuzzle script that is shown in the figure below that is for accessing and starting the sliding puzzle. A script is placed on a trigger box that reads the player’s input. When the player presses “F” to start the sliding puzzle, the screen is faded out, first person controls are disabled and the camera is switched over to the other camera that is placed a distance away from the house, as shown below in the figure. The script will also enable the slide puzzle game component from here and start the logic of setting up the puzzle from there.

The run puzzle script is only for disabling controls and swapping camera views. The slide puzzle then initializes after being called in the Run Puzzle script, the called script is to start the function of the sliding puzzle, it will move the puzzle blocks and spawn them in the position of the second camera I created in the game scene. A prefab is assigned into the “clone” variable so that we can spawn it later. A nested for loop is created for the y and x coordinates, the loop will make them increase in accordance from 0.0 to 0.1 to 0.2 then repeat to 1.0 to 1.1 and so on. A vector3 is created for use from the prefab blocks that will be spawning. The prefabs blocks will be created and set as active, a counter variable will be created and assigned onto the textmesh gameObject on top of the prefab blocks made for the puzzle. The prefab blocks that were created with different numbers are then
assigned into a 1 dimensional array (List) that was created earlier in the script. Then, we add that 1 dimensional array into our normal 2 dimensional array. After that, we will set the final puzzle block that spawns and disable it so that it would not appear in the puzzle, allowing us to navigate the slide puzzle. A short script is written that loops the x and y coordinates and the giving all the pieces new Vector3 positions so that they may swap their concurrent locations. This give the “shuffle puzzle” effect. In the update, there are code that detects the player’s input, these are used after I disabled the first person controls on the player. Pressing “R” will reset/shuffle the puzzle by calling the ResetPuzzle function that is shown in the figure below. Pressing the “F” button allows the player to exit the slide puzzle and return first person controls to the player along with switching back to the main camera. I added a “T” key for debugging purposes and instant winning.

There is a script in the game to shuffle the puzzle parts presented on the screen. It assigns the jagged array of the tiles into a temporary game object, two integers are picked from a random number between 0 and the amount of rows and columns picked for the puzzle, which we are using 3 for the game as it is the intended difficulty for the audience as for now. After that, the random numbers are picked and assigned and placed into the array, the array is then assigned back into the temporary game object. The positions are then updated at the end. A difficulty setting can be added into the game and the number of rows and columns of the puzzle can then increase which will increase the number that is selected in the random.range function. There is a script applied on the puzzle that loops the puzzle pieces and check for the “empty” puzzle piece position, it will then return true or false results. The result will then be used for checking in the next script. The same process is applied to all directions. The script will check if the player presses the “Up” key whilst also checking if the result from up check function returns true. If it’s true then it will add 1 to the y-axis which would put the empty block 1 spot above. The position is then updated again. The same process is applied onto all directions. Then, there is a function to check if the winning conditions are met, it loops through the puzzle blocks to see if they are in the right positions, I also enable back the “empty” block that was disabled earlier, assign a “Win” text on top of it and show it on screen. I then start the coroutine I wrote that allows the player to go back to the game. The debugging part for forcing the game to win immediately is also written here in the “if” section. A key will also appear next to the puzzle just to show that they have won and acquired a key.

3.4. Puzzle 3: Car Track Puzzle

![Figure 4: Car Track Puzzle in Kids Bedroom Present Timeline](image)

The third puzzle in the game is the toy car track puzzle seen in the present timeline of the kid’s bedroom. The player can see a cupboard with a golden key dangling on the edge of it but they player cannot reach it directly, even with jumping. The door in the present kids bedroom is locked and cannot be opened with any key or tool. The player can start the puzzle by picking up the toy car on the ground, it will automatically place it on the race track and start moving in a circle. If the player inspects further on the race track, they will notice there is a part of the track that has a different colour than the track itself. That is the hint to this puzzle. If the player decided to explore around in the level, they would have noticed there was another race track in the future timeline’s garage. The player could pick that race track piece up and come back to the present timeline kid’s bedroom with it. The player can then walk up to the track and it will allow the player to place the track piece onto it. This will start an animation where the car will fly upwards and onto the top of the cupboard. This will cause the key on top of the cupboard to drop onto the ground.
3.5. Puzzle 4: Exiting the House

After the player has acquired the key and exited the kid’s bedroom, the player then needs to go back to the current present garage to unlock the basement door with the basement key. Once entered, the player can then switch to the future timeline to find the exit of the game. The player can see the same spot that the player saw earlier when they were in the future timeline garage, the barricades would not break so the player can only change timelines to come over to this spot. If the player attempts to go to the past timeline the “something is blocking” message will pop out since I added a collider box there to stop the player from going there. The player then can walk up to the exit and it will show a message stating that it is the end of the game prototype and thanking the player for playing it.

3.6. Timeline Changing

In the above script, I show the logic of how the timeline change occurs. The player’s input is received and checks which number they have pressed, 1 for past, 2 for present and 3 for future. Above is just showing changing to past alone to get the idea. The game starts off in the present timeline so I made a variable that checks what timeline the player is on, assigning the corresponding numbers into that variable. Once the player presses the button to change the timeline, it will disable the controls of the player, show a fade to black animation and adds 0.02f to the y-axis position of the player so that they won’t suddenly get stuck while changing timelines, all these happen whilst I also freeze the y position on the rigidbody ensuring the player does not glitch into the ground. The freezing of the rigidbody After waiting for 3 seconds for the fade to black animation to fully occur, the script enables the corresponding timeline layers which has all the props, objects, puzzles and scripts of the specific timeline.

4. Conclusions

In this research, we researched on how very few first-person puzzle games are being released on the market right now and try to make one of my own to see what is required to create one. To ensure the quality of the game, I researched on basic game design that makes a puzzle game good, then going into the more technical part which is researching on how immersion is created and maintain which includes restrictions to the frame rate of the game while trying to balance the screen resolution, what kind of features should be implemented into the game that is present in other games too like aim assist. Finally, implementing all those theories and making the first-person puzzle game after. I found that the game requires a lot of thinking to even create in the first place and that it is not easy to create one, there are time and resource limitations which restrict the developers from creating it, technological changes that might also affect development. In the future, I would like to expand more on the game idea and make it an even better product for the public to play.
References

[1] Adams E 1999 Three Problems For Interactive Storytellers. S l Gamasutra
[2] Basile s 2016 every secret in the resident evil 7 beginning hour demo
[3] Claypool M Claypool K. & Damaa F 2006 The Effects of Frame Rate and Resolution on Users Playing First Person Shooter Games Multimedia Computing and Networking Volume 6071
[4] Colzato L S Leeuwen P J V Wildeberg W P V D & Hommel B 2010 Doom’d to switch superior cognitive flexibility in players of first person shooter games Front Psychology Issue 1
[5] Costikyan G 2003 Where Stories End and Games Begin s l s n
[6] Denisova A Cairns P 2015 First Person vs Third Person Perspective in Digital Games Do Player Preferences Affect Immersion Improving Game Experiences pp 145-148
[7] Eskelinen M 2001 The Gaming Situation Game Studies 1 Volume 1
[8] Gee J 2003 What Video Games Have to Teach Us About Learning and Literacy New York Palgrave Macmillan
[9] Harrigan N W F A P 2004 First Person New media as story, performance, and game. London The MIT Press
[10] Linehan C Et al 2014 Learning Curves: Analysing Pace and Challenge in Four Successful Puzzle Games Ireland University College Cork.
[11] Nacke L E 2009 Affective Ludology Scientific Measurement of User Experience in Interactive Entertainment s l Blekinge Institute of Technology.
[12] Rodrigo Vicencio-Moriera R L M C G S B 2014 The Effectiveness (or Lack Thereof) of Aim-Assist Techniques in First-Person Shooter Games Proceedings of the 32nd annual ACM conference on Human factors in computing systems pp 937-946.
[13] Schneider E F 2004 Death with a Story Human Communication Research p 361
[14] Sony 2016 Resident Evil 7 Teaser Demo Beginning Hour
[15] Spence I Feng J 2010 Video Games and Spatial Cognition Review of General Psychology pp 92-104
Steam 2013 Antichamber Steam 2016