HACK OS

A TERMINAL-BASED GAME BY:

Calvin Laughlin

Diego Adrian Valdez Duran

Peter Ling

Nick Stephen Hafer

Rubric:

Scope: ***MVP***

Architecture: ***Level Design + Narrative Architecture***

Documentation:

Artist Statement:

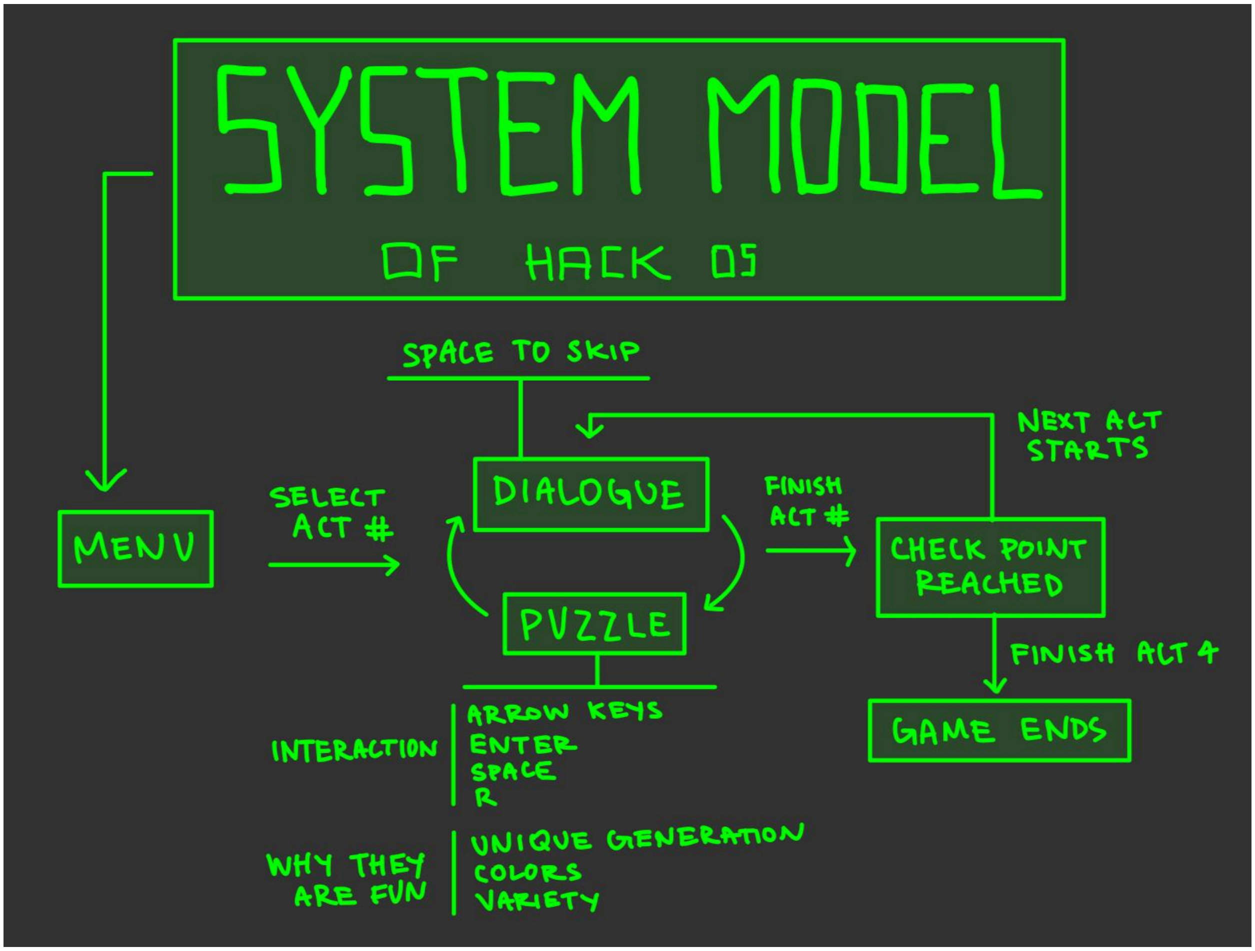
HackOS is a game we dreamt up during the brainstorming phase of CS 247G. We wanted to create a game that allowed players to *feel* like a hacker without actually needing any background knowledge in computer science or hacking. This hacker vibe was inspired by The Matrix, a movie that we had grown up on, and uses inspiration from AAA video game minigames included in the popular Fallout and GTA titles.

In order to get the desired "hacker" vibe, we intentionally developed the game in Terminal, and used techniques like a "matrix wash" from the very beginning of the game that gets a "wow" from the new players almost every time. This was a double edged sword because it meant that our puzzles had to be contained within Terminal, which was a creative constraint that helped us simplify our puzzles.

We didn't want our game to just be a mish mash of different puzzles, however, so we developed an overarching narrative with four different acts that help guide the player into each puzzle, showing them why it's necessary to the story.

Our intended audience spans both gamers and non gamers, as it requires no gaming experience to play, but those with more experience might recognize some of their favorite puzzles inspired by video games they grew up on.

Model of Chosen System:



Playtests (11 total):

5/7 Narrative Playtest:

- Who: The other team at our table + Annabelle
- Questions: Does this narrative make sense? What is simple enough yet still engaging? What's the best way to incorporate puzzles into this?
- What went well: Got lots of puzzle ideas. Assistant can help you through the game. Have levels/puzzles look very different depending on the story part. What is this puzzle accomplishing for the narrative? Add time limits to puzzles. How do we penalize players? Positive reinforcement or negative?
- What didn't go well: We should definitely start developing the narrative more at this point. The narrative should be 5 steps ahead of coding during development.

5/14 Playtest #1:

- Who: Didn't write it down
- Questions: How do the puzzles feel? What do you think of the aesthetic?
- What went well: Really like the matrix wash animation + the vibe of the game. Likes getting instructions from Roadman. Felt like I was being hacked (during a popup puzzle). Felt very immersed. Gave us more puzzle ideas.
- What didn't go well: A little confused by some puzzles. Would like more instructions on some puzzles (like fallout puzzle).

5/14 Playtest #2:

- Who: Didn't write it down
- Questions: How do the puzzles feel? What do you think of the aesthetic?
- What went well: Likes the matrix wash. Love the aesthetic. Controls are intuitive. Timer was good to apply pressure.
- What didn't go well: Text dialogue was too fast. Make it more clear that they win the puzzle (fallout game). More context for why we're doing the puzzles. Build the stage of the hacking into the narrative (like why you're looking for a word). Add a button to continue on intro text. Add identity creation at the beginning. Add checkpoints to restart puzzles. Add more dialogue between puzzles. Add more puzzles.

5/21 Playtest #1:

- Who: Parthav
- Questions: Were there any inconsistencies or disconnects between the theme/narrative and the mechanics?
- What went well: Feels stressed and under pressure in the game (that's what we were going for).
- What didn't go well: I understand things are going wrong but I don't understand the actual message. I feel like I'm not talking back to Roadman. Give more context on messages he's telling us. Through Roadman's narrative I'm stressed out, but if that's the goal then it works.

5/21 Playtest #2:

- Who: Charlotte
- Questions: Were there any inconsistencies or disconnects between the theme/narrative and the mechanics?

- What went well: Really like the hacker vibe, thought the narrative was funny. Love the visuals and different puzzles.
- What didn't go well: Not very compelled by the story. Wish there was more backstory on my relationship with Roadman. Give the player retries and have Roadman tell them to try harder. Let the player know that they've tried very hard and still failed but right now they don't know what they're doing so failure is annoying.

5/23 Playtest #1:

- Who: Didn't write their name down
- Questions: We have a more cohesive narrative now, how does it feel in tandem with more puzzles?
- What went well: Feels way better with a cohesive narrative. Very immersive.
- What didn't go well: Make the instructions clearer on each puzzle, not sure what to do for the first time. Make the slot the puzzle have 3 lives. Add more lives to puzzles so they can retry!. Add more choices for the player to do things in the game. Or maybe don't add choices, I love being along for the journey.

5/23 Playtest #2:

- Who: Didn't write their name down
- Questions: We have a more cohesive narrative now, how does it feel in tandem with more puzzles?
- What went well: Pretty straightforward. Likes the spacebar to skip. Love the different kinds of puzzles. Puzzle difficulty seemed adequate.
- What didn't go well: Explain puzzles more beforehand. If the player doesn't figure it out, throw in some dialogue for hints or guidance in puzzles. Would be cool to see answers from previous puzzles used in future puzzles. Slot machines are a little misleading because people press it early.

5/23 Playtest #3:

- Who: Didn't write their name down
- Questions: We have a more cohesive narrative now, how does it feel in tandem with more puzzles? How's the pacing feel? Should we introduce choices?
- What went well: really like the game and how we're building it. I like not having choices because it lets the narrative do the work. I like having someone else tell me what to do. Puzzle pacing felt really good. Dialogue wasn't boring, it was easy to follow along. Like that I didn't have to fail the very first puzzle, I gained confidence.

- What didn’t go well: players are on a track and don’t have agency. Be careful adding too much dialogue, avoid long paragraphs. Slow down the slot puzzle a little bit. Want the ending to be satisfying. Need something that changes that will allow players to have a moment of realization that will make the ending make sense. Introduce information about the CEOs, make the information make sense and have this come up later in the game

5/23 Playtest #4:

- Who: Didn’t write their name down
- Questions: We have a more cohesive narrative now, how does it feel in tandem with more puzzles? Do we want all the puzzles to be the same complexity or difficulty? What’s the skill curve?
- What went well: Like the vibe and like the story – felt immersive and hacker-esque
- What didn’t go well: want more feedback when you win. Balance the technical and thinking parts more. Test which puzzles are hardest for some people. To up the difficulty, can you add more easy, medium, hard and do more of the same puzzle in succession?

5/28 Playtest:

- Who: Didn’t write down
- Questions: How are the puzzles working in the game? Do people like a lot of different unique puzzles or a bunch of the same puzzles that get progressively harder?
- What went well: Love the split screen! Like the variety of puzzles. I think you should keep the variety of puzzles because its fun seeing all the different kinds.
- What didn’t go well: Would want more instructions on how to play the puzzles beforehand, and also how to replay puzzles if I fail without going back to the start.

6/4 Final Playtest:

- Who: Didn’t write down
- Questions: Final playtest, what are the things we need to fix before turning it in?
- What went well: Like the aesthetic and the concept. Likes the variety of different puzzles more than the idea of having lots of the same puzzle.
- What didn’t go well: Some puzzles are intuitive but others not as much. Have more instructions for some of the more complicated puzzles, like haystack and flow. Try to integrate these instructions into the story if possible.

Initial Decisions:

Formal Elements:

We wanted the game to be a single player game (one set of controls vs. the computer) but that someone else could watch and help figure out or try the puzzles. The overarching narrative objective is to hack into a bank and steal money with your comrades. However, the player is more or less stuck in this narrative without any choices, so the real objective is to complete the puzzles to progress to the next part of the story, and to unlock more puzzles. The player's outcome is winning each puzzle, and then also winning the entire game after they complete the whole narrative arc. The rules in this game pertain to each puzzle, so the player sort of has to re-learn the rules every time they encounter a new puzzle. We make this easier for them by building part of the puzzle into the narrative and providing instructions to each puzzle before the puzzle starts. The point of the puzzle shouldn't be figuring out how to play. The procedure of our game largely follows a cycle of narrative→puzzle→narrative so that there is a story unraveling, but also something the player gets to interact with and figure out. The bounds of our game are simply Terminal. We had a few puzzles that felt more immersive by opening in your desktop environment but they were buggy so we decided to keep it all in Terminal. We also chose to include a few characters, but not too many, because it would get confusing and take away from the story in our already short game.

Values:

We really valued the feeling and emotion that our game made the player feel. For example, we wanted the player to feel pressured and stressed during some of the puzzles. During playtesting, we recorded the players frequently saying "my palms are sweating" or "this is stressing me out because it reminds me of CS107". We also really wanted to give the players a wow factor. They would often be watching over our shoulder while we typed in the Terminal command to start the game, and then when they started playing it, it always exceeded their expectations. While our game is quite short, given the scope and time allotted this quarter, there is also a decent amount of replayability given how engaging the puzzles are. We added a "skip to Act ___" button in the menu so if there's a particular set of puzzles you want to retry, you can always go back to it.

MDA:

Mechanics:

In HackOS, the mechanics are rooted in the terminal-based interactions and set of puzzles that the player must solve. These puzzles range from word searches, ‘agility’ tests, and logic challenges, each requiring different types of input and problem-solving strategies. The game’s mechanics are designed to simulate a hacker's environment, using text commands and minimalistic visuals to create an immersive experience. This aligns with the course concept of formal elements, particularly the procedures as commands to interact with the game and the boundaries of the terminal environment. Players navigate the game mostly through arrow keys to traverse environments, spacebar to enter inputs, and other keyboard strokes specific to puzzles that are explained before entering puzzles. The single-player setup is also a key mechanic, as it emphasizes individual problem-solving and immersion, but it can also be played with a group where members can alternate taking turns solving puzzles.

Dynamics:

The dynamics of our game emerge from the interplay between the player's actions and the game's response, particularly through the narrative progression and the increasing complexity of puzzles. The game is structured by alternating between narrative segments and puzzles to create a rhythm that keeps players engaged and progresses the narrative. These interactions were inspired from mixing loops and arcs, where the repetitive nature of solving puzzles (loops) is interspersed with narrative developments (arcs). We also incorporated feedback on dynamics from playtesting, where we made puzzle instructions clearer and adjusted difficulty to make a smoother learning curve, incorporating elements of effective game onboarding principles.

Aesthetics:

The aesthetics of the game are designed to evoke the sensation and fantasy types of fun. The terminal-based interface, matrix-inspired visuals, and overall hacker theme create a strong sensory experience that aligns with the aesthetics of what is often perceived with ‘hacking’. By designing the game around this aesthetic it’s not only visually appealing but also enhances the narrative’s make-believe aspect, making players feel like they are part of an underground hacking operation. The game’s narrative and challenge further contribute to the player experience, making each puzzle not just a task but a stepping stone in our narrative.

This approach incorporates concepts of evocative spaces and embedded narratives, making the game engaging both visually and emotionally appealing as the arc unfolds. More specifically, it's evocative because of the terminal environment and limited acceptable inputs and the game follows an embedded narrative around hacking into a bank, where players follow our pre-constructed plot.

Iteration History:

*See **Playtests** section for testing history and findings.*

Big Findings:

- **Introduce puzzles more:** We had created a narrative and a set of puzzles, but we hadn't connected them at all. This feedback was really helpful because it showed that the players didn't feel like the puzzles mattered to the story at all. We decided to change the narrative a lot, and rearrange some of the puzzles temporally, then add dialogue introducing each of the puzzles and why you, the player, must complete it.
- **Add instructions:** We need to add instructions to each puzzle. In almost every playtest we did (out of more than 11 because we didn't record some of them), players said "I had a hard time figuring out what to do when I first tried the puzzle, but when I tried it a second time and I actually knew what I was doing, it was really fun". We had already introduced why the puzzle you're completing matters to the story, but we had forgotten to tell the player how to complete the puzzle. We added very simple, explicit instructions to each puzzle, so that the player isn't left guessing how to play the puzzle, and instead can play the puzzle as it's intended.
- **Shorten dialogue:** We changed the narrative a couple of times, so a few parts of our story were a bit "yappy" (AKA overly verbose), which in a short video game is quite annoying. We cut down a lot of these paragraph-long sections and added the [space-to-skip] option to move through the dialogue more quickly, or pause to read it slower.
- **Add flare to puzzles:** After getting the underlying mechanics working for a puzzle, we thought it added a lot more to the puzzle if we added colorful text or flashing bits that appear across the screen to make it seem like the computer is getting a virus as you're hacking. This continues the hacker vibe we were going for in the first place, and we believe that these small animations add a lot to convince the player that they're a hacker and immerse themselves in the bank heists.

Level Design

- How well-defined are the goals and objectives within each level? How do they relate to each other?
    - Initially we didn't have instructions for each puzzle. Realizing this was an issue, we added instructions to make it very clear what the objectives and controls were for each puzzle, so that the player wasn't left guessing what they were supposed to do. The narrative introduces each puzzle and explains why it's relevant that the player complete it.
- How does the level design offer a clear sense of progression, with challenges increasing in difficulty or complexity?
    - This is something that we asked in playtesting quite often. We wanted to know if players preferred a few kinds of puzzles that would increase in difficulty (like sticking to mazes but making them harder) or if they preferred a large variety of puzzles. We found almost unanimously that players preferred a large variety of puzzles. Because of this and the small scope of the game in this class, we had to figure out a way to organize the puzzles by difficulty. We made the first puzzle pretty easy with lots of lives to give the player confidence. After that, the player encounters an impossible puzzle that fits the narrative and why they must fail. After that, the difficulty of the puzzles is mostly in their uniqueness.
- How smoothly does the player move through the levels? Is there a logical layout that facilitates exploration and engagement?
    - The player player doesn't have a lot of agency, but it's by design. We didn't have time to scope a narrative that allowed for player choices and instead chose to focus more on developing puzzles, as that is more fun for the player. We did try to organize the puzzles as best as we could to fit the narrative, based on certain tasks that might be related to hacking into a bank.
- To what extent are game mechanics effectively utilized within the level design?
    - Each level is pretty different, but relies a lot on using the arrow keys to move around the terminal window, since each puzzle is constrained to a 2d canvas. We really enjoyed coming up with puzzles that were familiar to us and the

    - players, but showing them off in Terminal meant we had to come up with creative ways to display them (like using ASCII characters to make the maze).
- How effectively do the levels maintain player interest and engagement throughout?
    - Because of how different each puzzle is, the player never really gets bored. We also recognize that our game is quite short (<10 minutes long) and most players are willing to pay attention to a game for a few minutes regardless of how fun it is, so it’s hard to isolate this variable in playtesting. We tried really hard to make the narrative make sense with the puzzles while also being engaging because it's a little bit serious and a little bit funny. The last part of maintaining engagement/interest is how different our game is from most that are out there right now. We haven’t seen many games that take place solely in Terminal, and the players in our playtests were always “wowed” by our game and the little animations to make them feel more immersed.

Narrative Architecture

- How well does the game design support or enhance the narrative elements of the game?
    - The game design of HackOS supports and enhances narrative elements through a structured blend of puzzle and story elements. Each puzzle is integrated within the storyline, serving as a narrative device that progresses the story. So, puzzles are not merely challenges to be overcome but are meaningful parts of the story that reveal new information or present new obstacles that the player must navigate. Also, using the terminal environment reinforces the hacker theme, making the narrative feel authentic and immersive.
- To what extent does the game convey narrative elements, such as backstory, lore, or character development?
    - Hack OS conveys narrative elements such as backstory, lore, and character development through in-game dialogues and puzzles. The game immerses the players right in the middle of a narrative of hacking into a bank, guided by the character Roadman. As players progress, they uncover more about their mission, their relationships, and the stakes involved. By slowly introducing small chunks of narrative elements it keeps players engaged and curious about what happens next, making for checkpoints more meaningful. Through

    dialogue and user chat logs, we explain what is happening within the story, protagonist insights, as well as backstory into Roadman and the plot in general. These are sprinkled in within different acts of the story and by the end of the game, the player gets a more complete picture of the overarching plot and characters.

- How do gameplay and story work together to create a cohesive and immersive experience for the player?
    - In our game, gameplay and story work seamlessly together to create a cohesive and immersive experience. The narrative arcs guide the player from one puzzle to the next in a way such that each gameplay segment is contextualized within the story. More specifically, the game functions around loops and arcs, where loops are representative of puzzles and narrative as arks. For instance, a puzzle might represent a security system that the player must 'hack', which is contextualized with the narrative's goals or current state. This integration of puzzles within the story makes for a continuous flow of events in a cause-effect fashion, mitigating feelings of detachment or confusion from the narrative when solving puzzles.
- To what extent does the narrative architecture immerse the player in the game world and narrative experience?
    - We really wanted to focus the narrative architecture of HackOS in immersing the player within the hacker world. This was one of the main reasons why we chose a terminal-based interface, colorful text or flashing bits as the art style for puzzles, and a sense of urgency through the story. All of these elements draw the player deeply into the hacker experience and aesthetic. We also added to this immersion through terminal commands that would appear on the screen as events happened and other visual effects that made players feel like they were genuinely hacking or interacting with some system. In between acts, we also incorporated a brief explanation of time, place, and objectives to give players the proper context for events that followed. Having Roadman as a companion, or person that you interact with frequently, also helped to create this immersion more seamless and make the story more meaningful.
- How effectively does the architectural design evoke emotions or contribute to the atmosphere of the game?

  - The game's architectural design evokes emotions of urgency, and we intended to start with a fast-paced setting. There's a sense of pressure that's created by time-limited puzzles and a high-stakes narrative that make players feel the tension and excitement of being a hacker. But, we also didn't want to take ourselves too seriously so we added some elements of humor and lightheartedness to the dialogue to pull back from complete seriousness. Narrative moments such as receiving critical instructions from Roadman or discovering plot twists were designed specifically to elicit emotional responses. For instance, making the first act impossible to achieve emphasizes emotions of defeat but also encourages the player towards redemption. Puzzles are intended to have "aha" moments, such as figuring out one of the puzzles is similar to Wordle, induce anxiety, or have moments of triumph. The minimalist yet impactful visual and auditory elements reinforce these emotions, creating a cohesive atmosphere that keeps the player engaged and connected to the story.

Game:

- The game can be downloaded at https://github.com/calvinlaughlin/hackr and can be installed using the instructions found in the README (<1min installation).

Demonstration of Experience:

- https://youtu.be/9S335hk6Lvc