Product Backlog

Game design

Our initial, main idea was for our game to be simple yet enjoyable to play. Hence, we decided to look into important principles in Game Design to support and rationalize our choices.

When looking at the families of mechanics described by Robert Zubek in his book ‘Elements of game design’, we can see that control mechanics are how player input affects the game. According to the book, in cases where the mechanics for action games are more complex, it usually results in the player feeling greater excitement towards mastering these controls. Nonetheless, **on page 54**, it is said: *"If the game’s handling of players’ actions is clunky and inhibits their performance, that will be very perceptible to the player and it will feel wrong. Even if they cannot explain quite what is wrong with the game, players may complain that the controls feel “floaty” or “mushy” or “jittery"",* implying that it should be a priority that the controls cause intuitive responses in the game that are easy to pick up and learn rather than to make them overly complicated. Thus, we have concluded that to keep our controls simple, we will only rely on the usage of mouse clicks to be able to complete the game(s).

Moreover, we rely on uncertainty mechanics when it comes to the memory card game, which is also a relevant set of techniques in game design. **On pages 61-63**, Rubert Zuzek describes shuffling as one of the well-known applications of uncertainty mechanics. He also says: "*Nonstationary processes are even trickier. The player will be challenged to try to predict the next value, which requires figuring out how the distribution changes over time, as well as remembering previous values*.". The author talks about how uncertainty, when combined with the challenge of storing information of previous values in one's memory, adds excitement to the game for the player. In our specific memory game case, the player never knows the value of the card they are going to flip unless they have done so in previous moves, but a bit into the game, they know they can come across a card which is equal to one they have flipped beforehand in another position, which would allow them to for a pair and advance in the game.

We believe it is also important to reference progression mechanics, which are those that relate to feedback on the player's performance. Rubert Zuzek, **on page 59**, describes a subtype called indirect progression mechanics: "*Indirect progression mechanics change up the game in response to the player’s progression. Very often this can be shown via environmental changes*". Furthermore, **on page 60**: "*Whether direct or indirect, progression mechanics provide important feedback to the player on how well they are doing, and they also encourage the player to keep playing.*". Though subtle (as our intention is not to stress out the player over progress), progression mechanics is present in, for instance, the successful matches in the memory card game staying uncovered so that the player can visualize how far into the completion of the game they are. Thus, we make use of indirect progression mechanics rather than direct.

We would equally like to highlight the role of affordance in our project. The concept of affordance, according to an article by Machinations cited below, refers to the relationship between the user, objects, and possible actions in the game. It is explained that strong affordances let players instantly understand what they can do in a game without the need for tutorials. With our desktop pet, we are appealing to the user's knowledge and memory about previous game experiences and their intuition. It is to expect that, once they see the animation appear on their screen, they will conclude that they must click or hover their mouse over it for more events to unfold. This happens in the mini-games as well, because we have ensured their design is simple and intuitive so that the reader is not bothered by any additional explanations.

These have been our sources, cited in APA format:

Machinations (n.d.) Affordances in game systems design. *Machinations.* Retrieved from: Affordances in game systems design • Machinations.io

Zubek, R. (2020). *Elements of Game Design.* MIT Press. Elements of Game Design (The MIT Press) 9780262362870 - DOKUMEN.PUB

VCS

We used the version control system 'Git' and GitHub for this project to facilitate us being able to work simultaneously. This usage can be seen in the public GitHub repository linked in the README.md of the project.

Backlog:

Backlog for Set-Up Menu

Name	Description	Demo	Relevant to topic	MoSCoW
Menu	A menu for setting up the program. It should have a start button that creates the desktop pet. It closes after	Demoed by pressing the button and seeing if the desktop pet is properly created		Must

Backlog for the pet

Name	Description	Demo	Relevant to topic	MoSCoW
Pet movement	Moving and stopping the pet image across the screen randomly. This should have accompanying animations.	Demoed by watching if the image of the pet floats across the screen		Must
Pet opens game menu	When clicked the pet should open the game menu or close the menu depending on if it is currently	Demoed by clicking the pet twice and seeing if the frame directly opens and closes.	Game design: We must design the game so it is apparent that the bird is interactable. We do this by	Must

	opened. Closing the game menu or game only minimizes it and should not destroy game progress.		showing only the bird. The player will assume that it must be interactable, seeing it is the only thing on the screen.	

Backlog for the games:

Backlog for game menu

Name	Description	Demo	Relevant to topic	MoSCoW
Button that redirects to games	Buttons correlating to games that, when pressed on, redirect to the correlating game. There should also be a button that closes the program.	Demoed by pressing the button and seeing if it redirects to the correct game.		Must
Button that tells pet to sit in the corner	When pressed, the pet should move to a corner and not move from there until this button is pressed again	Demoed by pressing the button and observing if the pet moves to a corner and stays there	Game design	Could

Backlog for Memory

Name	Description	Demo	Relevant to topic	MoSCoW
Card design and pair setup	Create cohesive designs for the cards' fronts and backs, ensuring they all looked the same upside down. These designs must be shared in pairs. Then each pair of cards is assigned a value.	Confirm that a same designed is shared by exactly two pairs	Game design (visual cohesion, recognizability of pairs)	Must
Random layout	The cards are shuffled into a randomized layout at the start of each game	Replay several times to confirm the layout has changed from a game to another	VCS (different randomizations can be tested/versioned to tailor them to the game)	Must

Card interaction logic	By clicking an upside-down card, it turns around, revealing its value. After having chosen two, the program evaluates whether they match. If so, they remain uncovered. Otherwise, they flip again.	Play the game to observe the program's behavior in these scenarios	Game design (defines the main functioning of the game with a clear dynamic of cause and effect)	Must
Timings	At the start of the game, all the cards' positions are displayed for three seconds. After an incorrect match, both cards are displayed for a second to then flip back again as previously described.	Start the game and measure the time intervals described	Game design (ensures proper pacing for a fair difficulty)	Must
Animations	Appearance of a small congratulations or animation as a form of congratulation after a correct match is performed by the user	Complete pairs to check whether feedback appears		Could

Backlog for Tic-Tac-Toe

Name	Description	Demo	Relevant to topic	MoSCoW
Setup	The game must be played in a 3x3 grid where all cells are clickable. Design X's and O's simply cohesively	The grid itself can be visually checked. Then, hovering the mouse above each cell, it can be determined whether these also function as buttons	Game design (provides the game with the familiar, intuitive look of Tic Tac Toe)	Must
Decision algorithm	The computer must follow some criteria to determine its next move, as this should not be random but an attempt to avoid the user's win.	This can be tested in different board states to check whether the program follows any instructions	Game design	Must
Turns between computer and user	There must be a turn system where the player selects a position and then the computer does, alternating in a way that allows for the	This can be checked by seeing whether the computer places its own element after the player does, allowing the		Must

	correct playing of the game	player to place theirs again after that, in a turn-taking system		
Winning and draw conditions	The computer stops the game after detecting a victorious position (same element is present in an entire row, column, or diagonally) or after detecting that all the cells in the grid have been filled	Play several times provoking winning and draw scenarios to see whether the computer stops the game successfully		Must
Animations	A "You won!" or "It's a draw" shows up depending on how the game ends	Observing after winning or having a draw	Game design (signals a clear end to the game, congratulates)	Should