

Travis Moore

GAT261: Section A — Spring 2017

Professor: Richard Rowan

Menu Project Research & Project Proposal

Game Concept Research

Splice

Summary

Splice is a puzzle game that focuses on rearranging microbial cells in order to form the correct shape for each cell structure. The PC version of the game is primarily played with the mouse, and as such the menu is navigated primarily with the mouse. *Splice* is not only a fun game, but it is the first game that I played that caught my attention for how beautiful and fun-to-use a menu can be. Aesthetically, the game's menu is minimal in style, with a focus on getting the player quickly into the game, however the look and feel of the menu easily matches the atmosphere and feeling of the game itself.

Key Functions

- *Splice* successfully makes menu navigation feel both fluid and fun even when exploring menus by using interesting visuals (film grain / color aberration) and movement tied to the location of the cursor's position on the screen.



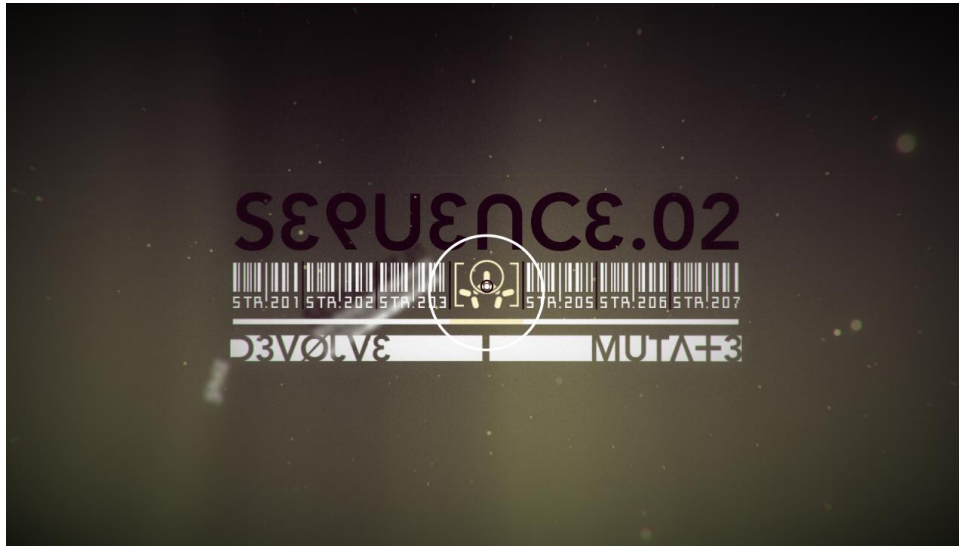
When the user moves the cursor around the screen, the main camera tilts and rotates around a central focal object (in this case, the logo of the game featured on the start screen).

- The cursor animates around selectable UI objects, sticks to a selectable UI objects, and plays a reverb sound effect that enables the user to know what is and is not interactable.



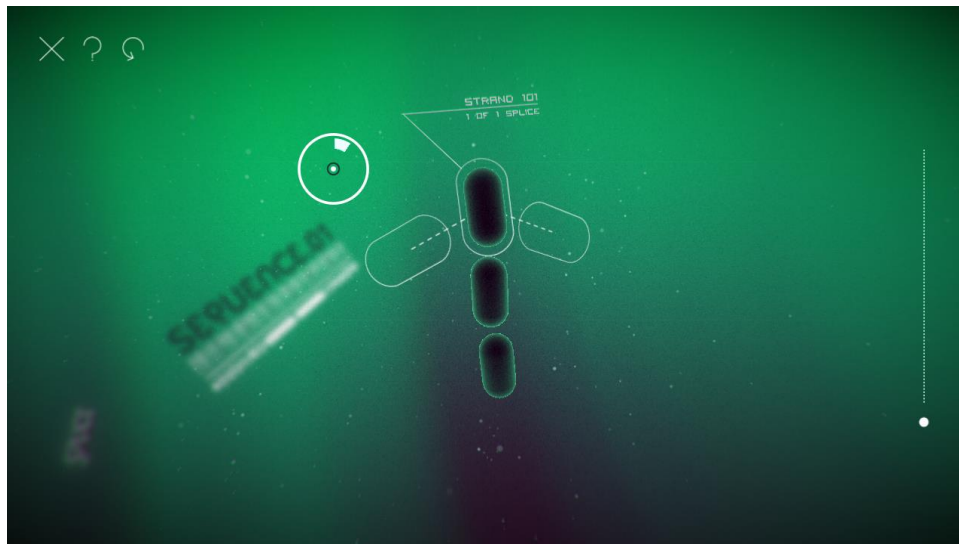
When the cursor moves down to the UI button "D3VOLVE", the cursor selection expands to encompass the selection, along with slightly shifting the color from pure white to a slight green.

- As menu elements are selected the camera zooms outward in a spiral-like pattern leaving the previous menu options blurred into the background. This feeling of zooming forward not only feels exciting, but acts as a sense of progression throughout the *Splice*.



Behind the foreground SEQUENCE.02 menu is the previous SEQUENCE.01 menu while further down the spiral is the start screen menu.

- While in-game the UI remains consistent with white being reserved for use specifically for UI elements. This deliberate design decision helps create consistency for the user while in-game and also navigating the menus in *Splice*.



The UI throughout Splice uses white exclusively for UI. Even the cursor displays how many moves the user has remaining radially around the cursor itself.

Key Takeaways

- Use visual effects that mimic the look/feel of your game, whether it is the actual look of the

game or something that matches the theme of the game, like film grain or color aberration.

- Cursor movement can be tied to the position/rotation of the main camera to give the menu a dynamic feel and make it fun to simply “move around” in the menu.
- Consider using very subtle movement in background with particle systems to give the game an other-worldly feel and to simply add a dynamic quality to what would otherwise be a boring background.
- Closely match the sound effects of your menu to the feeling or theme of your game to begin the immersion process early. Consider slight variations of these sounds through pitch changes as long as these variations do not change the sound too much and break consistency.
- Menu selection doesn’t have to mean a scene change. Instead, consider moving the camera to another menu somewhere in the scene through the use of a curved path for a fun shift in perspective of the scene. This feels fun and keeps the game going instead of providing a brief break in immersion while a new scene loads.
- Keep consistent with UI colors to allow the user to establish known patterns for how to recognize game UI and know what is and is not interactable.

Destiny

Summary

Destiny is a first person shooter game that is set in a sci-fi universe with RPG-like elements. *Destiny* is a console-only game so it is controlled primarily through gamepad controllers. *Destiny* UI is critically acclaimed for their implementation of the “free cursor”. Essentially, the free cursor allows a user to navigate menus using a gamepad analog stick in much the same way as a mouse cursor. This design decision for the UI is crucial considering how information and statistics heavy the RPG elements of *Destiny* can be, keeping it from feeling less spreadsheet-like and easier to navigate menus in general.

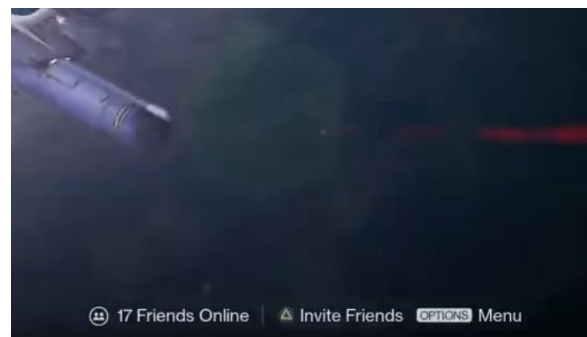
Key Functions

- The free cursor is a semi-transparent circle with a white outline that is moved about the menu screen much like a mouse cursor moves. Like in *Splice*, the camera moves with free cursor movement (this time in a parallax manner). This design decision eschews the traditional up/down/left/right node maps that most games use which works well for freedom of movement in an information filled screen.



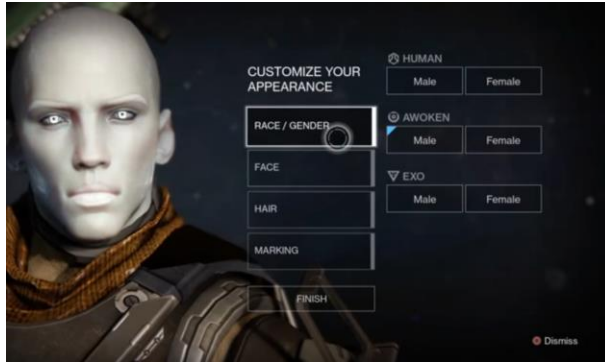
The free cursor is mostly transparent, allowing the user to still be able to see what it is on top of, while the faint white-outline of the cursor helps it from becoming completely lost.

- A semi-transparent black bar is located at the bottom of the screen when in menus and an additional semi-transparent black bar is added to tool tips when there are button interaction options available. This information is always displayed with button icons and text ensuring that the user knows how to interact with their gamepad and what the interaction option will do with the corresponding button press.



Interaction options are displayed with the button listed first on the left, followed by the explanation in text to the right of the icon. Text is kept as minimal as possible.

- Additional menu options and submenus are given tooltips that pop out from the side of the highlighted object. To indicate that the additional information belongs to the highlighted option, a thick white line is used on the side of the option where the additional information is displayed. This establishes a connection between the option and additional information when the cursor is then moved over to highlight the additional information.



When a submenu option is selected from a list of options a reserved color and shape combination is used to designate that option as selected.

Key Takeaways

- Once again, white is reserved as a UI color for both text and icons, only this time white is faded a bit when not selected and at full opacity when selected. Colors are used as a highlight to designate special selections or progression.
- Cursor movement is tied again to the main camera to give the menu a dynamic feel and make it fun to simply “move around” in the menu (this time with parallax movement).
- A persistent bar at either the top or bottom of the screen can be used to display icon/text combinations to help the user know how to navigate the menu and provide context information when needed.
- A free cursor can be effective even with gamepads when enough information makes traditional up/down/left/right node maps tedious and ineffective.
- Flat iconography can be used effectively to make things interesting without being overwhelming. Text is still the star when it comes to displaying need-to-know information, but icon paring can help make it visibly interesting.
- Consider using alpha transparency to show what is and is not currently highlighted. Be aware of color contrast with the background when doing this, however.
- Always let the player know where they are in a menu whether it is through breadcrumbs or by simply a title somewhere letting the player what menu they are on.

Star Wars: Tie Fighter

Summary

Star Wars: Tie Fighter is a space flight and space combat game based on the *Star Wars* movie franchise. Originally released for PC, *Star Wars: Tie Fighter* menus are navigated with either the mouse or keyboard. While *Star Wars: Tie Fighter* manages to capture that perfect *Star Wars*

feeling of being a pilot for the Empire and all of its menus capture this aesthetic well while still functioning well as menus.

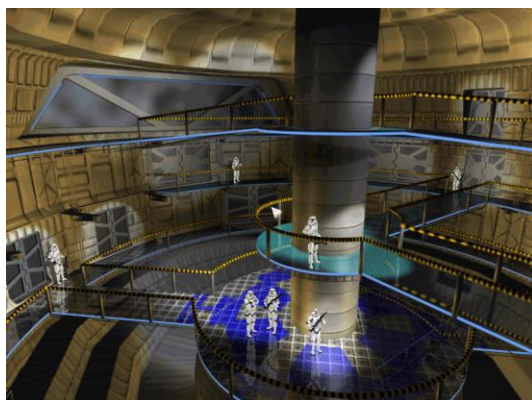
Key Functions

- Right from the start, the character select screen gives the user the experience of how intimidating the Empire is through the low camera angle looking up on the Imperial officer. This intimidation factor helps set the right tone for how the Empire views themselves as both a military juggernaut and superior.



Profiles show on the left, while the information for the selected profile shows on the right. If a profile is not selected, the storm troopers animate to block the door, not allowing the user to start the game.

- The feeling of being aboard a space ship is conveyed in the menus by making the options that you can select feel like they are aboard a spaceship. *Star Wars* music plays while hanger doors open with appropriate sound effects, while text displays what the option is.



The menus in the game make the user feel like they are exploring the space ship in a galaxy far, far away, while in reality they are only clicking through a menu.

- The ship selection and armament loadout menus make it look like you are in a ship hanger before playing a launching animation of the ship blasting off into space. All this makes the menus feel like you are already playing the game before you actually are. This is incredibly important because the user is already set up to feel like they are in the *Star Wars* universe.



Probably the most information heavy of all the menus, even the loadout selection screen is themed in a way to make it interesting.

Key Takeaways

- Each menu feels like a different environment and is not explicitly denoted with text until the user discovers it with the mouse cursor. This can lead to a sense of discovery while exploring the menu.
- Once a game's theme has been established, exploit it to the max to help the user immerse themselves into the game world you've created.
- Art assets can get you a ton of mileage in creating immersion before the gameplay has truly begun (granted you need to have the art assets first).

Project Proposal

Main Menu System / In-Game Menu System

Free Cursor

For the main menu system, I would like to create a system that utilizes both the mouse and keyboard, as well as an Xbox controller with a free cursor that is not unlike that of *Destiny* and *Splice*. Like *Splice* I would like the system to have an element to the cursor that animates to show selectable UI by either expanding around it or becoming part of the selectable UI element itself. Much like *Destiny* I plan on making the free cursor semi-transparent so that the user can see what is underneath the cursor, but with a white glow or edge to it so as to not lose the cursor entirely. This system will be used in the in-game menu system.

Context UI Bar

Throughout the main menu system, I would like to use a persistent context UI bar that is similar to that found in *Destiny*. The “always there” aspect of the bar may be a tad overkill, so I’ll have to test how that works out, but I like the idea of having it there updating with the context of each interactable UI object. Like in *Destiny*, I would like the bar to group icons with text so that the player always knows what button to press and what it will do. Unlike *Destiny*, I plan on supporting both the mouse and keyboard, along with gamepads, so I’ll have to make sure that context UI bar knows what the last used method of input is and swap out icons as needed. This system will be used in the in-game menu system.

Camera Movement

I really love how both *Splice* and *Destiny* make their menus feel dynamic through camera movement tied to cursor position. For this reason, I am going to utilize slight rotation and/or parallax movement tied to cursor position. In addition to this, I think menu transitions by having the camera follow a zoom in/out path could really make the menu fun, so I’ll be sure to implement some sort of spline path for the menu camera to follow between different menus or submenus in my main menu system. In-game menus will not feature this system.

Consistency

I truly believe that consistency is key in establishing patterns for ease of use for understanding and recognizing and navigating any kind of UI. I’ll be establishing a style with a set font and the color white will be utilized for all UI elements with a minor transparency. UI animations involving white lines or text will animate to full white when hovered/selected. I am

uncertain at this time whether I will reserve a color other than white for any additional UI purposes. Hierarchy in information will be established through size relationships and spacing to help create an order of importance for titles, headers, sub-headers, etc. Of course, this system will be used in the in-game menu system.

Theming

Perhaps the most difficult system to describe right now would be the theming for both the main menu and in-game menu system. Theming is highly dependent upon the game that these systems are being made for. I am interested in making a space flight and space combat system, so I suppose making a game that makes the user feel like they are on a space station, not unlike that of *Star Wars: Tie Fighter* is something I want to touch upon for this menu project.

To start, I want the main menu system to feel like a hanger bay, with the player's ship in view. Moving to different menu options will trigger a camera movement to a different part of the hanger, displaying relevant information for that menu. For example, clicking on the weapons for the ship will zoom the camera towards the munitions that can be installed that are right by the ship. Clicking to go back out to the main menu will zoom the camera back out to reveal more of the hanger, but with the player's ship as the central element in focus. The in-game menu system should feel like you're looking at diagnostics on the ship by tilting the player's view as if it were their head towards a screen away from the viewport of the ship.