CMPUT 302

Human Computer Interaction

Project Design

Interactive Surface - Target Tapping

By: Adam, Shawn; Boutet, Patrick; Padlesky, Aaron; Santos, Eddie

Motivation And Goals:

From the start of this project, each group member has some unique motivation and goals; simultaneously, the group as a whole share some of the same motivations and goals. The first common motivation for choosing this project was to learn or work with C#. Three of our members have had no, to very little experience with C#. The remaining member has used C# extensively in the past. The experience of our one member greatly increased the motivation of the remaining members to learn the language and attempt this project. Those without C# experience believe this is a great way to quickly pick up the language, and have in-group support when issues arise. Much of the same motivation was attributed towards using the XNA library as well. We see learning both C# and XNA as great experience for future endeavors and we look at it as a change from the traditional UNIX development we have become accustomed to.

We are also thrilled with the prospect of working with a client outside the academic world. Developing a program for the Glenrose Rehabilitation Hospital will provide us with experience dealing with a type of client most of us have not yet dealt with before. In addition, for many of us, this will mark the first time that there has been a possibility of our work being used on a day-to-day basis outside an academic setting. Having our work being used daily helping disabled patients regain movement is a rewarding task all on its own. We are very committed and confident towards delivering the right product the client is looking for.

Above all though we are grateful and excited to develop our skills not only by dealing with a client and learning some tools in the .net environment but to develop our skills in game development, interface design, and human-computer interaction as a whole.

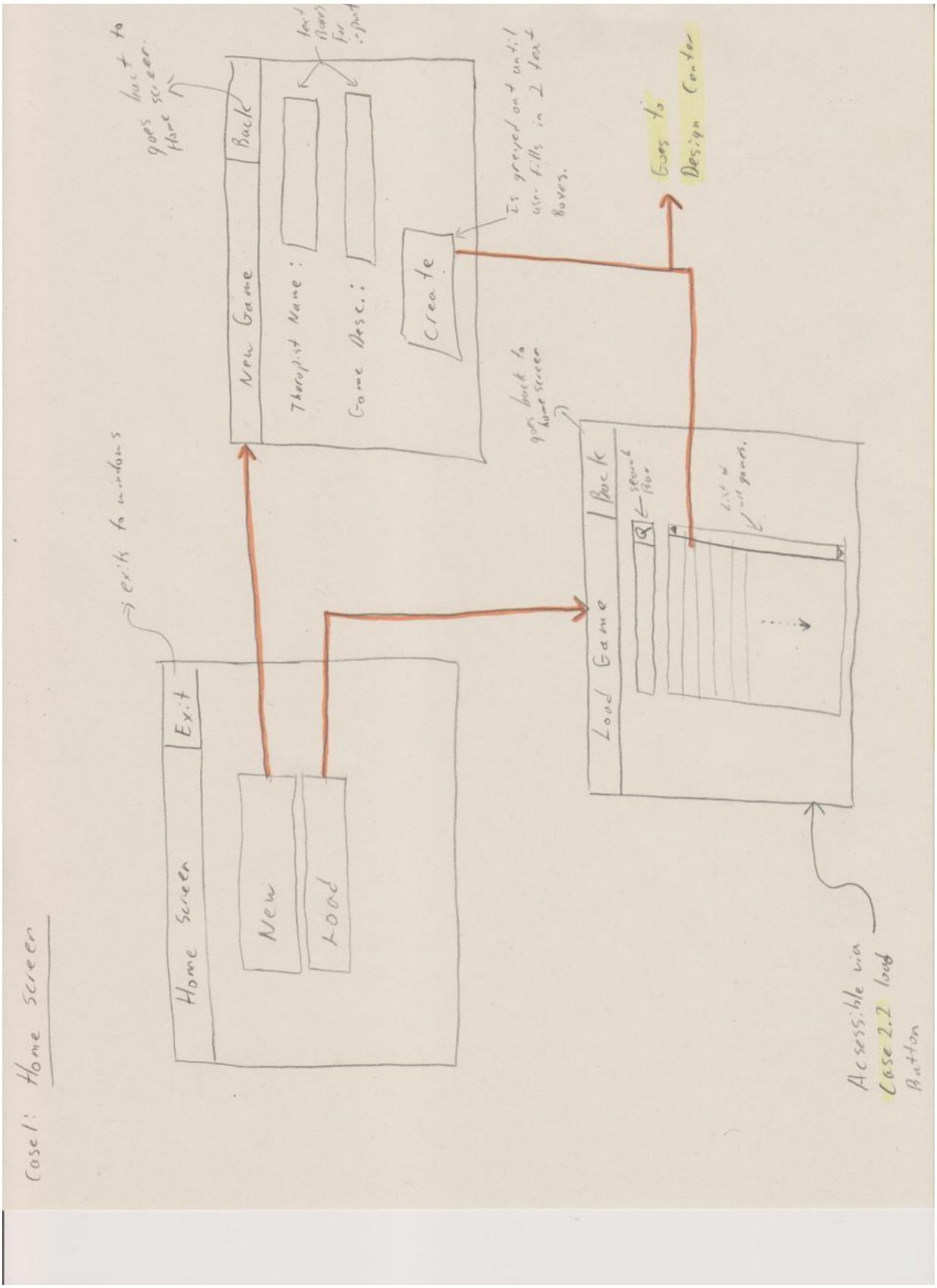
During our first couple meetings we discussed many of the goals we have. Our first goal is to simply make a program and tool that the therapists want to use. More specifically what we are hoping to achieve is to provide a high-quality interface that is simple and intuitive. We have debated as to which direction we would like to take this project. For a while our group of four was split on our ideology towards implementing this system. On the one hand we wanted to make a program that would give therapist all the tools they could dream about. On the other hand, we stressed simplicity, meeting the defined requirements, and planning to do little more then what was asked for to the best of our ability. Moving forward our approach will be to favor simplicity and quality over features. We intend to deal with changing requirements by writing quality code. If the need arises to add more features, we will deal with that when it comes. Until that time comes however we hope favoring simplicity and quality code will give us the end product we hope to deliver.

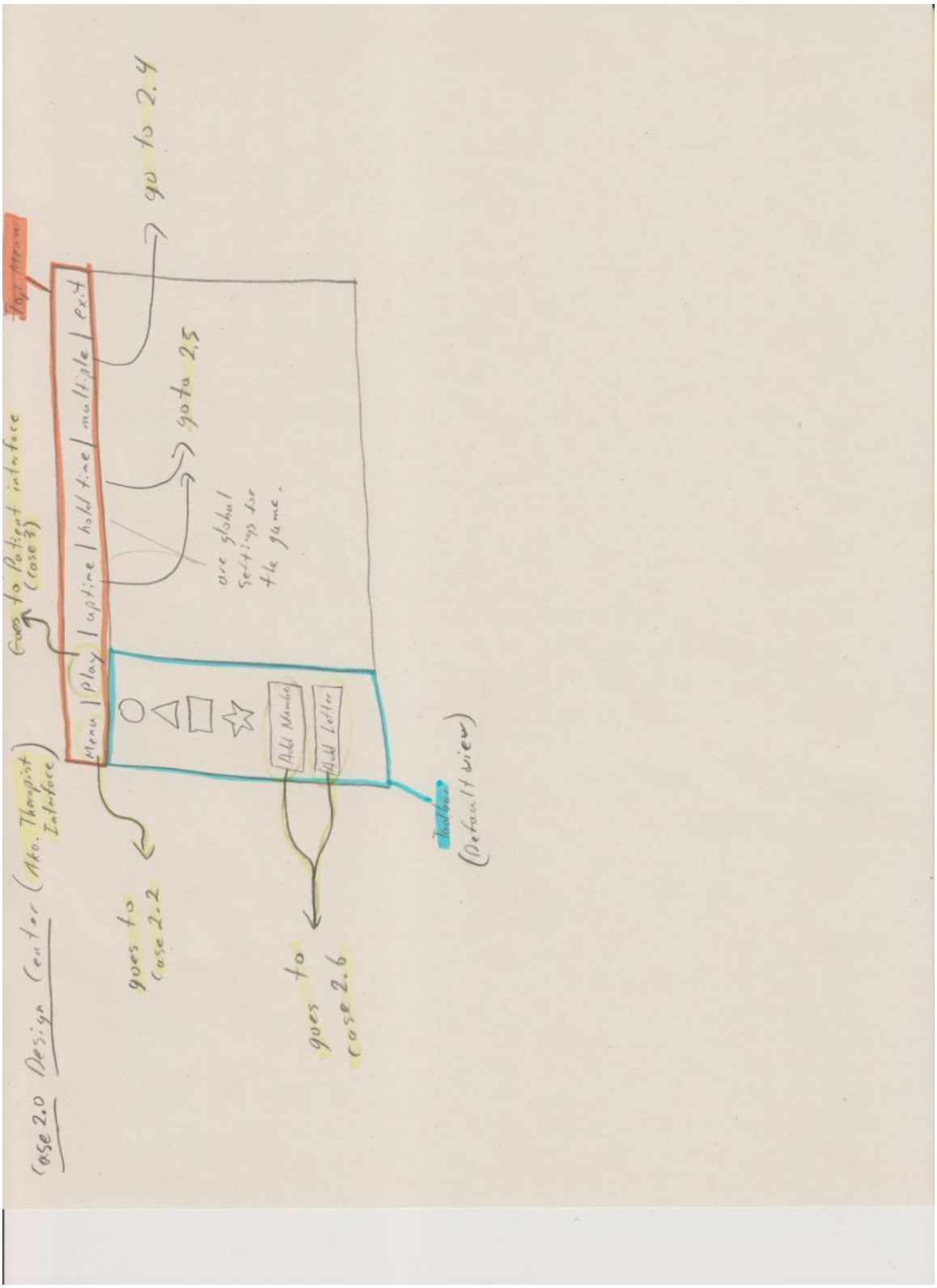
We have sub-goals to help us in achieving our end goal as well. Intuitiveness is our first sub-goal. “Intuitiveness” means that the therapist using our application can look at the interface and immediately know how to work the application. We want to minimize the use of complicated menus. We want the user interface to guide the therapist in how to set up the right game for the patient. For the patient then, we want their interface to be simple and solid. We realize that the patients will have a variety of conditions ranging from coordination problems to having very limited mobility and cognitive ability. Therefore we seek to eliminate all possibility that the patient may perform accidental events with the system. That is what we refer to as a solid patient interface. Our next sub-goal is to create an appealing GUI for the client and patient. We understand exorbitant amount of detail may overwhelm the patient rather then delight the patient. As such, all of our graphics are to be very simple, yet appealing and professional looking. Another sub-goal is to favor ease of learning over everything else. This means we want all therapists to master using the game. We will keep all un-needed features out, and will continually think of how to make every part of the user interface as simple to use as possible. One very specific goal we set was for at no point in our programs execution should we ever need a keyboard or mouse. We want all input to be touch based, and without the need for typing of any kind.

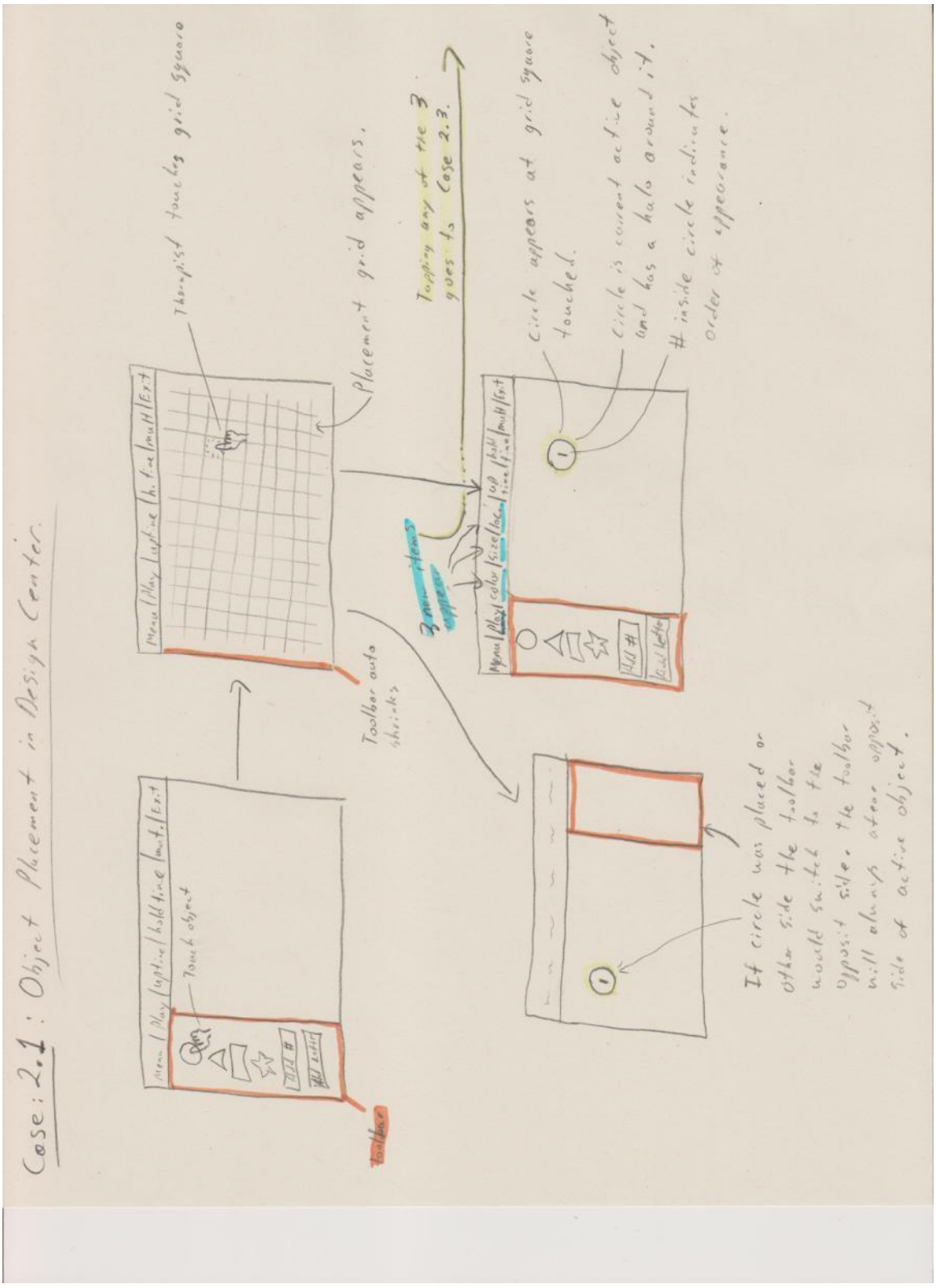
We have a second set of goals. These goals refer to development. Our first goal is to have working prototypes available very early in development. Next we will aim to have all absolute requirements done long before the delivery date. Nearing the deadline we hope to be polishing, editing, and testing, rather than adding and completing requirements. We hope that we have the right motivations, and set the right goals for ourselves to develop a quality game to the user.

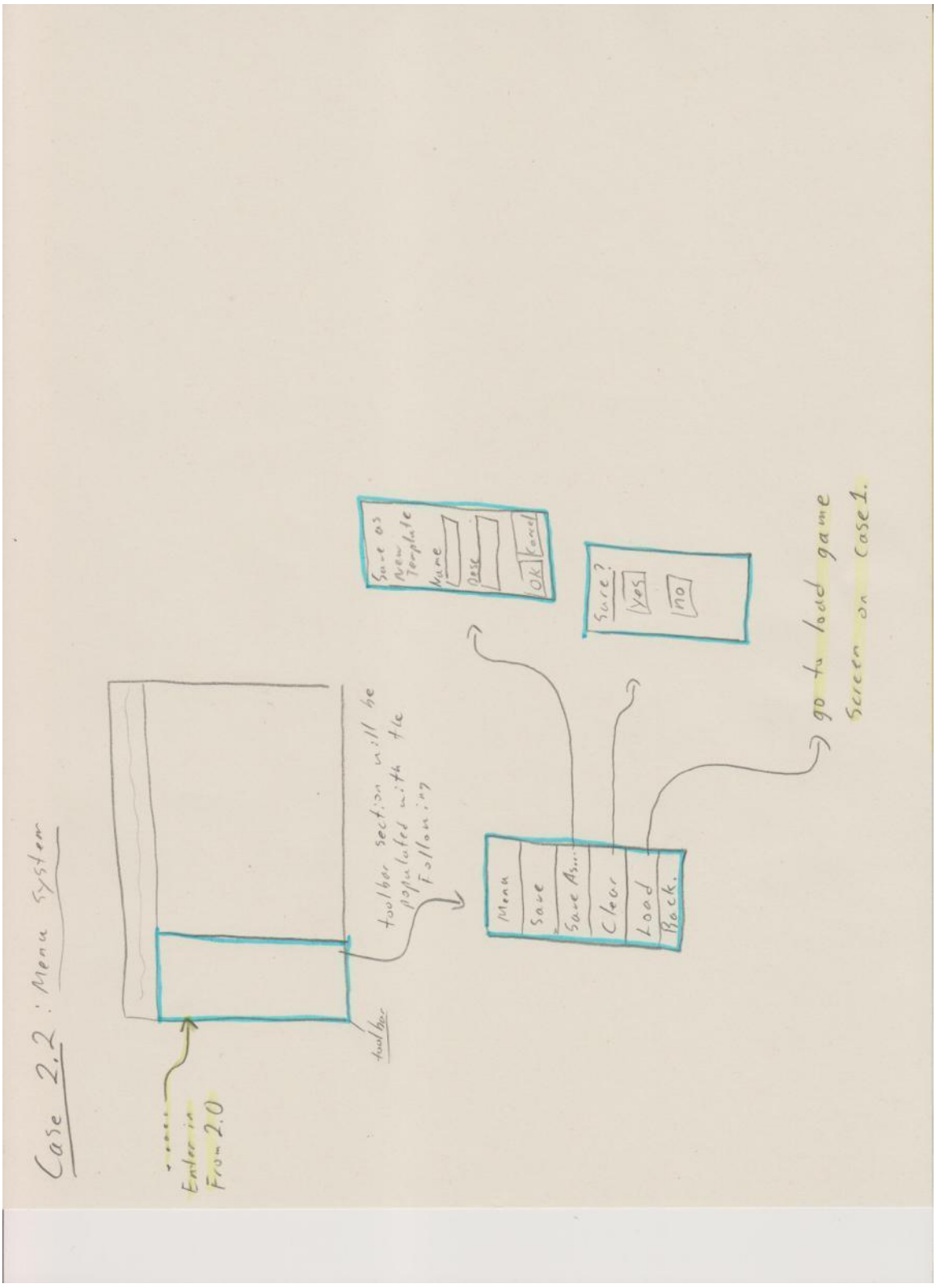
Interaction Design:

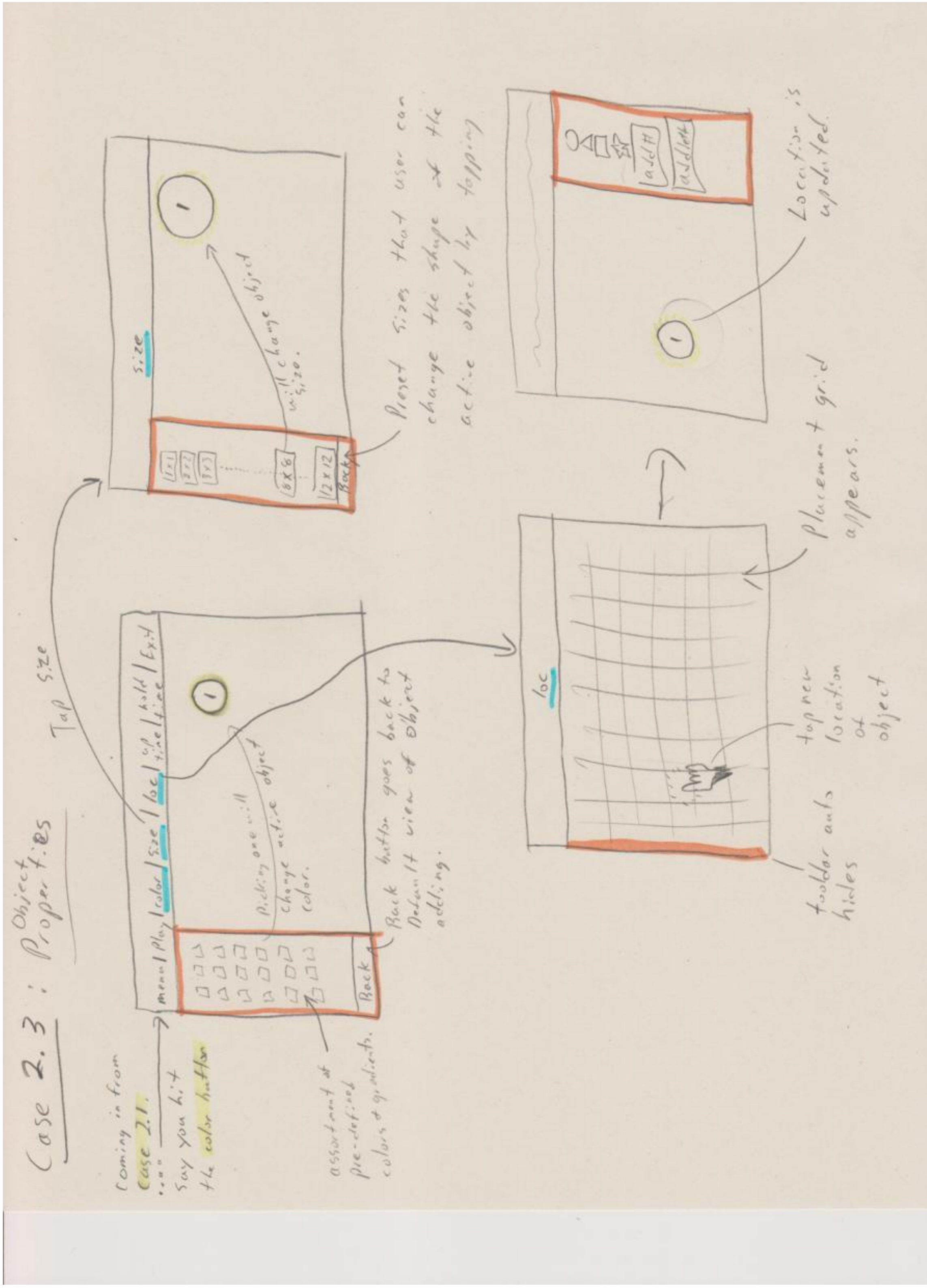
The following pages area mockup of our user interface in the context of multiple use cases.

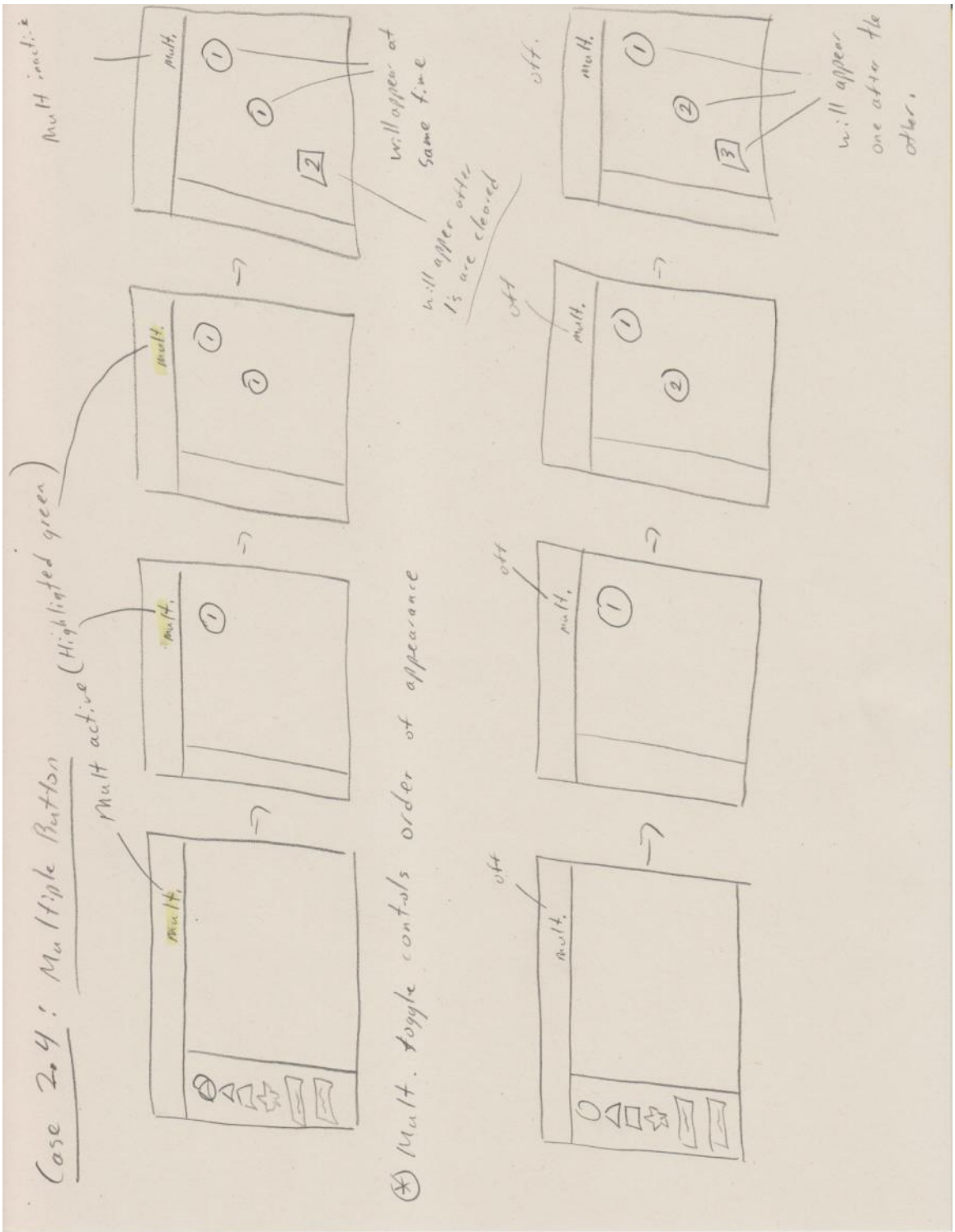

Case 1: Home screen
Home Screen
Exit
New
Load
exits to windows
New Game
Back
Therapist Name:
Game Desc.:
Create
text to goes Home screen
text boxes for input
ES greyed out until user fills in 2 text boxes.
goes back to Home screen
Load Game
Back
Search for
List of user games.
Goes to Design Center
Accessible via Case 2.2 load Button

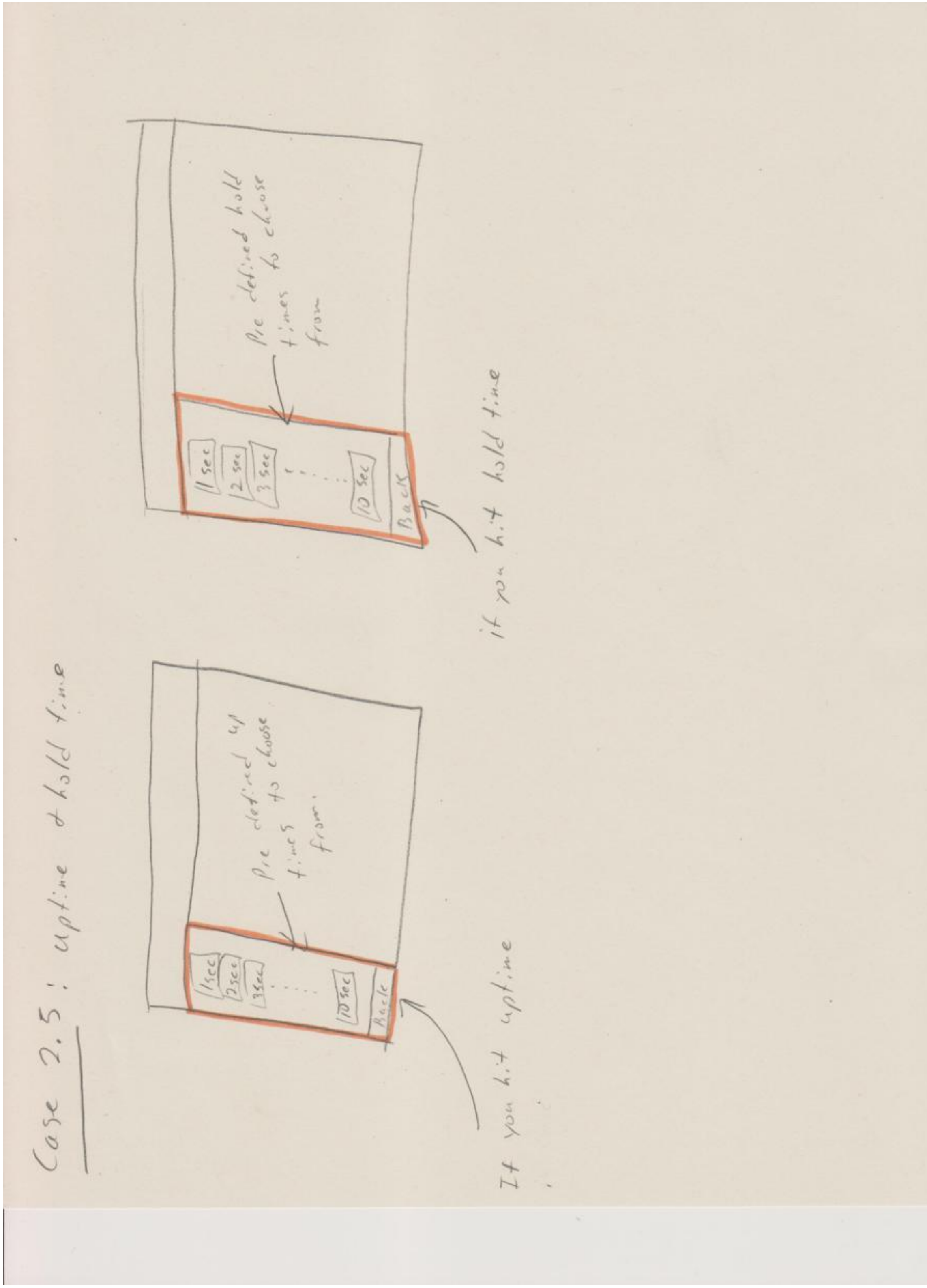

Case 2.0 Design Center (Aka. Therapist Interface)
Goes to Patient interface (case 3)
Menu | Play | options | hold time | multiple | ex.4
go to 2.4
goto 2.5
are global settings for the game.
goes to case 2.2
goes to case 2.6
Add Number
Add Letter
Toolbox
(Default view)

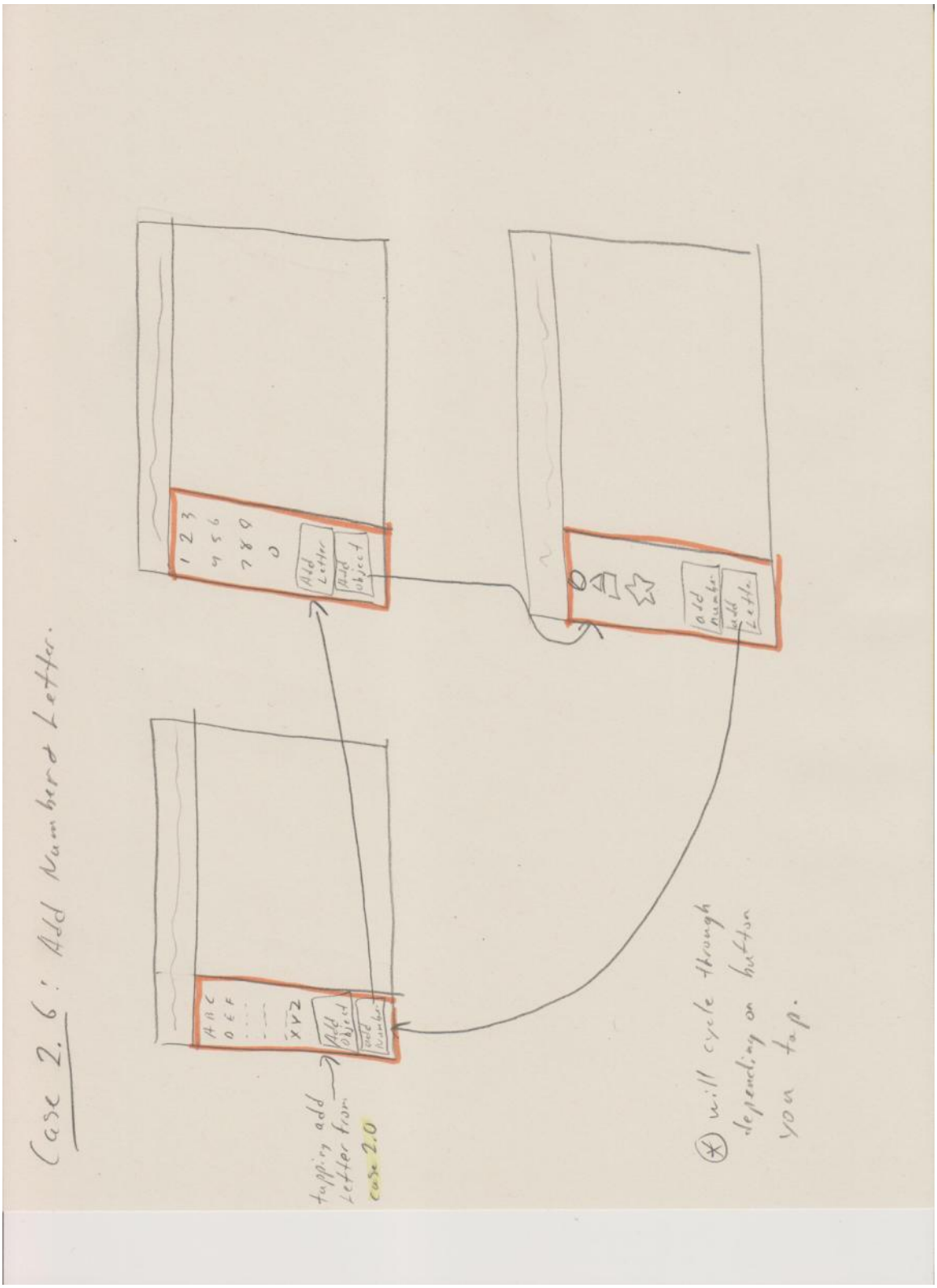

Case: 2.1 : Object Placement in Design Center.
Touch object
Toolbar
Toolbar auto shrinks
Placement grid appears.
Tapping any of the 3 goes to Case 2.3.
Circle appears at grid square touched.
Circle is current active object and has a halo around it.
inside circle indicates order of appearance.
If circle was placed on other side the toolbar would switch to the opposite side. The toolbar will always stay opposite side of active object.

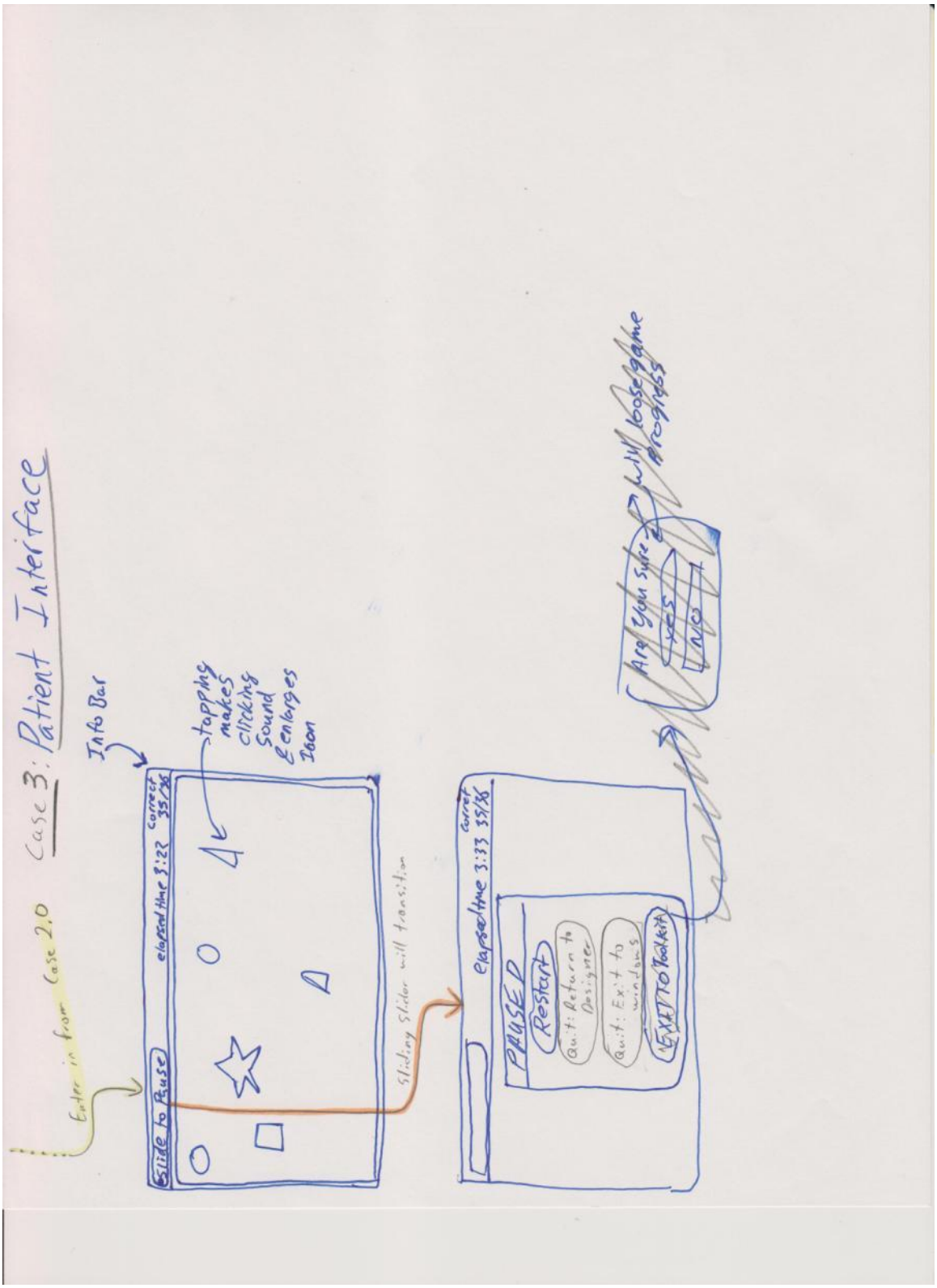

Case 2.2 : Menu System
Enter in From 2.0
toolbar
Toolbar section will be populated with the following:
Menu
Save
Save As...
Clear
Load
Back.
Save as Name Template
OK
Cancel
Sure?
Yes
no
go to load game Screen on case 1.


Case 2.3: Object Properties
Coming in from Case 2.1. ... Say you hit the color button
Tap size
size
loc
Back
assortment of pre-defined colors + gradients.
Picking one will change active object color.
Back button goes back to Default view of object adding.
will change object size.
Preset sizes that user can change the shape of the active object by tapping
Placement grid appears.
toolbar auto hides
tap new location of object
Location is updated.


Case 24: Multiple Button
Mult active (Highlighted green)
Mult inactive
Mult.
will appear at same time
will appear after 1's are cleared
will appear one after the other.
off
* Mult. toggle controls order of appearance


Case 2.5: uptime + hold time
1sec
2sec
3sec
10 sec
Back
Pre defined up times to choose from
If you hit uptime
1 sec
2 sec
3 sec
10 sec
Back
Pre defined hold times to choose from
if you hit hold time


Case 2.6: Add Numbers Letter.
1 2 3
4 5 6
7 8 0
0
Add Letter
Add Object
A B C
D E F
. - . .
. - - -
X Y Z
Add Object
Add Number
tapping add Letter from case 2.0
Add Number
Add Letter
* will cycle through depending on button you tap.


Case 3: Patient Interface
Enter in from Case 2.0
Info Bar
Slide to Pause
elapsed time 3:22
correct 35/35
tapping makes clicking sound & enlarges Icon
sliding slider will transition
elapsed time 3:33
correct 35/35
PAUSED
Restart
Quit: Return to Designer
Quit: Exit to windows
EXIT TO Toolkit
Are you sure
YES
NO
will lose game progress

System Design:

This application deliverable will be required to install and work alongside the Windows 7 operating system. We will be using C# and the XNA framework to implement the deliverable. There is a significant amount of code available for us to use if we stick to these development options. This will significantly lower many risks that we face in this project. One risk we could face is lack of knowledge in the subject area. By using as many resources as we have at our disposal we can decrease the amount of learning and time it will take to gain a strong working knowledge on the subject area. This will also decrease any time risks we have. The less we have to write, the more likely we will be able to meet the client's necessary requirements sooner. Hardware will consist of a PC running windows 7, and a custom designed touch screen tablet. This tablet is constructed from a large flat screen LCD TV with 1080P screen resolution. The television speakers are connected to the laptop, which allows us to implement audio feedback. Supporting the television is a desk-like custom device that is capable of many adjustments. A custom touch screen is positioned directly on top of the televisions screen. As of now, our plans are to be using the touch interface as the only source of input for the entire application. This way once the application is loaded, the applications will appear to be self-contained inside the touch device. Our goal is to give this custom application the feel of a stand-alone tablet, which many people are now becoming quite familiar with.

The application itself will have two main states. From now on we will refer to these states as the patient interface and the therapist interface. The patient interface will be very simple and intuitive. Essentially the only interactions available inside the patient interface are to tap/tap&hold the symbols that will be used for patient rehabilitation; and to slide across the menu bar to pause the game. Upon pausing the game from the patient interface a menu will appear giving options to the user on how to proceed next. The other side of the application is the therapist interface. The therapist interface will allow the user to start the game for the patient. As well from here the therapist can select previously saved game configurations, as well as creating new game configurations.

To develop this system, we will have our program retrieve and interpret all touch signals sent by the touch screen. We have been informed the touch screen will send updates at 1400 hz. The computer will receive these updates and process the events that have occurred on the touch screen. We will have the computer update the display with feedback showing the computer does in fact recognize the screen has been touched, and that the event is being processed. In this case we hope to give feedback through slight clicking sounds, and visual events that would be very similar to how a physical object might behave. The display of the TV will be our intended application to view the interface. Therefore we will optimize the game to be displayed in 1080P (1920x1080) resolution. To remove complication from the system we hope also to use an on screen touch keyboard whenever input is required from the user for labeling.

Implementation Details:

To first note, the implementation of our application will evolve over the duration of the project as we become more familiar with C# coding and best practices. Initially this is the general layout of the code structure we will begin with. We will have a myProgram class that contains the main method and initializes our myGame class. The myGame class inherits from Microsoft.Xna.Framework.Game and within the myGame class will be its own constructor then the initialize, loadGraphics, update, and draw methods. The myGame class displays the initial start menu for the game. Other classes that will be called from the myGame class will include myNewgame which handles creating a new game that the therapist wants to make. myLoadgame which handles loading an existing game. As well there will be a mySavegame class to handle saving of games. There will be the myDesigncenter class which handles displaying the design center (aka. therapist interface) and its functionality. There will also be multiple classes such as myMoveObj that will handle moving objects around in the design center that will be called from myDesigncenter that will handle the functionality that we envision for the design center to have. The myToolbar and myTopmenu classes which handle the menus within the design center and are called from within myDesigncenter. Classes for all the objects that the therapist can generate on the game area, such as circles, squares, letters, numbers, will also be present and most likely called from the myDesigncenter class. A myPatientview class that will handle the game that the therapist created and allow the patient to play it will also exist. The myPatientview will also call classes that handle menus that display things like pausing the game, restarting, resuming, quitting, returning to design center.

Lastly in order to help organize our code we will organize these classes into several packages that group similar functionality or closely tied classes together. These will be generated as we code and based on the needs of our program code structure and organization.

Proposed Empirical Evaluation:

In our project development we plan to have a functioning prototype with all required features completed long before the delivery date of the system. With this functioning prototype one of our objectives it to have it tested, in a very qualitative way, by a therapist that will be a potential user. Our objective is to first understand how well the user can infer the usage and the features of the program. After we evaluate the user's ability to use the system without help, we will request they rate the systems usability and learnability on a scale from 1 to 10. We then will explain the system and how we intend it to be used, after this we will give them another chance to explore and use the system, and once again get them to make a rating. If we find that the systems usability and learnability is sufficient we will feel quite confident in our design and move on to more quantitative testing. If however they find the usability to be unacceptable without first being explained the system, and

once again find it difficult after explanation we will have to seriously re-evaluate the design. However either way we will be asking those users who tested the system what kinds of changes, additions, or subtractions to the system would be helpful to aid in ease of use.

As mentioned above if our qualitative testing is successful we will attempt more rigorous quantitative test methods. We will gather approximately 20 user's worth of data; these users will be general population as for this test they simply have to be unfamiliar with the interface we designed which would be an identical situation to a therapist using it for the first time. We will write test code into the program that is able to track the time between different actions. We will then explain how to do an action, such as create a new design template, change an objects color, size, or location, etc.. The user will then attempt the action and the code will track their time. Once we have time data on actions from general users we will perform the same tests but with ourselves as experts on the interface. We can then take the average time of the users (beginners) and compare it to our average x 2 (experts) via a t-test. Then see how different the two data sets are and based on if we see a statistical difference or not make two conclusions. First is if there is a difference between the two data sets then we can say that our interface is not as usable as we would like it to be and take the effort to find out what the roadblocks for users are. Second is if there is no difference between the two data sets then we can say that we have built a very usable interface with a low learning curve. Overall we want to be testing how long it takes the users to figure out the user interface instead of how to use it.

Risk Management:

After meeting with our client, and determining many of the requirements for this system we have identified risks that could jeopardize the success of our project. The first risk we identified was the group lacked experience with C# and the XNA framework. We believe by using the one member of our group who has C# experience we can reduce this risk. Our strategy also includes keeping code as simple as it needs to be to meet requirements while using good coding style and standards. The lack of access to the hardware also poses a risk. With less time to test on the actual hardware it is possible to miss bugs at the times when we have to rush testing when the hardware is accessible. We have been assured however that mouse and keyboard testing will be quite accurate and representative of the performance of the real hardware. To lower this further, on the days we do have access to the hardware we will plan our tests and be prepared to maximize efficiency. Collaboration always poses risk. More specifically collaboration can be plagued with disagreements, conflicting personalities, skill differences and skill similarities, time management and more. To reduce these issues and the risk they bring we will hold weekly meeting, use voting to decide conflicts, always seek to maximize the benefits of multiple ideas and mind, and use a Wiki for constant communication, updates, technical details and to review issues. The last risk we feel is important to address is

meeting deadlines. We feel lack of experience in this area could lead us to run into unforeseen delays. As such we plan to have a working prototype ready early on, and always strive to meet the important requirements first. If we have a strong basis to work from, we feel we can limit the risk of nearing the deadline and being forced into a rush to accomplish functionality. To monitor all these risk we will constantly re evaluate and discuss risks, always looking to identify new risks. We will post all risks to the wiki, and monitor their probability and effects.

Project Schedule:

To have the finished product delivered on time and to specifications will be the most important requirement there is. To make sure we meet this requirement we have thoroughly thought through a schedule we plan to follow faithfully. We plan to have a solid functioning prototype at the half waypoint we have set. This will make sure we have the main requirements as well alleviate last minute stress. To limit the possibility of unexpected delays affecting the completion of the project we have set all main requirements to be met as soon as possible. In this way we plan to develop the system in layers. First meeting main requirements, then adding layers of functionality and performance. We have set many milestones and many tasks that will allow us to achieve said milestones. One is to develop custom aesthetic graphics for the application. The front end will consist of 5 main milestones. The menu bar and grid inside the patient interface; and menus, design screen, and home screen inside the therapist interface. These larger milestones will include smaller tasks inside their development. On the backend we have set the ability to create levels, the ability to save levels, the ability to process touch, the menu backend, event handling and object creation all as milestones. Again these milestones will have smaller self-contained tasks that do not cut as widely across the application. Deliverables include project design, ethics, prototype 1, the final product, and all code documentation and design documentation.

The above deliverables, milestones and their affiliated tasks have been laid out in the below Gantt chart which specifies specific dates and durations for the project.