Project Overview
Team Astropolis was tasked by the sponsor with creating and coding a mini-game from the ground up to add into the existing Astropolis framework. The scope of our project was to help complete not only this goal but also but some time into refactoring some of the more buggy parts of the existing code base. This portion of our task was largely confined to fixing some glaring errors in the coding of the Star Jack level editor. This left the scope of the remaining project to be a rather simplistic game which was finally narrowed down to a matching game using faces. This game had to be able to be coded in 20 weeks, and use facial matching to help identify the way people with autism spectrum conditions interpret faces.

Basic Requirements

Face Submission

3.1.1 Description and Priority
The customer should be able to easily add images into the game to be used. When uploading an image, the user should be able to specify which category of emotion that the picture belongs to. This task is of Low priority.

3.1.2 Stimulus/Response Sequences
When starting the mini-game, the user will be given the option on the main menu to add faces to the game. This will load an interface that will as for the path to the desired picture and the emotion of the picture. The system will then save the image into a directory based on the emotion of the face.

3.1.3 Functional Requirements
REQ-1: A picture must be added to the game.
REQ-2: A picture must be associated with the correct emotion.

Facial Expression Matching

3.2.1 Description and Priority
Users will be rewarded points for identifying which faces are missing from the game board. Refer to section 3.2.2 for more details. This feature is High priority.

Stimulus/Response Sequences

The users should be presented a game board of faces in a grid. See Appendix for screenshot. Certain faces will be removed from the grid and lined up at the bottom of the screen. The user must select which missing face will create a row of 3 identical emotions in the game board.

Functional Requirements

REQ-1: A game board will exist in a 3x4, 4x4, or 4x5 grid.
REQ-2: The game board will be filled with faces depicting emotions.
REQ-3: Random faces will be removed from the board and replaced with a blank space.
REQ-4: Faces will appear in a row at the bottom of the screen outside of the grid one-by-one at an interval of 1 second +- 200 millisecond jitter.
REQ-5: Players can place a face back on the game board by clicking on the face and then clicking on the blank space on the board.
REQ-6: Players can remove faces they have placed by clicking on the face and pressing a remove button.
REQ-7: Players will press a button labeled “Submit” to begin validation of the game board.
REQ-8: Players can press “Submit” with one or more blank space filled with their answer.
REQ-9: Pressing “Submit” will validate the board by checking to see if the filled in spaces connect 3 like emotions.
REQ-10: After pressing “Submit”, if the user-filled blank space does not connect 3 like emotions horizontally or vertically, then the answer was incorrect and the face is removed from the board and placed back at the bottom of the screen outside the grid.
REQ-11: If the answer was incorrect then the player loses points.
REQ-12: After pressing “Submit”, if the user-filled blank space connects 3 like emotions horizontally or vertically then the player will be rewarded points.
REQ-13: After pressing “Submit”, bonus points will be given if the player gets multiple correct answers.
REQ-14: There will always be faces available that will successfully connect 3 like emotions.

Events Logged

3.4.1 Description and Priority
All of the events in the game are logged into a file with the event name and time stamp. High Priority.

3.4.2 Stimulus/Response Sequences
Events from the system and user will be logged constantly. Events include: The appearance of the faces, the selection of faces, the placement of faces, etc.

3.4.3 Functional Requirements
REQ-1: Each log entry must have a timestamp.
REQ-2: All events are logged in a log file.

Multiple Levels

3.1.1 Description and Priority
The mini-game will consist of multiple levels. At the completion of a level, another board will be generated. Each new board generated will be slightly more challenging by adding new rows or columns to the board, adding more possible faces to fill-in the blank spaces with, or by adding emotions to the board that have subtle differences. This task is of High priority.

3.1.2 Stimulus/Response Sequences
Each level will be randomly generated when the new level is started.

3.1.3 Functional Requirements
REQ-1: At the completion of a level, a new level will be generated.
REQ-2: At the completion of the 5th level, the game will end.
REQ-3: More points will be given for more difficult levels.

Constraints
The team was required to work within the already existing Astropolis framework and we were limited to the technologies already implemented. We were also limited by the budgetary constraints of the grant that Matthew had for the research project. We were also constrained by the fact that the users would be high functioning autistics this was probably the largest driving factor behind many design decisions.

Development Process
Extreme programming was the process used by the team. This process was not explicitly approved by the sponsor but our sponsor seemed to approve of the way the process was used. The bi-weekly builds allowed Matthew to follow the team’s progress and see the state of the game. This also allowed for the team to garner more feedback from Matthew as our domain expertise was rather minimal. Our communications process was twofold
mainly we communicated by Skype every Wednesday at 5 pm we also used email to facilitate any gaps. The team had 3 roles defined project lead was Robert Larivee, William Wood was our scribe, and Phillip Levinson was our sponsor contact facilitator. Due to the small nature of the team these positions tended to be shared among the three team members.

**Project Schedule: Planned and Actual**

We started our planning based on the fact that Matt wanted some upkeep done to the overall astropolis system along with the creation of a new mini-game. We also had to make sure our schedule accommodated the extra time that would be spent on designing a mini-game. We did not have access to matt’s students who normally help with the creation of new mini-games. This made it so we were expecting to have a long drawn out requirements gathering process which we described during the first presentation. We had planned to work on mini-game at an earlier point in the schedule but we didn’t start serious work until the 10th week. This actually worked out well because it stretched out our requirements gathering which was important. As the game was fleshed out the requirements had to change to meet the specific needs of the target users with autism spectrum conditions.

**System Design**

![System Design Diagram](image)

This describes the architecture that we were given when we started the game and how the interactions of the game work. This was not really a choice on our part as XNA had already been chosen and the logger and Colony Simulation stuff was already created. It then made sense to use the already created code base as we knew that it worked.
Process and Product Metrics

Our metric was Project velocity this estimates the time that we think each release will take and compares it to the time it took.

Release 1
Generate Board Grid est. 5 Days act. 6 Days
Randomization of the faces est. 5 Days act. 8 Days

Release 2
Randomization of the faces est. 5 Days act. 5 Days
Face Tray est. 4 Days act. 4 Days
Jitter and artwork est. 5 Days act. 5

Release 3
Game Mechanics est. 10 Days act. 10 Days
Artwork and Start Screen est. 4 Days act. 4 Days

Release 4
Final polish est. 14 Days act. 21 Days

Product State at Time of Delivery

The state of the project at delivery time is that we have delivered the expected functionality and the only missing part currently is the tutorial which is expected to be finished by Friday. Overall we found that toward the end when we actually had hard implementations of a lot of the more specific parts of the game and Matt got to play around with the features there was some small feature creep. The one thing that we did not expect was the .exe based settings file and this is a mistake that was only found out recently. We do not have any plans to add that but our current implementation could be easily modified to use one in the future.

Project Reflection

The thing that really made the project work was the long time spent gathering requirements. If we had jumped into coding like the teams that had worked on the project previously we might have suffered the same issues. Even with this added buffer we still had some requirements creep. Also our use of the XP process worked out really well it helped hold us to release dates and also provided our sponsor with a lot of feedback. The only things that we are disappointed in was the lack of team members if we had more people we probably could have fleshed out the game a little more. There are a few things that Matt would still like to have added into the game but as the primary stakeholder he wants a lot of things added into the game.

References

1. For Project Support
   a. Sponsor Matthew Belmonte
   b. Faculty Coach Stephanie Ludi
2. For Game Content