Alaaeldean Omar

Christopher Peralta

Connor Farrel

Design Approach for Slippery Turf Wars

Starting out we had no particular design approach for our project, so you could say we used “Big Bang” for our lifecycle method. However, the first thing we did was meet up and spitball ideas for how the rules of the game would work. We used a whiteboard to draw a grid and sketch how the turns would play out to see if things made sense on a practical level. Once we felt comfortable with how everything would play, we listed out our major requirements for the project, rules, functionality, etc. and began our process for Slippery Turf Wars.

Requirement Analysis

Our idea for a game is a “turf war” type of game. We have a grid full of grey tiles, having four characters on each corner and the objective of the game is to cover as much of the grid as possible with your character’s color by the time the entire grid is filled out with colors. Using items that spawn every few turns, and a potential mechanic of “stealing” chunks of turf off another character by cutting off an area they conquered to turn the tides of each round, we wanted to make this game engaging, fast-paced, and addicting for players to enjoy.

Planning

As mentioned above, our planning only went as far as drawing out how a single game would go. We did not spend over a day on this planning process, and were satisfied by just simulating one run on a whiteboard and moved on to the next stage.

Software Design

Once again, there was no development for this stage. As we developed parts of our game, we found assets online and placed them on whatever part of the game was finished or we were planning to work on next. For example, before creating movement for our slimes on the board, we found assets of slime images to place on all four corners of the grid to see what they would look like before moving them.

Software Development

No planning for this process either. Once one of us began coding the basic stages, such as making the two dimensional array and getting the slimes to render and music to play, one would pass the baton on to the other and build code off of each others work. Some days we would sit together and try to solve some bugs on one computer, but it was difficult to be fully coordinated on this stage. In addition, not having any software such as GitHub to be able to save and load each others progress off of there was another reason for our lack of proper planning, let alone learning it.

Testing

One could say our testing process was similar to the iterative lifecycle method. Each time we implemented something new we would test it out right away to see if it works correctly or not. However, we would not develop it to completion before testing it. Using the slime moving example again, once we found assets for slime images to put on the field, we first checked if they were rendering correctly. Right after that, we implemented movement only for the one player you control and tested it out. We had our trial-and-errors, but once that worked out we moved on to AI, and so on.

Deployment/Maintenance

Currently, there are no plans to officially deploy this project. However, on the chance that we decide we may deploy it, we may first try to add some other features or stretch goals we may not have added due to lack of time, such as conquering the inside of a square that you draw with your slime.

Conclusion

All in all, this was a very fun project to make, and a great new experience of us. This may be something one or a few of us will want to add to our resumes if we are pursuing a position in the game development industry in the future.