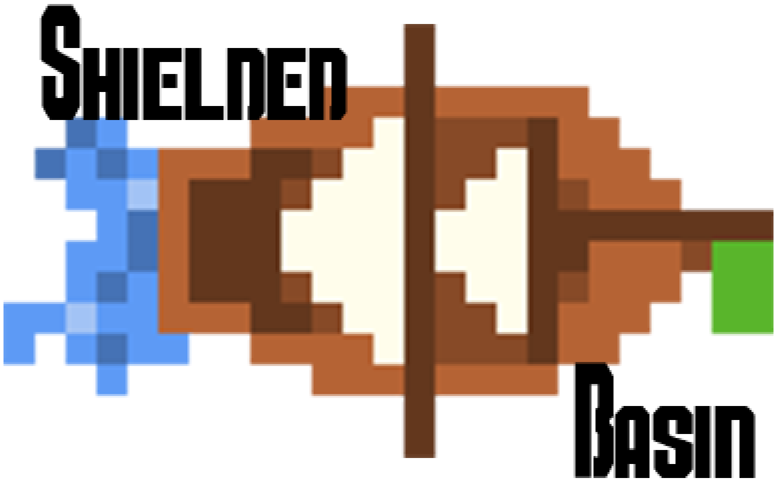

A game by Troy Kuang, Natasha Decoste, Nicolas Tristani, Dragan Visekruna, and Nathan Mangaoil

Description, Identity, and Inspiration:

Our Game, Shielded Basin, identifies itself as an open platform, multiplayer, intuitive game. Taking inspiration from games like Agar.io, we aimed for the game to not require a specific platform, as long as the player has a browser, mobile or stationary, they can play our game. This game is targeted for the browser platform (on all devices supporting modern browsers) and targeted to all audience experiences and genders. Through the use of the games mechanics, features, interface, and artstyle, we believe we successfully developed a game that a player of any age and experience can easily join, understand, and have fun.

Mechanics and Features:

Shielded Basin is a top down, pirate ship game that uses intuitive controls to navigate the player through the game. Each player spawns as a ship on water, from there they can adventure where they please. For keyboard users, we kept a familiar control scheme of WASD for directional input, as well as the arrow keys if a user find it more natural to move their ship using those inputs. We also implemented nipple.js, a library that allows us to create a virtual joystick. This allows players to easily control their ship if they happen to be on a mobile platform, or any touchscreen platform.

Shielded Basin features a score system that rewards players for collecting coins before other players. These coins are tallied on the user interface next to a scoreboard that keeps tracks of the top three scores in the current game. This is the players ultimate goal, to have the highest score. Also on the user interface is a mini-map, that will give the player a general sense of where they are on the map, and where other players are relative to them.

Shielded Basic features hazards in the forms of a kraken, islands, and other players. Players must be aware of these hazards as the adventure, as contact with any will result in the player losing their points and restarting from zero. All these hazards, excluding players, are not displayed on the minimap to players, and requires the player to also scan their surrounding as to avoid these hazards.

Interface, Art Style, and Sounds:

Shielded Basin features a clean user interface. We wanted the interface to not be obstructing and easy to understand. A small minimap is featured on the top left, out of the way of the action, with very simple squares to represent players. This makes it easy for players to glance at and have an understanding of their position. There are also scores on the right side of the screen, that cleanly shows scores of the player and the leaderboard.

The style of Shielded Basin replicates 2-dimensional pixel games that were common in older arcade interfaces. The theme is a pirate game with the goal of collecting more coins than the other players. Obstacles such as islands and kraken monsters are included in the game to further challenge the participants while complimenting the theme. This theme was chosen because this arcade style is still very easy to comprehend, pushing our ideology of the game being easy and intuitive further. As well as a ship on an ocean is self explanatory as to how it moves throughout the world.

Sounds are meant to compliment the 2 dimensional pixel art by having an 8-bit sound (not all are truely 8-bit). Both the sounds and artwork point to a heavy influence by Pirates of the Caribbean (kraken, other player ships, etc).

All credit for music and sounds go to their original authors.
Music can be found at
https://www.youtube.com/watch?v=ihGI4nJV7pk&fbclid=IwAR1F1oNusU9PBhHnZUkcyK1B2uyNo7XI78TineFfhEW-yYFMgT51iuTV9JQ
Sounds can be found at
https://www.zapsplat.com/music/game-tone-single-coin-collect-1/?fbclid=IwAR1RVIB_JhYtRHv-v2Oyx_hSCitWPPr2jPbeLznA2hGclmtfS1_EjUp6HgY

Programming Issues and Bugs:

Majority of the programming issues we had, though minor, was source control. When updating code and committing it to our groups repository, many times we would run into conflicts, as well as a lot of hesitation when pushing code to ensure it would work with other members. Most of these issues were due to lack of git experience and were very much resolved near the end of development. Because of the languages used (JS, HTML, and CSS), we encountered no compatibility issues, as these languages are very universal. This felt very fitting as one of our games main focuses was universality.

We encountered many bugs during development. Majority involving collision detection and variable types. Most of these bugs were ironed out during production, but a few remain as a result of insufficient time. The most prominent restriction the game current holds as a result of these bugs are the player limit, which at this time is at 4. Other features remain unimplemented as a result of this as well, such as player naming, leaderboard naming, player attacking, a death screen, and procedurally generated island and other hazards.

What We Learnt and What We Would Have Done Differently:

We learnt the importance of having a well structured schedule. Because of many external factors, it was hard to have everyone on the same page at the same times. In the future, time permitting, we would create a more strict and well structured schedule that we could all follow to get tasks done.

We learnt the importance of demoing and feedback. We found it very valuable to showcase our demo product to the class and get their opinions and reactions. Because of this, we were able to more clearly set our final goals. In the future, we would have loved to do this process even earlier to give more time to reflect and work on the feedback received from the demoing process.

What We Learned from the Course:

Perhaps the most prominent topic we learnt from this course is the importance of the design cycle. Through learning about scrums, google sprints, and many historical achievements like at IDEO, and mishaps like the Atari Cosmos, we have come to realize the huge emphasis that must be put on proper design thinking and planning. This learning has propelled us to a better understanding of the best way to create and other products in our near future.

Shielded Basin Can be reached at these locations:

Source: https://github.com/natashadecoste/multiplayergame/
(Instructions to run source code is on the github)

Running Example: https://shielded-basin-77873.herokuapp.com/