**Technical analysis**

|  |  |  |  |
| --- | --- | --- | --- |
| Item Name | Requirements | Difficulties | Functions |
| Ship | Needs to be controllable  Stays in place in x, moves Y  Should speed up longer up/down is held  When hits iceberg, loss of life | Understanding collisions, understanding ship as a sprite | moveUp()  moveDown()  loseLife()  getHit()  getCoin()  die() |
| Icebergs | Appear at random heights  Move as group across the screen leftwards  Gradually speed up | Understanding collisions | generateIceberg()  setIcebergFreq()  setIcebergSpeed()  collide()  shrink()  shift() |
| Shrink gun | Aimable; when projectile interacts with iceberg, iceberg shirnks | Aiming, projectile, understanding shrink-collisions | shoot()  aim()  collide()  activate()  deactivate() |
| Coins | Appear randomly | understanding collisions | generate()  collapse()  shift() |
| Lives | Displayed at top of screen |  | addLife()  removeLife()  setNumLives() |
| Multiplayer | Split-screen | duplicating; no overlap/lag | begin() |
| Mines | Appear randomly, set lives to 0 if collision | understanding and differentiating collision | generate()  shift()  setMineFreq() |

Dependencies: no libraries or pre-written code

Design assumptions: Moving background, pause button, play button

**Prioritization**

1. Ship - main character, necessary to be a game
2. Icebergs - main opponent, necessary to be a game
3. Lives - main tracker, necessary to be a game
4. Pause - useful, integral to most gameplay
5. Shrink gun - prioritized in README, very useful to longevity of game
6. Mines, Coins - extra features, not core to game but fun
7. Multiplayer - adds usefulness, but depends on other features

**Feasibility analysis**

*Most difficult items to program*: icebergs, shrink gun

*Time required*: medium amount?

*Technical issue:* it is unclear whether it is a side-scroller or the boats can move around the screen. Do new icebergs appear?

Data Storage and Variables:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Variable Name** | **Code Name** | **Structure** | **Global or Local** | **Description** |
| Coins | numCoins | Int (array for multi) | Local | Counts the number of players’ coins |
| Lives | numLives | Int | Local | Number of coins of player |
| Game Speed | gameSpeed | Int | Global | The speed of mines/icebergs as they move across screen (shift-rate) |
| Mine frequency | mineFreq | Int | Global | The frequency of mine generation |
| Coin frequency | coinFreq | Int | Global | The frequency of coin generation |
| Iceberg frequency | icebergFreq | Int | Global | The frequency of iceberg generation |

**Functional code relational map**

shift(): all attack-objects shift over at the same rate

collide(): When icebergs/mines and ship collide, they experience the same collision but each call collide() and experience their own deaths

pause(): saves last game speed, sets current game speed to 0

start(): sets game speed to >0, starting the game

die(): sets game speed to 0, lives to 0

PERFORMANCE IMPACT: Nothing should require excess memory or slow anything down

Deadlines:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| Start coding ship | Finish ship movement | Start icebergs | continue icebergs | Finish icebergs | Finish mines | Finish ship lives | Coins | Game speed | Start multi | Mp | Finish mp |