Final documentation

Groupwork:

Mogau Seroka – Coding and game development, presentation/Powerpoint

Mihle Ndabula – Design, testing phases improvement of game functionality, documentation

Background and motivation:

Appropriate - Speed versus Obstacles was created to provide an adrenaline rushing and challenging experience for people who enjoy arcade style racing games. The aim of the game is to combine entertainment with skill development, like hand- eye coordination and thinking strategically.

Clear aims – Engagement: creating an addictive gameplay loop that encourages continuous playing.

-Skill development : It enhances the players decision making speed and hand-eye coordination

Problem statement:

Why – A lot of games don’t engage players because they are complex and too difficult to learn. There needs to be a game that balances challenging with appealing to both casual and competitive players.

What – the game mitigates this gap by giving straightforward controls along with increasingly difficult gameplay, this makes sure that players stay engaged while developing the skills mentioned above.

Approach:

How – research: looking at the existing racing games like roadblusters to find successful elements.

* design: creating a simple user interface and gameplay mechanics
* development: we used Java’s AWT and swing libraries for graphics and handling events.
* Testing: we conducted multiple rounds of playtesting so we can refine gameplay accordingly.

Milestones:

* Developing the concept: final game design
* Prototype creation: developing an initial playable version
* Testing: getting feedback from small groups of testers
* Final release: published the game

SDLC

1. Requirement analysis: finding key features and player needs.
2. Planning: Setting timelines and resource to be used
3. Design: In the game, the player can move left and right, pause the game, resume it, restart it and quit. There are squares moving in opposite direction to that of the player and if the player hits it the game ends.
4. Implementation: writing code using Java's AWT and Swing libraries.
5. Testing: different types of testing to see potential threats that could break the games code or disrupt the game.
6. Deployment: submitting the game (ask group members)
7. Maintenance:  Providing updates based on user feedback.

Appropriate model:

An agile model was used to allow iterative development and flexibility in response to the players (testers) feedback throughout the project life cycle.

Visual guide?

Game

|  |
| --- |
| GAME |
|  |
|  |
| + main():void |

|  |
| --- |
| work |
| -space:int  -speed:int  -width:int  -height:int  -WIDTH:int  -HEIGHT:int  -move:int  -count:int  -countPrev:int  -ocars:ArrayList<Rectangle>  -rand:Random  -linef:boolean  -isPaused:boolean  -isGameOver:boolean  -t:Timer |
| work()  -initialize():void  +addocars(boolean first):void  +paintComponent(Graphics g):void  +actionPerformed(ActionEvent e):void  +moveLeft():void  +moveRight():void  +keyPressed(KeyEvent e):void  +keyReleased(KeyEvent e):void  +keyTyped(KeyEvent e):void |