RISK ASSESSMENT AND MITIGATION

GROUP 5 - BITCRUSHED BOB

Maryam Mathews
Joseph Hinde
Jacob Mace
Will Aston
Zathia Jacquesson-Ahmad
Bulganchimeg Munkhjargal
Evan Weston

Risk Management Process

The team followed a four-stage risk management process to ensure risks were identified early and effectively controlled throughout the project lifecycle.

1. Identification:
   - Potential risks were identified across key areas: technology, requirements, management, and people.
   - Collaborative brainstorming sessions and experiences from previous projects were used to capture a comprehensive list of risks.
2. Analysis:
   - Each risk was assessed for likelihood and impact.
   - This analysis helps in deciding which risks require the most attention and resources
3. Planning:
   - For each significant risk, the team defined:
     - Avoidance / mitigation strategies: Attuned to each risk to reduce it’s likelihood or impact
     - Contingency plans: to provide clear responses if the risk occurred.
4. Monitoring:
   - Risks were reviewed at weekly project meetings and a risk register was developed to aid reviewing / monitoring
   - The team maintained status reporting and re-assessment to track new or changing risks.

Risk Register Format:

The format of the risk register has been selected by the team in order to optimise the efficiency of the risk management plan by clearly categorising each risk with it’s description, likelihood, impact, mitigation, ownership, and status.

The likelihood and impact categories allowed the team to efficiently address the risks that actually may occur, and also risks that would have severe consequences to the project.

The ownership and mitigation categories allowed the team to have transparency on who will be responsible for dealing with each risk and executing the corresponding mitigation strategies, thereby ensuring clear accountability and prompt action when required.

Risk Register:

Risk	Description	Likelihood	Impact	Mitigation	Ownership	Status
Software	Technical issues such as game freezing, performance drops, or compatibility problems with different systems.	Middle	High	Test the game on multiple devices early; optimize performance and use version control for backups	Jacob MacE Joseph Hinde Zathia Jacquess... Evan Weston	Finished
Software	Player gets stuck or movement doesn't work properly because of bugs	Middle	High	Add debug mode and visual collision checks; test with edge cases	Jacob MacE Joseph Hinde Zathia Jacquess... Evan Weston	Finished
Software	Performance drops as maze complexity or number of objects increases	Middle	High	simplify maze structure and optimize loops	Jacob MacE Joseph Hinde Zathia Jacquess... Evan Weston	Finished
Software	The final game fails to meet key project requirements	Low	High	Review requirements checklist regularly and test all features against specifications	Will Aston Maryam Mathews Bulganchimeg ...	Finished
Software	Scoring or hidden events don't feel fair or rewarding to players	Middle	Normal	Define a clear scoring system and add small hints or effects to make hidden events feel worthwhile	Will Aston Maryam Mathews Bulganchimeg ...	In progress

Software	Some hindering events may be too complex to implement within available time	Low	Normal	Simplify ideas early, focus on basic versions that work reliably	All	Finished
Software	Maze lacks clear objectives or direction, confusing players	Low	High	Add visual cues and a minimap or timer. Give tutorial	Will Aston Maryam Mathews Bulganchimeg …	Finished
Management	adding unnecessary features that delay project completion	Low	High	Define “must-have” and “optional” features	Will Aston Maryam Mathews Bulganchimeg …	Finished
People	Unequal contribution or low motivation within the team	Low	High	Set clear roles and deadlines, monitor progress and support each other	All	Finished
People	Lack of technical knowledge or experience delays development progress	Low	High	Identify skill gaps early, use tutorials and consult instructors when needed	All	Finished
People	Miscommunication among team members or misunderstanding project requirements	Low	High	Hold regular check-ins, document design decisions clearly and share updates	All	In progress
Management	Inadequate documentation or unclear versioning leads to confusion during development or submission	Low	High	Keep updated documentation, label all versions clearly	All	Finished

Estimation	Development takes longer than expected, delaying submission	Low	High	Split the project into small steps and finish the main game first.	All	Finished
People	A team member falls ill and cannot perform their role	Middle	Low	Join teams back together to ease workload between the team	All	In progress