

# Quokka Blokka Post-Mortem

## Design

We have been incredibly happy with the design of Quokka Blokka. We went from feeling deflated and doomed after our initial pitch to delivering a product we were confident with that satisfied the judges. We've put heaps of effort into the design and development of the game, which has paid off with a gripping and challenging game that is a joy to play.

## What went well

- The core concept of combining an endless runner with Tetris works really well. The block placement provides a unique twist on the genre, and the row elimination is an exciting challenge that keeps the game balanced.
- The controls are simple, only requiring four buttons, which means that the player doesn't have to wrap their head around a variety of different functions. Their similarity to Tetris games on the market especially helps with making the game accessible, as most people have played one of them before.
- The power-ups do a good job of alleviating the difficulty of the game while not making the game a cakewalk. They're also a lot of fun to use, with the special effects that take place when one activates making players visibly excited.
- The aesthetics bring a sense of charm and character to the game. The quokka in particular highlights this- it looks really cute when it's hopping along, and you feel bad when you squish them because of the loud SPLAT noise.
- The combination of randomly-placed blocks and structured obstacles makes each playthrough different, and presents a variety of unique challenges for the player to overcome. The player never knows what will be coming next, but can develop reactionary strategies to what they can see in front of them.
- The game encourages replaying for a high score. Its mechanics can be well understood after a couple of tries, and can be quickly replayed with new block placements and obstacles. With a leaderboard shown at the end of each run, with the top scorer emphasised, the player is given a reason to strive towards doing better.

## What went badly

- The game is hard to explain. The tutorials explain the game in enough depth, but they're quite wordy and people often skip them. We didn't want to force people into an obtrusive tutorial, and tried to overcome this by adding hints to the Get Ready screen, but the game doesn't speak for itself enough.
- The difficulty was incredibly hard to balance, and even now we don't think we have the balance right. People kept saying the game was hard, but always offered feedback conflicting with what everyone else said, making it hard to identify the root causes of the difficulty. We added in a 'band-aid' solution in the form of multiple lives to extend the player's game time without really solving the difficulty problem, but players have said they feel it's an unnecessary feature.
- The animations aren't totally smooth. This is something people always comment on, but it's not something that could be helped. We have a single artist doing just about all the graphical work for the game, and they couldn't get everything to a professional standard within the

timeframe. This isn't due to a lack of effort or skill, it's just a very demanding schedule with which to deliver time-consuming work to that standard.

## Development

The game was to be produced over four sprints to coincide with the milestone meetings Thursday nights.

- Sprint 1 was 4 weeks long, from Weeks 3 to 6
- Sprint 2 was 4 weeks long, from the mid-semester break to Week 8.
- Sprint 3 was 3 weeks long, from Weeks 9 to 11
- Sprint 4 was 2 weeks long, during Weeks 12 and 13.

Development took place with the impression that Sprint 3 and 4 were the one sprint. We initially divided our ideal expectations for completion into thirds.

Two of the members of the team having prior commitments the entirety of Week 12 ensured that the majority of the predicted development was contained within the first two sprints.

In the end, Sprints 3 and especially 4 saw a boom in the amount of work required due to play-testing reports and incomplete tasks that were carried forward.

As a group, we strived to maintain one stand-up meeting per week during Thursday nights and one online conference call per week on the weekend. We also communicated informally throughout the week via social media to have our work peer-checked and communicate our progress to the rest of the group.

### Sprint 1

#### Goal

Our mantra for Sprint 1 was to “find the fun” after our design problems discovered at the Pitch meeting. After some discussion, the team came to a consensus to develop an “endless-runner” instead of a “puzzle-platformer”. Our goals were to implement the core Tetris mechanics and the movement of the entity that would become the Quokka as well as an infinitely side-scrolling level and some block obstacles that the entity would have to navigate past.

#### Outcomes

- The Vision Holder created and developed a build that quickly demonstrated the initial concept of “Tetris Bridge”.
- After a week had passed and design decisions had been mostly finalised, this prototype was used to inform the design of another implementation by the Scrum-Master which was built from the ground-up with Quokka Blokk in mind.
- The Visual Artist and Producer were tasked with creating block formations to serve as obstacles for the Quokka to navigate past.
- The core mechanics of Tetris, the scrolling level, and the movement of the Quokka were completed, the team was consulted to test whether some measure of fun had been

achieved.

- The Scrum-Master went ahead of schedule and added other mechanics that had been scheduled for Sprint 2 such as:
- An entity was implemented that would follow the Quokka across the screen and would result in game over upon catching it, discouraging too much climbing. This pursuing entity would later become the fire.
- The Quokka was able to gain momentum by running and lose momentum by climbing which would respectively increase or decrease the speed of the game.
- Also added were high scores, to encourage testing within the team and increase the “fun factor” at the milestone meeting and the Obstacle Editor which allowed the Visual Artist and Producer to add their creations to the game without needing the programmer to implement them.

### Lessons

- As we had heard before, **“Kill your darlings”** - a lot of time was spent in discussion trying to arrive at consensus in design and implementation. One of the challenges was knowing when to continue attempting to make a design or implementation work, or when to change tactics and try something new.
- **Proper pacing** – the Scrum-Master “binge-programmed” for the milestone meeting, and while it did pay off with the sprint goal and beyond being met – it led to some complacency and burn-out in the latter sprints.
- **Keeping Team Members Busy** – Here we also learnt about making the best use of the entire team's time. The goals for this Sprint were very programmer-reliant which took away from the remaining two members of the team. The correct thing to do would have been to bring forward some of their tasks from later Sprints so to reduce the pressure they would soon face.
- **Better communication of ability** – Not everybody was familiar with everyone else's ability. This meant that at times, team members weren't aware that another team member could offer advice or help in completing a task.
- **Focus on immediate design** – At times, the team would get distracted discussing features slated for Sprints 2 or 3. These discussions ended up leading nowhere as they were undertaken once again in later Sprints when they were more relevant. This time could have been better used following the above lessons.

### Sprint 2

#### Goal

Our goals for Sprint 2 were to:

- improve the core mechanics we had added in Sprint 1
- begin creating and implementing game art

- implement a proper interface for the game which the player could navigate through
- add new power-up mechanics to 'spice-up' the movement of the Quokka

### Outcomes

- The pursuing entity was given visual representation through a wall of shifting fire. Conditions were also placed on the speed of the fire so that the player could increase their momentum in order to place more distance between the Quokka and the fire after an extended period of climbing blocks.
- The Quokka was given visual representation, and would appear using different sprites depending on whether he was running, climbing or falling through the level. The running and climbing motions were also animated.
- Game Over conditions were properly added. If the Quokka was caught by the fire, fell off the bottom of the screen, was crushed or bumped its head on a block – the game would end and the player would be taken to a Game Over menu.
- All individual blocks in the game that formed Tetris Blocks were given artistic representations beyond the minimalistic neon squares from Sprint 1. This included distinct art for each of the power-up blocks that depicted a corresponding natural element.
- Power-Up Blocks were added, these blocks appeared distinct from the standard blocks and were labelled with a natural element corresponding to the navigation ability they gave the Quokka – these abilities allowed the Quokka to avoid certain limitations of its movement for short times including gravity, the loss of momentum when climbing, and the generation of a straight platform to run on.
- Game-play was framed using new menus. The game experience was now broken up into numerous states: The Main Menu, an intermediate “Get Ready” menu, Game-play, a Game Over menu and Leaderboards which displayed the ten highest scores.
- Tetris mechanics were improved; controls were made more intuitive, row elimination bugs were solved, blocks gave a visual warning they were disappearing, and blocks were ensured to spawn out of the way of the Quokka.
- Different “playlists” of obstacles were implemented, allowing for tailored obstacles depending on whether the Quokka was in the upper or lower half of the screen.
- Obstacles underwent balancing to ensure that they weren’t placing the players in impossible scenarios

### Lessons

- **Communicate to the Player what's going on** – When power-ups were implemented, there was little communication to the player whether a power-up was actually happening. This led to some confusion in our limited play-testing. The team learnt that in the future if we were to add functionality that was different or extraordinary, we needed to make sure it was expressed as such. In later sprints, we added visual and aural sound effects to express that a power-up was active and give some indication of what that power-up did.

- **Revisiting is necessary** – What was once considered done in Sprint 1 doesn't always stay that way. In this Sprint, a lot of work went into editing and tailoring obstacles to different entry heights, even though the actual design and implementation of the obstacles was completed by the end of the previous Sprint. Over the course of the remaining weeks, many other things such as the GUI, the game art, power-ups and the difficulty were finished and then revisited later on due to new information, the addition of features or play-testing feedback.

## Sprint 3

### Goal

Our goal for Sprint 3 was to take on feedback and focusing on polishing and communicating mechanics to the player, improving the user interface and implementing the final items of game art.

### Outcomes

- The GUI for Main Menu, Get Ready and Game Over screens was completed. The user could now use the mouse to navigate through some of the menus.
- A Tutorial “Click-Through” was added to the game that described the flow of game-play, the rules of momentum, the power-up abilities and the “Game Over” conditions.
- Power-ups now triggered unique transparent screen effects that would display over the background to indicate that a power-up was active. These effects included a lightning storm, an updraft of air, and an underwater appearance.
- Sounds were now added to the game in order to help communication with the player.
  - Music was added to the Main Menu
  - Ambient fire crackling played during game-play,
  - “Squish” or “Sizzle” sounds matching the Game Over condition would signal how the Quokka was lost.
  - Power-ups featured unique sound effects that matched their visual effect; lightning, a breeze, and a wave.
- The team discovered a bug that prevented exporting the complete game. This was a barrier to play-testing as we could not share the full game until it was solved.
- The Artist drew many things, not all of which made it into the build before the milestone was due.

### Lessons

- **Plan Ahead** – The bug preventing exporting was found fairly early in the Sprint. Had the Scrum-Master been more attentive and more concerned, the easy solution could have been found earlier and there may have been more time for play-testing.
- **Communication to the Player is Really Important** – This Sprint saw the most apparent increase in playability, not due to added mechanics but because of prioritising the user

experience. In the future, the team hopes to approach the game from the perspective of someone who hasn't watched it develop in order to assess how well the game communicates, in addition to ensuring play-testing occurs earlier in the development process.

- **Smaller Workload, Better Quality** – The list of Sprint Goals achieved for previous Sprints were much longer than the list for Sprint 3. Though the team was more ambitious in previous Sprints and not all the Sprint goals were met, the completed goals were completed more thoroughly and very little revisiting was needed. The team learnt that after establishing the core of the game, it helps to tighten the focus of subsequent Sprints to ensure a complete and satisfactory update and reduce scope creep.

## Sprint 4

### Goal

This final Sprint was the most tumultuous of the four. The programmer and the Visual Artist were overseas due to prior commitments throughout the first half, in which very little development took place. The remaining two members took this time to commence and complete more rigorous play-testing of the game than had been done previously, now that the exporting bug had been fixed.

In the second half, the team prioritised the goals of responding to the feedback from the play-testing results and polishing the game in time for the final demonstration in Week 13.

### Outcomes

- In accordance with the play-testing results which suggested the game was too hard, we incorporated a “Lives” mechanic that allowed some grace for player error and highlighted the “threat” of creating an inefficient steep path and of the pursuing fire.
- The remaining art assets (Backgrounds, Sprite animations) were completed.
- Incomplete goals from Sprint 3 were completed including making the Quokka display more smoothly when in motion and implementing transitioning backgrounds,
- The GUI was polished and completed for all the menus in the game.
- The remaining identified bugs were taken care of.

### Lessons

- **Play-Test Earlier** – Related to the next lesson, better play-testing earlier in the development of the game would have allowed greater understanding of trends, more time to address or potentially even avoid concerns, time to test solutions to play-testing concerns and more time to come out with the best solution to teaching the player how to play the game.
- **Play-Testing Creates More Work** – At the beginning of Sprint 4, the team considered the level of work required to complete the game as quite small. However after Play-Testing, the list of things to do grew to address concerns raised by testers with a very finite time to complete them within.
- **Take the time to analyse trends** – The team was very quick to implement the “Lives” mechanic to address play-tester concern that the game was too difficult. The mechanic was

added in the eleventh hour, against genre convention. This was noted at the final demonstration. In the future, we hope to have the time to analyse trends better, and make more cautious decisions on how we address some issues, if at all.

- **Balancing is hard** – The playtesting feedback offered a variety of contradictory comments on the game’s difficulty, which didn’t assist in identifying the causes of the game’s difficulty. We were unsure of how to proceed, which is what led to the implementation of the ‘band-aid’ solution in the lives mechanic rather than a less jarring, more satisfying solution.
- **Ensure that the game is done before last minute** – Some readily apparent things ended up needing to be done right before the demonstration. In the future, we'd like to get better to accounting for these and making sure they are done earlier on – so that last minute can be kept free for unpredictable accidents or other things that may go wrong.
- **No-one reads tutorials** - The team learnt that next to no-one reads tutorials and that “you have to really nail a tutorial” - this is something going forward that we'd like to take to heart and in the future we hope to come up with different techniques for introducing the player to mechanics that we can test throughout the development of a game. We hoped to alleviate the need for a tutorial with the ‘hint’ system, but there’s a lot to take in with the game which means that getting one mechanic explained per playthrough just isn’t enough.

### What Went Well

- Our team worked well within the time we had to complete the game. As we progressed through the thirteen weeks, we became more aware of each other's individual strengths and our division of tasks became more even – allowing every member to contribute greatly to the finished product on top of our committee-driven design.
- Though we used prototypes to start development, our actual implementation was built from scratch with a simpler grid-based model. The simplicity of this model paid off when we were adding more and more features to our game that required more and more maths. There were many limitations of our prototypes that our new implementation avoided – as a team, we showed increasing ability in being able to drop old ideas for new ideas where necessary.
- As a team, we became more experienced with the tools of the trade. Our programmers learnt more about Unity, especially Unity 2D in which we previously had no experience. We were able to research sprite animations, shaders and sounds in order to complete the finished product. Our artist used Adobe Photoshop and Adobe Flash in order to create the art assets for the finished product and learned new techniques to aid in drawing in the particular 2D art style we selected for the game.
- Our milestones went very well and we maintained a lot of momentum in getting tasks done and pinpointing where our priorities laid (except when it came to play-testing). Our crunch for Milestone 1 to get fun game-play from the start and our step-by-step narrowing of scope for subsequent milestones served us well and ensured that we were able to provide fun gameplay and an engaging aesthetic into a moderately polished package by the deadline.

### What Went Badly

- We failed to make the best use of the entirety of our team during the lead-up to the first milestone. Our focus on creating fun gameplay through honing the core mechanic meant that there was no chance of art or a GUI being added, and as such work never commenced on them until very late into the first Sprint. In the future, we would like to be more flexible

with our Sprint goals and bring certain goals forward in order to keep all team members busy if necessary.

- When we started, we didn't have a firm grasp of every other member's abilities and talent. This made division of tasks, asking for help and communication in general more difficult than they should have been. Over time, we grew more accustomed with each other and became more able to work in a team – however, we still lost a lot of ground in the first few weeks and the majority of the work was completed in the last week of the Sprint
- After our success in the first milestone, there was some complacency on the part of the programmer which hindered the contribution of the Artist and the Producer who was providing obstacle designs. A common catchphrase in the second Sprint was “2 weeks ahead of schedule”, though this was not true for long.

## Lessons

- Put the time aside to play-test early.
- Play-testing multiplies the amount of work you perceive is left to do.
- SCRUM works even better if you put more work into the team right at the beginning.
- Be flexible with your Sprint Goals if it means making the most of your team.
- Don't show off things before you're communicating them to the player adequately.
- Introducing players to mechanics can be a challenge.